ASSORTMENT OPTIMIZATION AND PRICING PROBLEMS UNDER MULTI-STAGE MULTINOMIAL LOGIT MODELS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Yuhang Ma

December 2019

ASSORTMENT OPTIMIZATION AND PRICING PROBLEMS UNDER MULTI-STAGE MULTINOMIAL LOGIT MODELS

Yuhang Ma, Ph.D.

Cornell University 2019

In most E-commerce scenarios such as hotel booking and online shopping, products are not offered to customers simultaneously. Instead, they are divided into different webpages and presented to customers sequentially. In this thesis, we focus on solving a common problem faced by online retailers: when products are revealed to customers sequentially, which products should the retailers display at each stage and what prices should the retailers charge for each product so that the expected revenue can be maximized? To solve those problems, we generalize the classical multinomial logit model to capture the customer's choice behavior under the sequential setting and present efficient algorithms for different generalized choice models and different operational constraints.

BIOGRAPHICAL SKETCH

Yuhang grew up in Yuyao, Zhejiang, a beautiful town along the east coast of China. In 2014, she received her Bachelor of Science in Applied Mathematics from Peking University, where she developed the foundation of her mathematics and optimization skills and became interested in their applications in the real world. She then joined the School of Operations Research and Information Engineering at Cornell to further pursue her interest. During her time at Cornell, she has been working on approximation algorithms, with focus on applications in revenue management problems.

To my family.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor, Professor Huseyin Topaloglu, for his invaluable and patient support throughout my time at Cornell. His passion about research has always encouraged me and I am so grateful to have him as my PhD advisor.

I would also like to express my sincere gratitude to Professor Shane Henderson and Professor Yoav Artzi for being on my thesis committee. I thank them for all the insightful comments and constructive feedback on my thesis.

I also feel very grateful to have the precious opportunity to work with Professor Paat Rusmevichientong. I thank him for all the inspiring discussions. It is a great honor to work with him.

I would like to thank Xin Zhao, Wentian Huang, Jingyuan Wang, Pu Yang, Tiandong Wang, Sijia Ma, Zhen Lian, Mika Sumida and many other friends who have accompanied me in such a special journey as well. It is them who made my past years at Cornell so unforgettable.

Last but not least, I would also like to thank my parents, Minlong Ma and Lingling Xu, and my boyfriend, Chendi Ni. I thank them for always believing in me. I couldn't have finished this thesis without them. I thank them for all.

TABLE OF CONTENTS

Biographical Sketch iii
Dedication iv
Acknowledgements v
Table of Contents vi
List of Tables viii

1 Introduction **1**
1.1 Background 1
1.2 Related Work 3
1.3 Thesis Organization 6

2 Multinomial Logit Model with Sequential Offerings **9**
2.1 Introduction 9
2.2 Problem Formulation and Complexity 16
2.3 Multiple Knapsack Representation 20
2.4 Approximate Feasible Set 23
2.5 Performance Guarantee 27
2.6 Dynamic Programming Formulation 31
2.7 Fully Polynomial-Time Approximation Scheme 37
2.8 Constraints on the Offered Sets of Products 41
2.9 Numerical Experiments 46
2.9.1 Upper Bound on the Optimal Expected Revenue 47
2.9.2 Randomly Generated Test Problems 50
2.9.3 Test Problems for Appointment Slot Choices 55
2.10 Conclusions 59

3 Multinomial Logit Model with Impatient Customers **60**
3.1 Introduction 60
3.2 Multinomial Logit Model with Impatient Customers 67
3.3 Unconstrained Assortment Optimization 71
3.4 Joint Pricing and Assortment Optimization 76
3.4.1 Optimal Prices under Fixed Assortments 76
3.4.2 Optimal Assortments and Prices 84
3.5 Assortment Optimization under a Space Constraint 88
3.5.1 Constructing Collections of Candidate Assortments 92
3.5.2 Combining Candidate Assortments 97
3.6 Computational Experiments 101
3.6.1 Prediction Ability on the Dataset from Expedia 102
3.6.2 Joint Pricing and Assortment Optimization 107
3.6.3 Assortment Optimization under a Space Constraint 112
3.7 Conclusions 115

4 Conclusion **117**

A Appendix for Chapter 2 **119**
A.1 Upper Bound on State Variable 119
A.2 Computation of Thresholds 120
A.3 Nested by Revenue Sets 122

B Appendix for Chapter 3 **125**
B.1 Change in Expected Revenue with Product Exchanges 125
B.2 Structure of the Optimal Solution under Space Constraints 126
B.3 Constructing Candidate Assortments 126
B.4 Combining Candidate Assorments 129
B.5 Bound on the State Variable 134
B.6 Assortment Optimization under a Cardinality Constraint 135
B.7 Preprocessing the Dataset from Expedia 143
B.8 Running Time for Fitting the Choice Models 144
B.9 Upper Bound for Joint Pricing and Assortment Optimization . . . 146
B.10 Upper Bound under a Space Constraint 147

Bibliography **150**

LIST OF TABLES

2.1 Performance of our FPTAS and the neighborhood search heuristic. 53
2.2 Performance improvement when we make offers in two stages rather than in one stage. 55
2.3 Performance of our FPTAS and the approach that offers appointment slots only in the first stage. 59

3.1 Comparison of the fitted IML and SML models on the dataset from Expedia. 106
3.2 Performance of the greedy search heuristic for joint pricing and assortment optimization. 112
3.3 Performance of the FPTAS for assortment optimization under a space constraint. 115

B.1 CPU seconds to estimate the parameters of our choice model. . . 145

CHAPTER 1
INTRODUCTION

1.1 Background

In traditional revenue management literature, customer's demand is modeled by using an exogenous random variable. However, such modeling cannot capture certain aspects of customer behavior. For example, if a consumer cannot find an item she wanted among the currently offered products, she probably would switch to purchasing something similar from the available products. The fact that the demand for one product depends on the availability of other products as well as their attributes and prices cannot be characterized by the exogenous demand models. Recent years have seen prosperity in revenue management field since the introduction of discrete choice models that capture such *substitution behaviors*. Under these choice models, it is more difficult for retailers to decide which products to offer and what prices to charge for each product to maximize the expected revenue.

In the rest of section, we will introduce some basic concepts in the revenue management field that are important in our thesis.

Choice Models. Consider a universe of n products $\mathcal{N} = \{1, ..., n\}$, and the no-purchase option is denoted as 0. Given any offer set $S \subset \mathcal{N}$, a discrete choice model characterizes the probability that a customer purchases product j from the offer set S as $P_j(S)$ and the probability that she would leave without purchasing anything as $P_0(S)$.

Random Utility Maximization Principle. For a rational customer, her choices should follow the Random Utility Maximization (RUM) principle, in which she associates random utilities with the products and the no-purchase option and always chooses the item with the largest utility. Denote the random utility associated with product $i \in \mathcal{N}$ as U_i and the random utility associated with the no-purchase option as U_0, then the probability of purchasing product j from offer set S is

$$P_j(S) = \mathbb{P}(U_j \geq \max_{i \in (S \setminus \{j\}) \cup \{0\}} U_i),$$

and the no-purchase probability is

$$P_0(S) = \mathbb{P}(U_0 \geq \max_{i \in S} U_i).$$

Assortment Optimization Problems. Due to the substitution behavior of customers, there is a trade-off between offering more products to earn more potential purchases and providing fewer items to attract customers to higher-revenue products. Denote the revenue of product $i \in \mathcal{N}$ as r_i. The retailer wants to offer a set of products S^* to maximize the expected revenue, i.e.

$$S^* = \arg\max_{S \subset \mathcal{N}} \sum_{j \in S} r_j P_j(S).$$

There could be additional operational constraints on the offered set of products. For example, there may be only limited space on a brick-and-mortar shelf and the retailer is only able to offer a set of products satisfying such total *space* constraints. Assume the space associated with each product $i \in \mathcal{N}$ is c_i and the available total space consumption is b. The best set of products to offer to maximize the expected revenue is

$$S^* = \arg\max_{S \in \mathcal{F}} \sum_{j \in S} r_j P_j(S),$$

where $\mathcal{F} = \{S \subset \mathcal{N} : \sum_{j \in S} c_j \leq b\}$ is the set of all the feasible sets of offered products. In particular, if every product consumes a unit space, i.e. $c_i = 1, \forall i \in \mathcal{N}$, this problem is called the assortment optimization problem under the total *cardinality* constraint.

Joint Assortment and Pricing Problems. Besides the choice of which products to offer, the retailers may be also interested in finding the revenue-maximizing price to charge for each product. Following the standard assumption in the pricing literature, we assume the utilities of products are linear in price and all the products have the same price sensitivity. Denoting the price of any product $i \in \mathcal{N}$ as p_i, the utility of product i can be written as,

$$U_i(p_i) = \alpha_i - \beta p_i + \epsilon_i,$$

where α_i is called the quality constant of product i, β is the homogeneous price sensitivity and ϵ_i is the random part of the utility. Given an offer set $S \subset \mathcal{N}$ and a price vector $\boldsymbol{p} = (p_1, ..., p_n) \in \mathbb{R}^n_+$, the choice model satisfying the Random Utility Maximization principle is $P_j(S, \boldsymbol{p}) = \mathbb{P}(U_j(p_j) \geq \max_{i \in S \setminus \{j\} \cup \{0\}} U_i(p_i))$, and the retailers are interested in solving the following problem

$$\max \left\{ \sum_{j \in S} p_j P_j(S, \boldsymbol{p}) : S \subset \mathcal{N}, \boldsymbol{p} \in \mathbb{R}^n_+ \right\}.$$

1.2 Related Work

A lot of choice models have been developed to capture different kinds of interactions between the demands of substitute alternatives, such as the multinomial logit model, the nested logit model, the mixed logit model, etc. For the multinomial logit model, the unconstrained assortment optimization problem is first discussed by [19] and [41]. [36] then studies the assortment optimization problem under a total cardinality constraint. A linear programming formulation is later presented in [40] for the assortment optimization problem under the constraints with a totally unimodular structure. [43] discusses the joint assortment optimization and pricing problem. For the nested logit model, the products are divided into disjoint nests and customers have different substitution behavior within each nest. [7] first studies the unconstrained assortment problem. [22] discusses the assortment optimization problem when there are space constraints within each nest. [17] later extends the discussion on the assortment optimization problem to the constraints across nests. [28] studies the pricing problem when price sensitives are the same within each nest, and [23] extends the analysis to more general nested attraction models. For a mixture of logit models, we assume there are multiple customer types and customers of different types choose according to different multinomial logit models. [6], [33] and [37] show that the unconstrained assortment optimization problem is already NP-hard, provide integer programming formulations, and study some special cases where efficient solutions and approximation algorithm exist. [34], [9] and [16] design FPTAS for solving the general unconstrained assortment problem.

In addition to those variants of the multinomial logit model, there are other choice models that capture the customer's substitution behavior as well. [5] considers a choice model, where customer's substitution behavior is modeled by transitions in a Markov chain, and solves the unconstrained assortment problem. [11] further studies the assortment optimization problem under the capacity constraints. In [12], the authors consider three different settings of the pricing problems under the Markov chain choice model. We refer readers to [4], [10] for the assortment problems under more general choice models, where customers make purchase decisions according to their preference lists. Morevover, some variants of the assortment problems are also studied. [38] considers a robust version of the assortment optimization problems under the multinomial logit model, where the parameters of the model are not known but are taken from some uncertainty set. In [44] and [21], the authors discuss assortment optimization problems under a generalized multinomial logit model, where the preference weight of the no purchase option is a function of the products that are not offered.

All the choice models discussed so far mainly focused on the single-stage problem, where the retailers always offer all the available products at once. However, in most E-commerce scenarios such as hotel booking and online shopping, products are divided into different webpages and presented to customers sequentially. The customers may make the final decision of purchasing or leaving before viewing all the offered products. Under such a setting, the online retailers not only need to decide what products to offer, but also in which sequence to offer the products to maximize the expected revenue. In this thesis, we will generalize the classical multinomial logit model from the single-stage setting to a multi-stage framework by proposing two different choice models.

There is already some work on the assortment optimization problems, where the assortments are offered through multiple stages. In [20] and [3], the authors consider choice models where the number of stages is sampled from an exogenous distribution and is independent of assortments offered on each page. However, in our model, the number of stages that the customers will view is also based on the assortments of offered products, which aligns more with reality. [18] studies a two-stage multinomial logit model, where there are two disjoint sets of potential products that can be offered at each stage. Thus, if the retailers decide to offer one product, they would know which stage to offer it immediately. However, in our model, retailers need to further decide at which stage to offer such a product, which adds more complexity to the problem. [2] focuses on a choice model where the customers include each product in the consideration set with a fixed probability. Although they also consider the case where the customers make a decision without seeing the whole assortment, they do not incrementally view the products. [35] discusses a cascade click model where the sequence of offered products is already fixed and only the pricing problem is considered.

1.3 Thesis Organization

In this thesis, we consider two different multi-stage choice models, where the customers will incrementally view the assortments of offered products. In both models, we assume the customer's choice process will take place in multiple stages, and at each stage of the choice process, the customer will be offered a set of products, which does not overlap with the previous stages.

In Chapter 2, we assume the number of stages is fixed a priori. If the customer makes a purchase within the current stage, she terminates the choice process with the purchase; otherwise, she proceeds to the next stage. If she reaches the no-purchase option at the final stage of the choice process, she leaves the system without buying anything. In this model, we assume the choice process within each stage is governed by a multinomial logit model. The goal is to find what products to offer at each stage to maximize the expected revenue.

We first show that it is an NP-hard problem to find an assortment to offer at each stage under this choice model. Motivated by this hardness result, we propose a fully-polynomial time approximation scheme (FPTAS) to solve the problem. The running time of our FPTAS is polynomial in the number of products but is *exponential* in the number of stages, which is fixed a priori by the assumption. We then further discuss the case where there is a limit on the number of offered products at each stage or a limit on the total number of offered products from all of stages. We generalize the previous proposed FPTAS to solve these constrained problems.

In Chapter 3, we consider the case where the consumers are impatient in the sense (i) they will leave with a purchase as soon as there is an offered product with larger utility than a minimum acceptable utility, which we refer to as the utility of the outside option; (ii) each customer has her own patience level, i.e. the maximum number of stages that she is willing to stay. Therefore, at each stage, the customer could terminate the choice process (i) either with a purchase if the product with the largest utility at the current stage has larger utility than the outside option; (ii) or without buying anything if she already reaches the final stage of her patience level. Otherwise, she will proceed to the next stage. In this model, we use the same utility model as in the multinomial logit model, where the random part of utilities all follow the Gumbel distribution. We then study three different assortment optimization and pricing problems.

First, we consider the unconstrained assortment optimization problem, where we choose the assortment of products to offer as well as the sequence of offering the products. We show that the optimal assortments follow the ordering of product revenues, and building on this result, we develop a dynamic programming algorithm to compute the optimal assortments in polynomial time. Second, we consider the joint assortment and pricing problem, where we not only need to choose the assortments of products to offer and their offer sequence, but also the prices to charge. We show that under a fixed assortment decision, the expected revenue is concave in the no-purchase probabilities over different numbers of stages and the optimal prices can be recovered from those optimal no-purchase probabilities . Building on this result, we give a heuristic that is guaranteed to obtain at least 50% of the optimal expected revenue when both the assortments and prices are decision variables. Third, we study the assortment problem under a total space constraint of the offered assortments. This problem is NP-hard even when there is only one stage. We give an FPTAS with a runtime *polynomial* in both the number of products and the number of stages. Finally, through numerical experiments, we show that our approximation algorithms perform much better than their theoretical guarantees in practice.

In Chapter 4, we conclude the thesis and discuss some possible future directions on the choice models under the multi-stage framework.

CHAPTER 2

MULTINOMIAL LOGIT MODEL WITH SEQUENTIAL OFFERINGS

2.1 Introduction

In traditional revenue management models, it is common to model the demand for each product by using an exogenous random variable that does not depend on what other products are made available to the customers. In many retail settings, however, customers choose and substitute among the products that are offered to them, in which case, the demand for a product depends on what other products are made available to the customer. There is a recent surge of revenue management models that explicitly capture such a customer choice process. Nevertheless, much of the work in this stream of literature assumes that the customers view the whole assortment of products offered to them simultaneously, but it is not difficult to run into cases where the customers gradually view the assortment in multiple stages.

When selling products in online retail, for example, the firm may display the search results to a customer sequentially through multiple webpages. In this case, the goal is to decide which products to offer on each page of search results and in which order to present the pages, to maximize the expected revenue obtained from a customer. When scheduling healthcare appointments over the phone, a reasonable objective for the service provider is to maximize the probability that a patient books an appointment. To gently guide the patient through the choice process, the service provider may offer sets of appointment slots sequentially. In this case, the goal is to decide which sets of appointment slots to offer to the patient and in which order to offer the sets, to maximize the probability that the patient books an appointment slot. Thus, if the customers gradually view the assortment in multiple stages, then we need to decide not only what assortment of products to offer, but also the order in which we should offer the products. In addition, it may be necessary to limit the number of products offered in each stage, for example, to ensure that the limited space on the webpage can accommodate the products offered in each stage. Similarly, to avoid overwhelming the patient with a large number of options, it may be desirable to limit the total number of appointment slots offered over all stages.

In this chapter, we consider assortment optimization problems, where the choice process of a customer takes place in multiple stages. There is a finite number of stages in the choice process, which is fixed a priori. In each stage of the choice process, we offer an assortment of products that does not overlap with the assortments offered in the previous stages. If the customer makes a purchase within the assortment offered in the current stage, then she leaves the system with a purchase. Otherwise, the customer proceeds to the next stage. If the customer reaches the end of the last stage without a purchase, then she leaves the system without a purchase. We use the multinomial logit model to capture the choice process of the customer in each stage. The goal is to find an assortment to offer in each stage to maximize the expected revenue obtained from a customer.

In our initial treatment of the assortment optimization problem, we focus on the case where there is no limit on the number of products that we can offer in any stage, but we also discuss the case where there is a limit on the number of products offered in each stage or there is a limit on the total number of products offered over all stages.

Main Contributions. We show that the problem of finding an assortment to offer in each stage to maximize the expected revenue obtained from a customer is NP-hard (Theorem 2.2.1). Motivated by this result, we develop a fully polynomial-time approximation scheme (FPTAS) for the problem as follows. First, we cast our assortment optimization problem as maximizing a nonlinear function over a certain feasible set $\mathcal{P}$, but checking whether a given point is in $\mathcal{P}$ requires finding a feasible solution to an intractable multiple knapsack problem (Lemma 2.3.1). Second, we give an approximate version of the feasible set $\mathcal{P}$ by aligning the cumulative capacity consumptions of the products in the multiple knapsack to a geometric grid. Using $\widetilde{\mathcal{P}}$ to denote the approximate version of the feasible set, we bound the loss in the expected revenue when we maximize the nonlinear function over the approximate feasible set $\widetilde{\mathcal{P}}$ (Theorem 2.5.2). Third, we show that we can use a dynamic program to check whether a given point is in $\widetilde{\mathcal{P}}$ and we enumerate over the elements of $\widetilde{\mathcal{P}}$ to maximize the nonlinear function over the approximate feasible set $\widetilde{\mathcal{P}}$. Accounting for the number of operations to solve the dynamic program, we get our FPTAS (Theorem 2.7.1).

Letting n be the number of products among which we choose a sequence of assortments and m be the number of stages, for any $\epsilon \in (0, 1)$, our FPTAS runs in $O(mn^{2m}(\log(n\kappa))^{m-1} (\log(n\nu))^m/\epsilon^{2m-1})$ operations to provide a $(1 - \epsilon)$-approximate solution. Here, κ is the largest value for the product of the revenue and preference weight of a product and ν is the largest value for the preference weight of a product, after normalizing the smallest revenue and preference weight to one. Thus, for a fixed number of stages, the running time of our FPTAS is polynomial in input size and the reciprocal of the precision.

If we have a limit on the number of products offered in each stage, then the term n^{2m} in the running time is replaced by n^{3m}, whereas if we have a limit on the total number of products offered in all stages, then the term n^{2m} in the running time is replaced by n^{2m+1}.

We also provide some insight into the form of the optimal assortment. [19] and [41] show that if there is a single stage in the choice process, then the optimal assortment is nested by revenue, including a certain number of products with the largest revenues. So, we can efficiently find the optimal assortment by checking the expected revenue from each nested by revenue assortment. We show that if there are multiple stages, then the union of the optimal assortments to offer in each stage is nested by revenue. However, this result does not allow us to find the optimal assortment efficiently, since it does not characterize the stage in which each product should be offered. Thus, we defer this result to the appendix.

Lastly, in our assortment optimization problem, we a priori fix the assortments that we offer in all of the stages. A natural question is whether there is any value in adjusting the assortment offered in a particular stage based on the choice trajectory of a customer in the previous stages. Note that if a customer is in a particular stage in the choice process, then her choices in all of the previous stages must have been no purchase. Therefore, given that a customer is in a particular stage in the choice process, there is only one possible choice trajectory of the customer in the previous stages, indicating that there is no value in adjusting the assortment offered in a particular stage based on the choice trajectory of a customer in the previous stages.

Literature Review. There is some work on assortment optimization problems, where the assortment is offered in multiple stages. [20] study a problem in online retail, where the assortments of products are presented in multiple pages. Each customer picks the number of pages to view according to a distribution that is exogenously fixed and she chooses among all of the products offered on those pages. The authors give an approximation algorithm. Focusing on healthcare appointment scheduling on the phone, [31] consider the case where the service provider offers assortments of appointment slots to a patient in multiple stages. If a patient is offered an assortment that includes appointment slots she is interested in, then she chooses among them uniformly. The authors characterize the optimal sequence of appointment slots to offer. [18] work with a two stage multinomial logit model, where there are two disjoint sets of products that can potentially be offered in the two stages. Thus, if a product is offered, then the stage in which it will be offered is fixed a priori. Focusing on the case with two stages, the authors give an efficient optimal algorithm. In our model, if we decide to offer a product, then we still need to choose the stage in which to offer the product. As a result, our problem is NP-hard. Furthermore, we work with multiple stages.

As mentioned above, [19] and [41] show that the assortment optimization problem under the multinomial logit model can be solved efficiently when there is a single stage. [36] show how to solve the same assortment optimization problem when there is a cardinality constraint limiting the total number of offered products. [38] study robust assortment optimization problems under the multinomial logit model when the parameters of the model are not known, but they take values in an uncertainty set. [43] considers the problem of jointly finding an assortment of products to offer and their corresponding prices, when the customers choose under the multinomial logit model and there is a constraint on the total number of offered products. [40] give a linear programming formulation when the constraints have a totally unimodular structure. [44] and [21] solve assortment optimization problems under a more general version of the multinomial logit model, where the preference weight of the no purchase option increases as a function of the products that are not offered.

In a mixture of multinomial logit models, we have multiple customer types and customers of different types choose according to different multinomial logit models. [6], [33] and [37] show that the assortment optimization problem under a mixture of multinomial logit models is NP-hard, provide integer programming formulations, study special cases admitting efficient solutions and give approximations. [32] provide an upper bound on the optimal expected revenue by using a relaxation that offers different assortments to customers of different types. The objective function of our assortment optimization problem is a sum of fractions. Considering assortment optimization problems with multiple customer types, [34], [9] and [16] design FPTAS for maximizing various sums of fractions.

In our FTPAS, we draw on [9], where the authors use the connections of their problem to the knapsack problem by aligning the cumulative capacity consumptions to a geometric grid. Due to the multiple stages in our choice process, the numerators and denominators in our fractions are nonlinear and we need to carefully account for the errors resulting from the geometric grid. The error is exponential in the number of products, but by a judicious choice of the performance guarantee, we get an FPTAS.

For representative assortment optimization work under other choice models, we refer the reader to [7] for the nested logit model, [1] for the ranking-based choice model and [5] for the Markov chain choice model.

Organization. In Section 2.2, we formulate our assortment optimization problem with multiple stages and show that it is NP-hard. In the rest of this chapter, we focus on developing an FPTAS. In Section 2.3, we give an alternative formulation of our assortment optimization problem that maximizes a nonlinear function over a certain feasible set $\mathcal{P}$. Checking whether a given point is in $\mathcal{P}$ is difficult. In Section 2.4, we give an approximation to the feasible set $\mathcal{P}$. We refer to this approximation as $\widetilde{\mathcal{P}}$. In Section 2.5, we bound the loss in the expected revenue when we maximize the nonlinear function over the approximate feasible set $\widetilde{\mathcal{P}}$. In Section 2.6, we give a dynamic program to check whether a given point is in $\widetilde{\mathcal{P}}$. In Section 2.7, by accounting for the number of operations to solve the dynamic program and to enumerate over the elements of $\widetilde{\mathcal{P}}$, we give our FPTAS. In Section 2.8, we make extensions to the case where there is a limit on the number of products that we can offer. In Section 2.9, we give numerical experiments. In Section 2.10, we conclude.

2.2 Problem Formulation and Complexity

We have n products indexed by $N = \{1, \ldots, n\}$. In the choice process, we have m stages indexed by $M = \{1, \ldots, m\}$. We use the set $S^k \subseteq N$ to denote the set of products that we offer in stage k. Since we can offer a product in at most one stage, the feasible sets of products that we can offer over all stages are $\mathcal{F} = \{(S^1, \ldots, S^m) : S^k \subseteq N \; \forall\, k \in M,\; S^k \cap S^\ell = \varnothing \; \forall\, k \neq \ell\}$. In the choice process, a customer chooses within the offered set of products in each stage according to the multinomial logit model and the choices of a customer in different stages are independent. In particular, we use $v_i^k > 0$ to denote the preference weight of product i when this product is offered in stage k. Normalizing the preference weight of the no purchase option in each stage to one, if we offer the sets of products $(S^1, \ldots, S^m)$ over all of the stages, then a customer in stage k chooses product i with probability $\mathbf{1}(i \in S^k)\, v_i^k / (1 + \sum_{j \in S^k} v_j^k)$, where $\mathbf{1}(\cdot)$ is the indicator function. A customer in stage k does not make a purchase with probability $1/(1 + \sum_{j \in S^k} v_j^k)$, in which case, she moves on to stage $k+1$. If a customer does not make a purchase by the end of the last stage m, then she leaves the system without a purchase. For a customer to purchase product i in stage k, she needs to not make a purchase in stages $1, \ldots, k-1$ and she needs to purchase product i in stage k. Therefore, if the sets of products offered over all stages are $(S^1, \ldots, S^m)$, then a customer purchases product i in stage k with probability $\prod_{\ell=1}^{k-1} \frac{1}{1+\sum_{j \in S^\ell} v_j^\ell} \times \frac{\mathbf{1}(i \in S^k)\, v_i^k}{1+\sum_{j \in S^k} v_j^k}$. The revenue associated with product i is $r_i > 0$. Our goal is to find sets of products to offer over all stages to maximize the expected revenue obtained from a customer, which yields the problem

$$\widehat{Z} = \max_{(S^1,\ldots,S^m)\in\mathcal{F}} \left\{ \sum_{i\in N}\sum_{k\in M} r_i \left\{ \prod_{\ell=1}^{k-1} \frac{1}{1+\sum_{j\in S^\ell} v_j^\ell} \right\} \frac{\mathbf{1}(i\in S^k)\, v_i^k}{1+\sum_{j\in S^k} v_j^k} \right\}$$
$$= \max_{(S^1,\ldots,S^m)\in\mathcal{F}} \left\{ \sum_{k\in M} \left\{ \prod_{\ell=1}^{k} \frac{1}{1+\sum_{i\in S^\ell} v_i^\ell} \right\} \sum_{i\in S^k} r_i\, v_i^k \right\}, \quad (2.1)$$

where the second equality follows simply by arranging the terms. Note that we can interpret $\prod_{\ell=1}^k \frac{1}{1+\sum_{i\in S^\ell} v_i^\ell} \times \sum_{i\in S^k} r_i\, v_i^k$ as the expected revenue in stage k.

Computational Complexity. Next, we characterize the computational complexity of problem (2.1). In particular, we focus on the feasibility version of problem (2.1), where the goal is to find a solution $(S^1,\ldots,S^m)\in\mathcal{F}$ with an expected revenue that is no smaller than a given threshold. In the next theorem, we show that the feasibility version of problem (2.1) is NP-complete.

Theorem 2.2.1 *The feasibility version of problem* (2.1) *is NP-complete.*

Proof. The problem is in NP. We use a reduction from the partition problem, which is a well-known NP-complete problem; see [24]. In the partition problem, we have n products indexed by $N=\{1,\ldots,n\}$. The weight of product i is c_i. The weight of each product is an integer and we have $\sum_{i\in N} c_i = 2\,t$. The goal of the partition problem is to find a set of products S such that $\sum_{i\in S} c_i = \sum_{i\in N\setminus S} c_i = t$. Using the partition problem, we construct an instance of the feasibility version of problem (2.1) as follows. We have n products indexed by $N=\{1,\ldots,n\}$ and two stages indexed by $M=\{1,2\}$. The revenue of product i is $r_i = 1$ for all $i\in N$. The preference weight of product i in stage k is $v_i^k = c_i/t$ for all $i\in N$, $k\in M$. The expected revenue threshold is $3/4$. We proceed to showing that there exists $(S_1,S_2)\in\mathcal{F}$ such that the expected revenue from the solution (S_1,S_2) is $3/4$ or more if and only if there exists $S\subseteq N$ such that $\sum_{i\in S} c_i = t$. Noting the expression for the expected revenue in problem (2.1), for the solution $(S_1, S_2) \in \mathcal{F}$ to provide an expected revenue of 3/4 or more, this solution must satisfy the inequality

$$\frac{\sum_{i \in S_1} c_i / t}{1 + \sum_{i \in S_1} c_i / t} + \frac{1}{1 + \sum_{i \in S_1} c_i / t} \times \frac{\sum_{i \in S_2} c_i / t}{1 + \sum_{i \in S_2} c_i / t} \geq \frac{3}{4},$$

which is, arranging the terms, equivalent to $(t + \sum_{i \in S_1} c_i)\,(t + \sum_{i \in S_2} c_i) \geq 4\,t^2$. Also, if $(S_1, S_2) \in \mathcal{F}$, then we have $S_1 \cap S_2 = \emptyset$, which implies that $\sum_{i \in S_1} c_i + \sum_{i \in S_2} c_i \leq \sum_{i \in N} c_i = 2\,t$, so that $\sum_{i \in S_2} c_i \leq 2\,t - \sum_{i \in S_1} c_i$. In this case, if $(S_1, S_2) \in \mathcal{F}$, then we have $(t + \sum_{i \in S_1} c_i)\,(t + \sum_{i \in S_2} c_i) \leq (t + \sum_{i \in S_1} c_i)\,(3\,t - \sum_{i \in S_1} c_i) = 4\,t^2 - (\sum_{i \in S_1} c_i - t)^2 \leq 4\,t^2$. Thus, for the solution $(S_1, S_2) \in \mathcal{F}$ to provide an expected revenue of 3/4 or more, the last chain of inequalities must hold as equalities, which happens only when $\sum_{i \in S_1} c_i = t$ and $\sum_{i \in S_2} c_i = \sum_{i \in N \setminus S_1} c_i = t$. So, there exists $(S_1, S_2) \in \mathcal{F}$ such that the expected revenue from the solution (S_1, S_2) is 3/4 or more if and only if there exists $S \subseteq N$ such that $\sum_{i \in S} c_i = \sum_{i \in N \setminus S} c_i = t$. ■

Thus, problem (2.1) is NP-hard even when there are only two stages in the choice process with $v_i^1 = v_i^2$ for all $i \in N$ and $r_i = 1$ for all $i \in N$. Note that if the revenues of all products are one, then the objective function of problem (2.1) is the probability that a customer makes a purchase. Before us, to show that the assortment optimization problem under a mixture of multinomial logit models is NP-hard, [37] use a reduction from the partition problem, but the specifics of their reduction are different. Also, if the revenues of all products are one, then their assortment optimization problem has a trivial optimal solution that offers all products. Our assortment optimization problem is NP-hard even when the revenues of all products are one.

Random Utility Maximization. In random utility maximization, a customer associates random utilities with the products and the no purchase option, choosing the alternative with the largest utility. We can justify our choice model by using random utility maximization. The utility of purchasing product i in stage k is U_i^k. For the vector of utilities $(U_i^1, \ldots, U_i^m)$ associated with product i in different stages, the marginal distribution of U_i^k is Gumbel with location and scale parameters $(\mu_i^k, 1)$, but the different components of the vector can be dependent. Through the dependence between the utilities associated with a product in different stages, we can capture the situation where if a customer favors a certain product in a certain stage, then she is likely to favor this product in other stages as well. The utility of not purchasing anything in stage k is U_0^k. For the vector of utilities $(U_0^1, \ldots, U_0^m)$ associated with the no purchase option in different stages, the marginal distribution of U_0^k is Gumbel with location and scale parameters $(\mu_0^k, 1)$, but the different components of the vector are independent. Not having any dependence between the utilities associated with the no purchase option in different stages yields a tractable expression for the choice probabilities and it is partially motivated by the fact that the no purchase options in different stages are very different since a customer has "less to lose" when she chooses the no purchase option in an earlier stage when there are other stages to follow. In any case, one can certainly and admittedly argue that if a customer favors the no purchase option in a certain stage, then she is also likely to favor this option in other stages. Also, the location parameter of the utility associated with the no purchase option in different stages can be different, once again, indicating that not purchasing anything in a later stage when the choice process is about to terminate can have a different utility implication than not purchasing anything in an earlier stage. For two products $i \neq j$, the vectors $(U_i^1, \ldots, U_i^m)$ and $(U_j^1, \ldots, U_j^m)$ are independent. Considering the independent random variables $\{X_i : i \in G\}$ for some generic index set G, if X_i has a Gumbel distribution with location and scale parameters $(\beta_i, 1)$, then

$\mathbb{P}\{X_i = \max_{j \in G} X_j\} = e^{\mu_i} / \sum_{j \in G} e^{\mu_j}$. If we offer the sets of products $(S^1, \ldots, S^m)$ over all stages, then for a customer to purchase product $i \in S^k$ in stage k, she needs to not make a purchase in stages $1, \ldots, k-1$ and she needs to purchase product i in stage k, in which case, a customer purchases product $i \in S^k$ in stage k with probability

$$\mathbb{P}\Big\{U_0^\ell = \max_{j \in S^\ell \cup \{0\}} U_j^\ell \;\; \forall\, \ell = 1, \ldots, k-1 \text{ and } U_i^k = \max_{j \in S^k \cup \{0\}} U_j^k\Big\} = \prod_{\ell=1}^{k-1} \frac{e^{\mu_0^\ell}}{\sum_{j \in S^\ell \cup \{0\}} e^{\mu_j^\ell}} \times \frac{e^{\mu_i^k}}{\sum_{j \in S^k \cup \{0\}} e^{\mu_j^k}},$$

where we use the fact that the sets of products offered in the different stages are disjoint so that the events on the left side above are independent. Letting $e^{\mu_i^k - \mu_0^k} = v_i^k$ and multiplying the numerator and denominator of all of the fractions on the right side above by $e^{-\mu_0^\ell}$ for $\ell = 1, \ldots, k$, we obtain the choice probability $\prod_{\ell=1}^{k-1} \frac{1}{1 + \sum_{j \in S^\ell} v_j^\ell} \times \frac{v_i^k}{1 + \sum_{j \in S^k} v_j^k}$.

In the rest of the chapter, noting Theorem 2.2.1, we focus on developing an FPTAS.

2.3 Multiple Knapsack Representation

Noting the objective function in (2.1), intuitively speaking, a good solution should keep the quantity $\sum_{i \in S^k} r_i v_i^k$ large and the quantity $\sum_{i \in S^k} v_i^k$ small for all $k \in M$. This observation motivates the following approach. First, we guess lower bounds on the quantity $\sum_{i \in S^k} r_i v_i^k$ and upper bounds on the quantity $\sum_{i \in S^k} v_i^k$ for all $k \in M$. Second, we check whether there exists a solution $(S^1, \ldots, S^m) \in \mathcal{F}$ that satisfies our guesses. Carrying out an exhaustive search over our guesses, we pick the best solution. To pursue this approach, we use $\mathcal{P}$ to denote the set of vectors $\boldsymbol{f} = (f^1, \ldots, f^m)$ and $\boldsymbol{h} = (h^1, \ldots, h^m)$ such that there exists a solution

$(S^1, \ldots, S^m) \in \mathcal{F}$ satisfying $\sum_{i \in S^k} r_i v_i^k \geq f^k$ and $\sum_{i \in S^k} v_i^k \leq h^k$ for all $k \in M$. Thus, $\mathcal{P}$ is given by

$$\mathcal{P} = \Big\{(\boldsymbol{f}, \boldsymbol{h}) \in \Re_+^m \times \Re_+^m \ : \ \exists (S^1, \ldots, S^m) \in \mathcal{F} \text{ that satisfies} \\ \sum_{i \in S^k} r_i v_i^k \geq f^k \ \ \forall k \in M \ \text{ and } \ \sum_{i \in S^k} v_i^k \leq h^k \ \ \forall k \in M\Big\}. \tag{2.2}$$

Note that the two sets of constraints that need to be satisfied by $(S^1, \ldots, S^m)$ above are similar to multiple knapsack constraints. Noting the objective function of problem (2.1), if we have $\sum_{i \in S^k} r_i v_i^k \geq f^k$ and $\sum_{i \in S^k} v_i^k \leq h^k$ for all $k \in M$, then the solution $(S^1, \ldots, S^m)$ provides an expected revenue of at least $\sum_{k \in M} \prod_{\ell=1}^k \frac{1}{1+f^\ell} \times h^k$. Therefore, we consider the problem

$$\max_{(\boldsymbol{f}, \boldsymbol{h}) \in \mathcal{P}} \left\{ \sum_{k \in M} \left\{ \prod_{\ell=1}^k \frac{1}{1 + h^\ell} \right\} f^k \right\}. \tag{2.3}$$

In the next lemma, we show that the optimal objective value of problem (2.1) corresponds to the optimal objective value of problem (2.3).

Lemma 2.3.1 *The optimal objective value of problem* (2.1) *is equal to the optimal objective value of problem* (2.3).

Proof. Let $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ be an optimal solution to problem (2.3) providing the optimal objective value $\widehat{\zeta}$. We have $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}}) \in \mathcal{P}$, in which case, by the definition of $\mathcal{P}$, there exists a solution $(\widehat{S}^1, \ldots, \widehat{S}^m) \in \mathcal{F}$ such that $\sum_{i \in \widehat{S}^k} r_i v_i^k \geq \widehat{f}^k$ and $\sum_{i \in \widehat{S}^k} v_i^k \leq \widehat{h}^k$ for all $k \in M$. Let $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$ be an optimal solution to problem (2.1) providing the optimal objective value $\widehat{Z}$. Define $\widetilde{f}^k = \sum_{i \in \widetilde{S}^k} r_i v_i^k$ and $\widetilde{h}^k = \sum_{i \in \widetilde{S}^k} v_i^k$ for all $k \in M$. Note that $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}}) \in \mathcal{P}$ since $\sum_{i \in \widetilde{S}^k} r_i v_i^k \geq \widetilde{f}^k$ and $\sum_{i \in \widetilde{S}^k} v_i^k \leq \widetilde{h}^k$ for all $k \in M$. Therefore, $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$ is a feasible solution to problem (2.3). Since $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ is an optimal solution to problem (2.3), we get $\sum_{k \in M} \prod_{\ell=1}^k \frac{1}{1+\widehat{h}^\ell} \times \widehat{f}^k = \widehat{\zeta} \geq \sum_{k \in M} \prod_{\ell=1}^k \frac{1}{1+\widetilde{h}^\ell} \times \widetilde{f}^k$. In this case, noting that we have $\sum_{i\in\widehat{S}^k} r_i\, v_i^k \ \geq\ \widehat{f}^k$ and $\sum_{i\in\widehat{S}^k} v_i^k \ \leq\ \widehat{h}^k$, the objective value provided by the solution $(\widehat{S}^1,\ldots,\widehat{S}^m)$ for problem (2.1) satisfies

$$\sum_{k\in M}\left\{\prod_{\ell=1}^{k}\frac{1}{1+\sum_{i\in\widehat{S}^\ell} v_i^\ell}\right\}\sum_{i\in\widehat{S}^k} r_i\, v_i^k \geq \sum_{k\in M}\left\{\prod_{\ell=1}^{k}\frac{1}{1+\widehat{h}^k}\right\}\widehat{f}^k$$
$$\geq \sum_{k\in M}\left\{\prod_{\ell=1}^{k}\frac{1}{1+\widetilde{h}^\ell}\right\}\widetilde{f}^k = \sum_{k\in M}\left\{\prod_{\ell=1}^{k}\frac{1}{1+\sum_{i\in\widetilde{S}^\ell} v_i^\ell}\right\}\sum_{i\in\widetilde{S}^k} r_i\, v_i^k = \widehat{Z}.$$

Since $(\widehat{S}^1,\ldots,\widehat{S}^m)\in\mathcal{F}$, the left side above is at most the optimal objective value of problem (2.1), which is $\widehat{Z}$. Thus, all of the inequalities above hold as equalities and we get $\widehat{Z}=\widehat{\zeta}$. ■

Using the lemma above, we can try to solve problem (2.1) in two steps. First, we find an optimal solution $(\widehat{\boldsymbol{f}},\widehat{\boldsymbol{h}})$ to problem (2.3). Second, we find $(\widehat{S}^1,\ldots,\widehat{S}^m)\ \in\ \mathcal{F}$ that satisfies $\sum_{i\in\widehat{S}^k} r_i\, v_i^k\ \geq\ \widehat{f}^k$ and $\sum_{i\in\widehat{S}^k} v_i^k\ \leq\ \widehat{h}^k$ for all $k\ \in\ M$. By the discussion right before problem (2.3), the expected revenue from the solution $(\widehat{S}^1,\ldots,\widehat{S}^m)$ is at least $\sum_{k\in M}\prod_{\ell=1}^{k}\frac{1}{1+\widehat{f}^\ell}\times\widehat{h}^k$, which is, by Lemma 2.3.1, equal to the optimal objective value of problem (2.1). Both of these two steps are computationally difficult. In particular, the objective function of problem (2.3) is not necessarily concave and finding $(\widehat{S}^1,\ldots,\widehat{S}^m)$ satisfying the last two inequalities is a combinatorial problem. In our FPTAS, we carry out these two steps approximately. First, we use a geometric grid over $\Re_+^m\times\Re_+^m$ to check the objective value of problem (2.3) at a limited number of guesses for $(\boldsymbol{f},\boldsymbol{h})$. Second, we give an approximate version of the set $\mathcal{P}$, in which case, we can use a dynamic program to find $(\widehat{S}^1,\ldots,\widehat{S}^m)\ \in\ \mathcal{F}$ that approximately satisfies $\sum_{i\in\widehat{S}^k} r_i\, v_i^k\ \geq\ \widehat{f}^k$ and $\sum_{i\in\widehat{S}^k} v_i^k\ \leq\ \widehat{h}^k$ for all $k\in M$. In the next section, we construct the geometric grid and the approximate version of the set $\mathcal{P}$.

2.4 Approximate Feasible Set

To give an approximation to the set $\mathcal{P}$, we begin by computing $\sum_{i \in S^k} r_i v_i^k$ and $\sum_{i \in S^k} v_i^k$ recursively. For $f^k \in \Re_+$, $h^k \in \Re_+$ and $S^k \subseteq N$, we define $F_i^k(f^k, S^k)$ and $H_i^k(h^k, S^k)$ recursively as

$$F_{i+1}^k(f^k, S^k) = F_i^k(f^k, S^k) - r_i v_i^k \mathbf{1}(i \in S^k) \qquad (2.4)$$
$$H_{i+1}^k(h^k, S^k) = H_i^k(h^k, S^k) - v_i^k \mathbf{1}(i \in S^k)$$

with the initial condition that $F_1^k(f^k, S^k) = f^k$ and $H_1^k(h^k, S^k) = h^k$. Adding the first equality above over all $i \in N$ and noting that $F_1^k(f^k, S^k) = f^k$, we obtain $F_{n+1}^k(f^k, S^k) = f^k - \sum_{i \in N} r_i v_i^k \mathbf{1}(i \in S^k) = f^k - \sum_{i \in S^k} r_i v_i^k$. Therefore, we have $\sum_{i \in S^k} r_i v_i^k \geq f^k$ if and only if we have $F_{n+1}^k(f^k, S^k) \leq 0$. Similarly, we have $\sum_{i \in S^k} v_i^k \leq h^k$ if and only if we have $H_{n+1}^k(h^k, S^k) \geq 0$. In this case, we can replace the condition $\sum_{i \in S^k} r_i v_i^k \geq f^k$ and $\sum_{i \in S^k} v_i^k \leq h^k$ for all $k \in M$ in (2.2) with the condition $F_{n+1}^k(f^k, S^k) \leq 0$ and $H_{n+1}^k(h^k, S^k) \geq 0$ for all $k \in M$ to express the set $\mathcal{P}$ equivalently. In other words, the set $\mathcal{P}$ is also given by

$$\mathcal{P} = \Big\{ (\boldsymbol{f}, \boldsymbol{h}) \in \Re_+^m \times \Re_+^m \, : \, \exists (S^1, \ldots, S^m) \in \mathcal{F} \text{ that satisfies}$$
$$F_{n+1}^k(f^k, S^k) \leq 0 \;\; \forall\, k \in M \text{ and } H_{n+1}^k(h^k, S^k) \geq 0 \;\; \forall\, k \in M \Big\}. \qquad (2.5)$$

By restricting the values of $F_i^k(f^k, S^k)$ and $H_i^k(h^k, S^k)$ on a geometric grid, we proceed to giving an approximate version of the set $\mathcal{P}$.

For fixed $\rho > 0$, we define $\text{DOM} = \{(1+\rho)^\ell : \ell = \ldots, -1, 0, 1, \ldots\} \cup \{-\infty, 0\}$, which is a geometric grid augmented by the points $\{-\infty, 0\}$. We define the round up operator $\lceil \cdot \rceil$ that rounds its argument up to the nearest element of DOM. In particular, $\lceil x \rceil = \min\{y \in \text{DOM} : y \geq x\}$. Similarly, we define the round down operator $\lfloor \cdot \rfloor$ that rounds its argument down to the nearest element of DOM.

Therefore, we have $\lfloor x \rfloor = \max\{y \in \text{DOM} : y \leq x\}$. Note that if $x < 0$, then $\lceil x \rceil = 0$ and $\lfloor x \rfloor = -\infty$. We use $\Phi_i^k(f^k, S^k)$ and $\Gamma_i^k(h^k, S^k)$ to denote approximate versions of $F_i^k(f^k, S^k)$ and $H_i^k(h^k, S^k)$, which are also defined recursively as

$$\Phi_{i+1}^k(f^k, S^k) = \lceil \Phi_i^k(f^k, S^k) - r_i\, v_i^k\, \mathbf{1}(i \in S^k) \rceil \tag{2.6}$$
$$\Gamma_{i+1}^k(h^k, S^k) = \lfloor \Gamma_i^k(h^k, S^k) - v_i^k\, \mathbf{1}(i \in S^k) \rfloor$$

with the initial condition that $\Phi_1^k(f^k, S^k) = f^k$ and $\Gamma_1^k(h^k, S^k) = h^k$. Noting the round up operator in the definition of $\Phi_{i+1}^k(f^k, S^k)$, we get $\Phi_{i+1}^k(f^k, S^k) \geq \Phi_i^k(f^k, S^k) - r_i\, v_i^k\, \mathbf{1}(i \in S^k)$. If we add this inequality over all $i \in N$ and note that $\Phi_1^k(f^k, S^k) = f^k$, then we obtain $\Phi_{n+1}^k(f^k, S^k) \geq f^k - \sum_{i \in N} r_i\, v_i^k\, \mathbf{1}(i \in S^k) = f^k - \sum_{i \in S^k} r_i\, v_i^k$. Therefore, if $\Phi_{n+1}^k(f^k, S^k) \leq 0$, then we have $f^k - \sum_{i \in S^k} r_i\, v_i^k \leq 0$, which implies that $F_{n+1}^k(f^k, S^k) \leq 0$ as well. Using a similar argument, we also obtain $\Gamma_{n+1}^k(h^k, S^k) \leq h^k - \sum_{i \in S^k} v_i^k$. In this case, if $\Gamma_{n+1}^k(h^k, S^k) \geq 0$, then we have $h^k - \sum_{i \in S^k} v_i^k \geq 0$, which implies that $H_{n+1}^k(h^k, S^k) \geq 0$ as well.

By the discussion in the previous paragraph, if the vector $(\boldsymbol{f}, \boldsymbol{h}) \in \Re_+^m \times \Re_+^m$ satisfies $\Phi_{n+1}^k(f^k, S^k) \leq 0$ and $\Gamma_{n+1}^k(h^k, S^k) \geq 0$ for all $k \in M$, then it also satisfies $F_{n+1}^k(f^k, S^k) \leq 0$ and $H_{n+1}^k(h^k, S^k) \geq 0$ for all $k \in M$. Thus, we can define a restricted version of the set $\mathcal{P}$ using $\Phi_{n+1}^k(f^k, S^k)$ and $\Gamma_{n+1}^k(h^k, S^k)$. Denoting this restricted version by $\widetilde{\mathcal{P}}$, we have

$$\widetilde{\mathcal{P}} = \Big\{ (\boldsymbol{f}, \boldsymbol{h}) \in \text{DOM}_+^m \times \text{DOM}_+^m \;:\; \exists\, (S^1, \ldots, S^m) \in \mathcal{F} \text{ that satisfies}$$
$$\Phi_{n+1}^k(f^k, S^k) \leq 0 \;\; \forall\, k \in M \;\text{ and }\; \Gamma_{n+1}^k(h^k, S^k) \geq 0 \;\; \forall\, k \in M \Big\}, \tag{2.7}$$

where we use $\text{DOM}_+ = \text{DOM} \setminus \{-\infty\}$. By the discussion above, $\widetilde{\mathcal{P}} \subseteq \mathcal{P}$. In the next proposition, we show that we can perturb an element of $\mathcal{P}$ to obtain an element of $\widetilde{\mathcal{P}}$.

Proposition 2.4.1 *For any $f^k \in \text{DOM}_+$, $h^k \in \text{DOM}_+$ and $S^k \subseteq N$, if $F^k_{n+1}(f^k, S^k) \leq 0$, then we have $\Phi^k_{n+1}(f^k/(1+\rho)^{|S^k|}, S^k) \leq 0$. Also, if $H^k_{n+1}(h^k, S^k) \geq 0$, then we have $\Gamma^k_{n+1}((1+\rho)^{|S^k|} h^k, S^k) \geq 0$.*

Proof. Fix f^k and S^k. For notational brevity, we let $\Phi^k_i = \Phi^k_i(f^k/(1+\rho)^{|S^k|}, S^k)$ and $S^k_i = S^k \cap \{i, \ldots, n\}$. We follow the convention that $S^k_{n+1} = \emptyset$. By the definition of $\Phi^k_i(f^k/(1+\rho)^{|S^k|}, S^k)$, we have $\Phi^k_{i+1} = \lceil \Phi^k_i - r_i\, v^k_i\, \mathbf{1}(i \in S^k) \rceil$ with $\Phi^k_1 = f^k/(1+\rho)^{|S^k|}$. Noting the definition of S^k_i, we have $i \in S^k$ if and only if $i \in S^k_i$. Also, we have $S^k_1 = S^k$. Therefore, we can write the recursion that we use to compute Φ^k_i equivalently as $\Phi^k_{i+1} = \lceil \Phi^k_i - r_i\, v^k_i\, \mathbf{1}(i \in S^k_i) \rceil$ with $\Phi^k_1 = f^k/(1+\rho)^{|S^k_1|}$. We use induction over the products to show that $\Phi^k_i \leq \sum_{j \in S^k_i} r_j\, v^k_j/(1+\rho)^{|S^k_i|}$ or $\Phi^k_i = 0$ for all $i \in N \cup \{n+1\}$. Since $F^k_{n+1}(f^k, S^k) \leq 0$, by the discussion at the beginning of this section, we have $\sum_{j \in S^k} r_j\, v^k_j \geq f^k$. In this case, noting that $\Phi^k_1 = f^k/(1+\rho)^{|S^k_1|}$, we obtain $\Phi^k_1 = f^k/(1+\rho)^{|S^k_1|} \leq \sum_{j \in S^k_1} r_j\, v^k_j/(1+\rho)^{|S^k_1|}$, which implies that the result holds for product 1. Next, we assume that the result holds for product i, so that $\Phi^k_i \leq \sum_{j \in S^k_i} r_j\, v^k_j/(1+\rho)^{|S^k_i|}$ or $\Phi^k_i = 0$. If $\Phi^k_i = 0$, then $\Phi^k_i - r_i\, v^k_i\, \mathbf{1}(i \in S^k_i) \leq 0$, which implies that $\Phi^k_{i+1} = \lceil \Phi^k_i - r_i\, v^k_i\, \mathbf{1}(i \in S^k_i) \rceil = 0$, in which case, the result holds for product $i+1$ as well. Thus, we assume that $\Phi^k_i \leq \sum_{j \in S^k_i} r_j\, v^k_j/(1+\rho)^{|S^k_i|}$ in the rest of the induction argument. Note that if $\Phi^k_i \leq \sum_{j \in S^k_i} r_j\, v^k_j/(1+\rho)^{|S^k_i|}$, then we have

$$
\begin{aligned}
\Phi^k_i - \mathbf{1}(i \in S^k_i)\, r_i\, v^k_i &\leq \frac{\sum_{j \in S^k_i} r_j\, v^k_j}{(1+\rho)^{|S^k_i|}} - \mathbf{1}(i \in S^k_i)\, r_i\, v^k_i \\
&= \frac{\sum_{j \in S^k_{i+1}} r_j\, v^k_j + \mathbf{1}(i \in S^k_i)\, r_i\, v^k_i}{(1+\rho)^{|S^k_{i+1}| + \mathbf{1}(i \in S^k_i)}} - \mathbf{1}(i \in S^k_i)\, r_i\, v^k_i \leq \frac{\sum_{j \in S^k_{i+1}} r_j\, v^k_j}{(1+\rho)^{|S^k_{i+1}| + \mathbf{1}(i \in S^k_i)}}, \qquad (2.8)
\end{aligned}
$$

where the equality uses the fact that if $i \in S^k_i$, then $S^k_i = S^k_{i+1} \cup \{i\}$ and the second inequality uses the fact that $1/(1+\rho)^{|S^k_{i+1}| + \mathbf{1}(i \in S^k_i)} \leq 1$.

If $\Phi^k_i - \mathbf{1}(i \in S^k_i)\, r_i\, v^k_i \leq 0$, then $\Phi^k_{i+1} = \lceil \Phi^k_i - \mathbf{1}(i \in S^k_i)\, r_i\, v^k_i \rceil = 0$ and the result holds for product $i+1$ as well. So, we consider the chain of inequalities in

(2.8) under the assumption that $\Phi_i^k - \mathbf{1}(i \in S_i^k)\, r_i\, v_i^k > 0$. First, we consider the case $i \notin S_i^k$. By (2.8), we obtain $\Phi_i^k \leq \sum_{j \in S_{i+1}^k} r_j\, v_j^k/(1+\rho)^{|S_{i+1}^k|}$. Furthermore, we have $\Phi_{i+1}^k = \lceil \Phi_i^k - r_i\, v_i^k\, \mathbf{1}(i \in S_i^k) \rceil = \lceil \Phi_i^k \rceil$. Lastly, since $f^k \in \textsc{Dom}_+$, we get $\Phi_1^k = f^k/(1+\rho)^{|S_1^k|} \in \textsc{Dom}_+$. Since $\Phi_{j+1}^k = \lceil \Phi_j^k - r_j\, v_i^k\, \mathbf{1}(j \in S_i^k) \rceil$ for all $j \in N$ and $\Phi_1^k \in \textsc{Dom}_+$, we obtain $\Phi_j^k \in \textsc{Dom}$ for all $j \in N$, so that $\lceil \Phi_j^k \rceil = \Phi_j^k$. Therefore, we have $\Phi_{i+1}^k = \lceil \Phi_i^k \rceil = \Phi_i^k \leq \sum_{j \in S_{i+1}^k} r_j\, v_j^k/(1+\rho)^{|S_{i+1}^k|}$, in which case, the result holds for product $i+1$ as well. Second, we consider the case $i \in S_i^k$. By (2.8), we have $\Phi_i^k - \mathbf{1}(i \in S_i^k)\, r_i\, v_i^k \leq \sum_{j \in S_{i+1}^k} r_j\, v_j^k/(1+\rho)^{|S_{i+1}^k|+1}$. For $x \geq 0$, note that $\lceil x \rceil \leq (1+\rho)\, x$. In this case, since we assume that $\Phi_i^k - \mathbf{1}(i \in S_i^k)\, r_i\, v_i^k > 0$, by the last inequality, we obtain $\Phi_{i+1}^k = \lceil \Phi_i^k - \mathbf{1}(i \in S_i^k)\, r_i\, v_i^k \rceil \leq (1+\rho) \times (\Phi_i^k - \mathbf{1}(i \in S_i^k)\, r_i\, v_i^k) \leq \sum_{j \in S_{i+1}^k} r_j\, v_j^k/(1+\rho)^{|S_{i+1}^k|}$, which implies that the result holds for product $i+1$ as well, completing the induction argument. Therefore, the discussion so far establishes that $\Phi_i^k \leq \sum_{j \in S_i^k} r_j\, v_j^k/(1+\rho)^{|S_i^k|}$ or $\Phi_i^k = 0$ for all $i \in N \cup \{n+1\}$. Using this result with $i = n+1$ and noting that $S_{n+1}^k = \emptyset$, we get $\Phi_{n+1}^k \leq 0$ or $\Phi_{n+1}^k = 0$. So, $\Phi_{n+1}^k = \Phi_{n+1}^k(f^k/(1+\rho)^{|S^k|}, S^k) \leq 0$, showing the first statement in the proposition. The second statement uses a similar reasoning. ■

By the proposition above, if $\rho > 0$ is small, then we can perturb an element of $\mathcal{P}$ by a small amount to obtain an element of $\widetilde{\mathcal{P}}$. Noting the discussion at the end of Section 2.3, we can solve problem (2.1) by obtaining an optimal solution $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ to problem (2.3) and finding $(\widehat{S}^1, \ldots, \widehat{S}^m) \in \mathcal{F}$ such that $F_{n+1}^k(\widehat{f}^k, \widehat{S}^k) \leq 0$ and $H_{n+1}^k(\widehat{f}^k, \widehat{S}^k) \geq 0$ for all $k \in M$. Replacing $\mathcal{P}$ with $\widetilde{\mathcal{P}}$, $F_{n+1}^k(f^k, S^k)$ with $\Phi_{n+1}^k(f^k, S^k)$ and $H_{n+1}^k(h^k, S^k)$ with $\Gamma_{n+1}^k(f^k, S^k)$, we can approximately solve problem (2.1) by obtaining an optimal solution $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$ to the problem $\max_{(\boldsymbol{f},\boldsymbol{h}) \in \widetilde{\mathcal{P}}} \left\{ \sum_{k \in M} \prod_{\ell=1}^k \frac{1}{1+h^\ell} \times f^k \right\}$ and finding $(\widetilde{S}^1, \ldots, \widetilde{S}^m) \in \mathcal{F}$ such that $\Phi_{n+1}^k(\widetilde{f}^k, \widetilde{S}^k) \leq 0$ and $\Gamma_{n+1}^k(\widetilde{h}^k, \widetilde{S}^k) \geq 0$ for all $k \in M$. In the next section, we give a performance guarantee for this approach. Proposition 2.4.1 plays an important role in coming up with this performance guarantee.

2.5 Performance Guarantee

To obtain a solution to problem (2.1) with a performance guarantee, we propose the following algorithm, referred to as APPROX.

Step 1. Solve the problem $\max_{(\boldsymbol{f},\boldsymbol{h})\in\widetilde{\mathcal{P}}} \left\{ \sum_{k\in M} \prod_{\ell=1}^{k} \frac{1}{1+h^\ell} \times f^k \right\}$ and denote an optimal solution to this problem by $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$.

Step 2. Since $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}}) \in \widetilde{\mathcal{P}}$, there exists a solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m) \in \mathcal{F}$ such that $\Phi_{n+1}^k(\widetilde{f}^k, \widetilde{S}^k) \leq 0$ and $\Gamma_{n+1}^k(\widetilde{f}^k, \widetilde{S}^k) \geq 0$ for all $k \in M$. Return one such solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$.

In this section, we give a performance guarantee for the solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$ provided by the APPROX algorithm. In the next section, we give an approach that allows us to execute the APPROX algorithm efficiently when the number of stages is fixed. Putting these two results together yields an FPTAS for problem (2.1). We proceed to giving a performance guarantee for the solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$ provided by the APPROX algorithm. We let $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}}) \in \mathcal{P}$ be an optimal solution to problem (2.3). By Lemma 2.3.1, the optimal objective value of problem (2.3) is equal to the optimal objective value of problem (2.1). Since $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}}) \in \mathcal{P}$, by the alternative definition of $\mathcal{P}$ in (2.5), there exists $(\widehat{S}^1, \ldots, \widehat{S}^m) \in \mathcal{F}$ such that $F_{n+1}^k(\widehat{f}^k, \widehat{S}^k) \leq 0$ and $H_{n+1}^k(\widehat{h}^k, \widehat{S}^k) \geq 0$ for all $k \in M$. We define $(\overline{\boldsymbol{f}}, \overline{\boldsymbol{h}})$ as $\overline{f}^k = \lfloor \widehat{f}^k \rfloor / (1+\rho)^{|\widehat{S}^k|}$ and $\hbar^k = (1+\rho)^{|\widehat{S}^k|} \lceil \widehat{h}^k \rceil$ for all $k \in M$.

In the next lemma, we show that $(\overline{\boldsymbol{f}}, \overline{\boldsymbol{h}})$ is feasible to the problem in Step 1 above.

Lemma 2.5.1 *For any* $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}}) \in \Re_+^m \times \Re_+^m$ *and* $(\widehat{S}^1, \ldots, \widehat{S}^m) \in \mathcal{F}$ *that satisfies* $F_{n+1}^k(\widehat{f}^k, \widehat{S}^k) \leq 0$ *and* $H_{n+1}^k(\widehat{h}^k, \widehat{S}^k) \geq 0$ *for all* $k \in M$*, let* $(\overline{\boldsymbol{f}}, \overline{\boldsymbol{h}})$ *be such that* $\overline{f}^k = \lfloor \widehat{f}^k \rfloor / (1+\rho)^{|\widehat{S}^k|}$ *and* $\hbar^k = (1+\rho)^{|\widehat{S}^k|} \lceil \widehat{h}^k \rceil$ *for all* $k \in M$*. Then, we have* $(\overline{\boldsymbol{f}}, \overline{\boldsymbol{h}}) \in \widetilde{\mathcal{P}}$*.*

Proof. Adding the equality in (2.4) over all $i \in N$ and noting that $F_1^k(f^k, S^k) = f^k$, we obtain $F_{n+1}^k(f^k, S^k) = f^k - \sum_{i \in S^k} r_i v_i^k$, which implies that $F_{n+1}^k(f^k, S^k)$ is increasing in f^k. In this case, noting that $F_{n+1}^k(\widehat{f}^k, \widehat{S}^k) \leq 0$ and $\lfloor f^k \rfloor \leq f^k$, we obtain $F_{n+1}^k(\lfloor \widehat{f}^k \rfloor, \widehat{S}^k) \leq F_{n+1}^k(\widehat{f}^k, \widehat{S}^k) \leq 0$ as well. Therefore, we have $F_{n+1}^k(\lfloor \widehat{f}^k \rfloor, \widehat{S}^k) \leq 0$, in which case, using the fact that $\lfloor \widehat{f}^k \rfloor \in \text{DOM}_+$, by Proposition 2.4.1, we obtain $\Phi_{n+1}^k(\overline{f}^k, \widehat{S}^k) = \Phi_{n+1}^k(\lfloor \widehat{f}^k \rfloor / (1+\rho)^{|\widehat{S}^k|}, \widehat{S}^k) \leq 0$. Using a similar reasoning, we also have $\Gamma_{n+1}^k(\hbar^k, \widehat{S}^k) = \Gamma_{n+1}^k((1+\rho)^{|\widehat{S}^k|} \lceil \widehat{h}^k \rceil, \widehat{S}^k) \geq 0$. In this case, there exists $(\widehat{S}^1, \ldots, \widehat{S}^m) \in \mathcal{F}$ such that $\Phi_{n+1}^k(\overline{f}^k, \widehat{S}^k) \leq 0$ and $\Gamma_{n+1}^k(\hbar^k, \widehat{S}^k) \geq 0$ for all $k \in M$, so that noting the definition of $\widetilde{\mathcal{P}}$ in (2.7), we have $(\overline{\boldsymbol{f}}, \overline{\boldsymbol{h}}) \in \widetilde{\mathcal{P}}$. ■

When $\rho > 0$ is close to zero, $(1+\rho)^{|S^k|}$ is close to one. Thus, by Lemma 2.5.1, we can scale any solution $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}}) \in \mathcal{P}$ by a factor close to one to obtain a solution $(\overline{\boldsymbol{f}}, \overline{\boldsymbol{h}}) \in \widetilde{\mathcal{P}}$, as long as $\rho > 0$ is small. In other words, given a solution $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}}) \in \mathcal{P}$, which is optimal to problem (2.3), we can scale this solution by a factor close to one to obtain a solution $(\overline{\boldsymbol{f}}, \overline{\boldsymbol{h}}) \in \widetilde{\mathcal{P}}$, which is feasible to the problem in Step 1 of the APPROX algorithm. In the next theorem, we use this observation to give a performance guarantee for the solution provided by the APPROX algorithm. In this theorem and throughout the rest of the chapter, we use $\text{REV}(S^1, \ldots, S^m)$ to denote the objective function of problem (2.1), which is the expected revenue from the solution $(S^1, \ldots, S^m)$.

Theorem 2.5.2 *Letting* $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$ *be the output of the* APPROX *algorithm and* $\widehat{Z}$ *be the optimal objective value of problem* (2.1)*, we have* $\text{REV}(\widetilde{S}^1, \ldots, \widetilde{S}^m) \geq \widehat{Z}/(1+\rho)^{3n+1}$*.*

Proof. We let $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$ be an optimal solution to the problem in Step 1 of the Approx algorithm. By (2.6), we have $\Phi^k_{i+1}(\widetilde{f}^k, \widetilde{S}^k) \geq \Phi^k_i(\widetilde{f}^k, \widetilde{S}^k) - r_i\, v^k_i\, \mathbf{1}(i \in \widetilde{S}^k)$ for all $i \in N$, $k \in M$, in which case, adding this inequality over all $i \in N$ and noting that $\Phi^k_1(\widetilde{f}^k, \widetilde{S}^k) = \widetilde{f}^k$, we obtain $\Phi^k_{n+1}(\widetilde{f}^k, \widetilde{S}^k) \geq \widetilde{f}^k - \sum_{i \in N} r_i\, v^k_i\, \mathbf{1}(i \in \widetilde{S}^k) = \widetilde{f}^k - \sum_{i \in \widetilde{S}^k} r_i\, v^k_i$ for all $k \in M$. Furthermore, by the definition of $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$ in Step 2 of the Approx algorithm, we also have $\Phi^k_{n+1}(\widetilde{f}^k, \widetilde{S}^k) \leq 0$ for all $k \in M$. In this case, we obtain $\widetilde{f}^k - \sum_{i \in \widetilde{S}^k} r_i\, v^k_i \leq \Phi^k_{n+1}(\widetilde{f}^k, \widetilde{S}^k) \leq 0$ so that $\sum_{i \in \widetilde{S}^k} r_i\, v^k_i \geq \widetilde{f}^k$ for all $k \in M$. Using a similar reasoning, we have $\sum_{i \in \widetilde{S}^k} v^k_i \leq \widetilde{h}^k$ for all $k \in M$ as well. Therefore, the expected revenue from the solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$ satisfies

$$\textsc{Rev}(\widetilde{S}^1, \ldots, \widetilde{S}^m) = \sum_{k \in M} \left\{ \prod_{\ell=1}^{k} \frac{1}{1 + \sum_{i \in \widetilde{S}^\ell} v^\ell_i} \right\} \sum_{i \in \widetilde{S}^k} r_i\, v^k_i \geq \sum_{k \in M} \left\{ \prod_{\ell=1}^{k} \frac{1}{1 + \widetilde{h}^\ell} \right\} \widetilde{f}^k, \qquad (2.9)$$

which shows that the optimal objective value of the problem in Step 1 of the Approx algorithm is a lower bound on the expected revenue from the solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$.

Next, we construct a lower bound on the optimal objective value of the problem in Step 1 of the Approx algorithm by giving a feasible solution to this problem. Using $(\widehat{S}^1, \ldots, \widehat{S}^m)$ to denote an optimal solution to problem (2.1), we let $\widehat{f}^k = \sum_{i \in \widehat{S}^k} r_i\, v^k_i$ and $\widehat{h}^k = \sum_{i \in \widehat{S}^k} v^k_i$ for all $k \in M$. By the discussion that follows the definition of $F^k_i(f^k, S^k)$ in (2.4), we have $F^k_{n+1}(f^k, S^k) = f^k - \sum_{i \in S^k} r_i\, v^k_i$. Thus, we have $F^k_{n+1}(\widehat{f}^k, \widehat{S}^k) = \widehat{f}^k - \sum_{i \in \widehat{S}^k} r_i\, v^k_i = 0$ for all $k \in M$. Using a similar reasoning, we also have $H^k_{n+1}(\widehat{h}^k, \widehat{S}^k) = 0$ for all $k \in M$. In this case, letting $\overline{f}^k = \lfloor \widehat{f}^k \rfloor / (1+\rho)^{|\widehat{S}^k|}$ and $\hbar^k = (1+\rho)^{|\widehat{S}^k|} \lceil \widehat{h}^k \rceil$ for all $k \in M$, by Lemma 2.5.1, we obtain $(\overline{\boldsymbol{f}}, \overline{\boldsymbol{h}}) \in \widetilde{\mathcal{P}}$, which implies that $(\overline{\boldsymbol{f}}, \overline{\boldsymbol{h}})$ is a feasible solution to the problem in Step 1 of the Approx algorithm. Noting that $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$ is an optimal solution to this problem, we obtain

$$\sum_{k\in M}\left\{\prod_{\ell=1}^{k}\frac{1}{1+\widetilde{h}^\ell}\right\}\widetilde{f}^k \geq \sum_{k\in M}\left\{\prod_{\ell=1}^{k}\frac{1}{1+\bar{h}^\ell}\right\}\bar{f}^k$$
$$= \sum_{k\in M}\left\{\prod_{\ell=1}^{k}\frac{1}{1+(1+\rho)^{|\widehat{S}^\ell|}\lceil\widehat{h}^\ell\rceil}\right\}\frac{\lfloor\widehat{f}^k\rfloor}{(1+\rho)^{|\widehat{S}^k|}}$$
$$\geq \sum_{k\in M}\left\{\prod_{\ell=1}^{k}\frac{1}{1+(1+\rho)^{|\widehat{S}^\ell|+\mathbf{1}(\widehat{h}^\ell>0)}\,\widehat{h}^\ell}\right\}\frac{\widehat{f}^k}{(1+\rho)^{|\widehat{S}^k|+1}}, \quad (2.10)$$

where the last equality is by the fact that $\lceil x\rceil \leq (1+\rho)^{\mathbf{1}(x>0)}\,x$ and $\lfloor x\rfloor \geq x/(1+\rho)$ for any $x \in \Re_+$. Since $\widehat{h}^k = \sum_{i\in\widehat{S}^k} v_i^k$, we have $\widehat{h}^k > 0$ if and only if $\widehat{S}^k \neq \varnothing$. Therefore, we obtain

$$\sum_{k\in M}\left\{\prod_{\ell=1}^{k}\frac{1}{1+(1+\rho)^{|\widehat{S}^\ell|+\mathbf{1}(\widehat{h}^\ell>0)}\,\widehat{h}^\ell}\right\}\frac{\widehat{f}^k}{(1+\rho)^{|\widehat{S}^k|+1}}$$
$$= \sum_{k\in M}\left\{\prod_{\ell=1}^{k}\frac{1}{1+(1+\rho)^{|\widehat{S}^\ell|+\mathbf{1}(\widehat{S}^\ell\neq\varnothing)}\,\widehat{h}^\ell}\right\}\frac{\widehat{f}^k}{(1+\rho)^{|\widehat{S}^k|+1}}$$
$$\geq \sum_{k\in M}\left\{\prod_{\ell=1}^{k}\frac{1}{(1+\rho)^{|\widehat{S}^\ell|+\mathbf{1}(\widehat{S}^\ell\neq\varnothing)}\,(1+\widehat{h}^\ell)}\right\}\frac{\widehat{f}^k}{(1+\rho)^{|\widehat{S}^k|+1}}. \quad (2.11)$$

Since $(\widehat{S}^1,\ldots,\widehat{S}^m) \in \mathcal{F}$, the sets $\widehat{S}^1,\ldots,\widehat{S}^m$ are disjoint. Therefore, we have $\sum_{\ell=1}^{k}|\widehat{S}^\ell| \leq n$ and $\sum_{\ell=1}^{k}\mathbf{1}(\widehat{S}^\ell \neq \varnothing) \leq n$ for all $k \in M$. Also, $|\widehat{S}^k| \leq n$. So, we have

$$\sum_{k\in M}\left\{\prod_{\ell=1}^{k}\frac{1}{(1+\rho)^{|\widehat{S}^\ell|+\mathbf{1}(\widehat{S}^\ell\neq\varnothing)}\,(1+\widehat{h}^\ell)}\right\}\frac{\widehat{f}^k}{(1+\rho)^{|\widehat{S}^k|+1}}$$
$$\geq \sum_{k\in M}\frac{1}{(1+\rho)^{2n}}\left\{\prod_{\ell=1}^{k}\frac{1}{1+\widehat{h}^\ell}\right\}\frac{\widehat{f}^k}{(1+\rho)^{n+1}}$$
$$= \sum_{k\in M}\left\{\prod_{\ell=1}^{k}\frac{1}{1+\sum_{i\in\widehat{S}^\ell} v_i^\ell}\right\}\frac{\sum_{i\in\widehat{S}^k} r_i\, v_i^k}{(1+\rho)^{3n+1}} = \frac{\widehat{Z}}{(1+\rho)^{3n+1}}, \quad (2.12)$$

where the first equality is by the definition of $(\widehat{f}^k, \widehat{h}^k)$ and the second equality holds as $(\widehat{S}^1,\ldots,\widehat{S}^m)$ is an optimal solution to problem (2.1). The desired result follows by (2.9), (2.10), (2.11) and (2.12). ∎

Thus, the expected revenue from the solution provided by the APPROX algorithm deviates from the optimal expected revenue by no more than a factor of $(1+\rho)^{3n+1}$. For any $\epsilon \in (0,1)$, consider executing the APPROX algorithm with $\rho = \epsilon/(8n)$. Since $\epsilon < 1$, we have $(1+\rho)^{3n+1} \le (1+\rho)^{4n} = (1+\frac{\epsilon}{8n})^{4n} \le \exp(\epsilon/2) \le 1+\epsilon$, so that $\text{Rev}(\widetilde{S}^1, \ldots, \widetilde{S}^m) \ge \widehat{Z}/(1+\rho)^{3n+1} \ge \widehat{Z}/(1+\epsilon) \ge (1-\epsilon)\widehat{Z}$. Thus, the expected revenue from the solution provided by the APPROX algorithm is at least $1-\epsilon$ fraction of the optimal expected revenue. Although the solution provided by the APPROX algorithm has a performance guarantee, it is not yet clear that we can execute the APPROX algorithm efficiently. In the next section, we give a dynamic program that allows us to execute the APPROX algorithm efficiently when the number of stages is fixed. By accounting for the number of operations to solve the dynamic program, we ultimately obtain our FPTAS.

2.6 Dynamic Programming Formulation

To execute the APPROX algorithm efficiently, we make use of two observations. First, we can use a dynamic program to check whether a given value of $(\boldsymbol{f}, \boldsymbol{h}) \in \text{Dom}_+^m \times \text{Dom}_+^m$ satisfies $(\boldsymbol{f}, \boldsymbol{h}) \in \widetilde{\mathcal{P}}$, allowing us to check the feasibility of a solution to the problem in Step 1 of the APPROX algorithm. Second, we can bound the components of an optimal solution to the problem in Step 1 of the APPROX algorithm. Using the bound, the number of values of $(\boldsymbol{f}, \boldsymbol{h}) \in \text{Dom}_+^m \times \text{Dom}_+^m$ that can possibly be an optimal solution becomes polynomial in input size, when the number of stages is fixed. In this case, we can execute Step 1 of the APPROX algorithm by checking whether each value of $(\boldsymbol{f}, \boldsymbol{h}) \in \text{Dom}_+^m \times \text{Dom}_+^m$ that can be possibly be an optimal solution to the problem in this step satisfies $(\boldsymbol{f}, \boldsymbol{h}) \in \widetilde{\mathcal{P}}$ and by picking one that provides the best expected revenue. In Step 2 of the APPROX algorithm, we need to find a solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m) \in \mathcal{F}$ such that $\Phi^k_{n+1}(\widetilde{f}^k, \widetilde{S}^k) \leq 0$ and $\Gamma^k_{n+1}(\widetilde{f}^k, \widetilde{S}^k) \geq 0$ for all $k \in M$. Noting (2.7), checking whether a given value of $(\boldsymbol{f}, \boldsymbol{h}) \in \text{DOM}^m_+ \times \text{DOM}^m_+$ satisfies $(\boldsymbol{f}, \boldsymbol{h}) \in \widetilde{\mathcal{P}}$ requires finding a solution $(S^1, \ldots, S^m) \in \mathcal{F}$ such that $\Phi^k_{n+1}(f^k, S^k) \leq 0$ and $\Gamma^k_{n+1}(f^k, S^k) \geq 0$ for all $k \in M$. Therefore, we can use the dynamic program that we use in Step 1 of the APPROX algorithm to execute Step 2 as well.

We proceed to giving a dynamic program that allows us to check whether a given value of $(\boldsymbol{f}, \boldsymbol{h}) \in \text{DOM}^m_+ \times \text{DOM}^m_+$ is feasible to the problem in Step 1 of the APPROX algorithm. Consider a fixed value of $(\boldsymbol{f}, \boldsymbol{h}) \in \text{DOM}^m_+ \times \text{DOM}^m_+$. Noting (2.6), the values of $\Phi^k_i(f^k, S^k)$ and $\Gamma^k_i(h^k, S^k)$ depend on the decisions that we make for the products $\{1, \ldots, i-1\}$, but not on the decisions that we make for the products $\{i, \ldots, n\}$. In our dynamic program, the decision epochs correspond to the products. At the decision epoch corresponding to product i, we choose the stage at which we should offer product i. Note that we may decide not to offer product i at all. In particular, to capture the decisions that we make at this decision epoch, we use the vector $\boldsymbol{x}_i = (\boldsymbol{x}^1_i, \ldots, \boldsymbol{x}^m_i) \in \{0, 1\}^m$, where $x^k_i = 1$ if and only if we offer product i in stage k. Since we can offer a product in no more than one stage, the decision should satisfy $\sum_{k \in M} x^k_i \leq 1$. At the decision epoch corresponding to product i, we have already made the decisions for the products $\{1, \ldots, i-1\}$. Therefore, the state variable at the decision epoch corresponding to product i are the values of $\Phi^k_i(f^k, S^k)$ and $\Gamma^k_i(h^k, S^k)$ for all $k \in M$, which are determined by the decisions that we make for the products in $\{1, \ldots, i-1\}$. Given that $\Phi^k_i(f^k, S^k) = f^k_i$ and $\Gamma^k_i(h^k, S^k) = h^k_i$ for all $k \in M$, by (2.6), after we make the decision for product i, we can compute $\Phi^k_{i+1}(f^k, S^k)$ and $\Gamma^k_{i+1}(h^k, S^k)$ as $\lceil f^k_i - r_i\, v^k_i\, x^k_i \rceil$ and $\lfloor f^k_i - v^k_i\, x^k_i \rfloor$ for all $k \in M$. To capture the state at the decision epoch corresponding to product i, we define the vectors

$\boldsymbol{f}_i = (f_i^1, \ldots, f_i^m)$ and $\boldsymbol{h}_i = (h_i^1, \ldots, h_i^m)$. Thus, letting $\boldsymbol{e}^k \in \{0,1\}^m$ be a unit vector with a one in the k-th component, to check whether a fixed value of $(\boldsymbol{f}, \boldsymbol{h}) \in \textsc{Dom}_+^m \times \textsc{Dom}_+^m$ satisfies $(\boldsymbol{f}, \boldsymbol{h}) \in \widetilde{\mathcal{P}}$, we can solve the dynamic program

$$V_i(\boldsymbol{f}_i, \boldsymbol{h}_i) = \max_{\substack{\boldsymbol{x}_i \in \{0,1\}^m : \\ \sum_{k \in M} x_i^k \leq 1}} \left\{ V_{i+1}\left(\left\lceil \boldsymbol{f}_i - \sum_{k \in M} \boldsymbol{e}^k \, r_i \, v_i^k \, x_i^k \right\rceil, \left\lfloor \boldsymbol{h}_i - \sum_{k \in M} \boldsymbol{e}^k \, v_i^k \, x_i^k \right\rfloor \right) \right\}, \tag{2.13}$$

with the boundary condition that $V_{n+1}(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1}) = 0$ if $f_{n+1}^k \leq 0$ and $h_{n+1}^k \geq 0$ for all $k \in M$. Otherwise, we have $V_{n+1}(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1}) = -\infty$. Once we compute the value functions $\{V_i(\cdot, \cdot) : i \in N\}$ through the dynamic program above, for a given value of $(\boldsymbol{f}, \boldsymbol{h}) \in \textsc{Dom}_+^m \times \textsc{Dom}_+^m$, we have $V_1(\boldsymbol{f}, \boldsymbol{h}) = 0$ if and only if $(\boldsymbol{f}, \boldsymbol{h}) \in \widetilde{\mathcal{P}}$. Note that we apply the operators $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ on the vectors in (2.13) componentwise.

The state variable $(\boldsymbol{f}_i, \boldsymbol{h}_i)$ in the dynamic program in (2.13) takes values in the set $\textsc{Dom}_+^m \times \textsc{Dom}_+^m$. Therefore, the number of possible values for the state variable is countable but not yet finite. Next, we give a natural bound on the state variable in this dynamic program, in which case, the number of possible values for the state variable becomes finite. Thus, we can solve the dynamic program in (2.13) in finite number of operations. In the next lemma, along with the discussion that follows this lemma, we show that we do not need to consider the values of the state variable whose components exceed a certain upper bound. In this lemma and throughout the rest of the chapter, for notational brevity, we let $R_{\max} = \max\{r_i \, v_i^k : i \in N, \; k \in M\}$, $R_{\min} = \min\{r_i \, v_i^k : i \in N, \; k \in M\}$, $V_{\max} = \max\{v_i^k : i \in N, \; k \in M\}$ and $V_{\min} = \min\{v_i^k : i \in N, \; k \in M\}$. Also, we define the function $\Delta(\rho, n) = ((1+\rho)^n - 1)/\rho$. Note that $\Delta(\rho, n) \geq (1 + \rho \, n - 1)/\rho = n$.

Lemma 2.6.1 *For any $\{\boldsymbol{x}_i : i \in N\}$ and $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}}) \in \Re_+^m \times \Re_+^m$, assume that $\{(\boldsymbol{f}_i, \boldsymbol{h}_i) : i \in N\}$ are given by $f_{i+1}^k = \lceil f_i^k - r_i \, v_i^k \, x_i^k \rceil$ and $h_{i+1}^k = \lfloor h_i^k - v_i^k \, x_i^k \rfloor$ for all $i \in N$, $k \in M$ with $f_1^k = \widehat{f}^k$ and $h_1^k = \widehat{h}^k$. If $\widehat{f}^k > \lceil n R_{\max} \rceil$, then we have $f_{n+1}^k > 0$. Similarly, if $\widehat{h}^k \geq \lceil \Delta(\rho, n) \, V_{\max} \rceil$,* *then we have* $h^k_{n+1} \geq 0$.

The proof of the lemma above follows from an induction over the products and we defer it to Appendix A.1. The value function $V_{n+1}(\cdot,\cdot)$ in (2.13) takes the value 0 or $-\infty$, depending only on the signs of the components of the state variable. Furthermore, given the state variable $(\boldsymbol{f}_i, \boldsymbol{h}_i)$ in the decision epoch corresponding to product i, each component of the state variable at the next decision epoch is computed by using the recursion $f^k_{i+1} = \lceil f^k_i - r_i\, v^k_i\, x^k_i \rceil$ and $h^k_{i+1} = \lfloor h^k_i - v^k_i\, x^k_i \rfloor$ for all $k \in M$. Therefore, if we start with the initial state $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$, then each component of the state variable is computed by using the recursion in Lemma 2.6.1. In this case, by Lemma 2.6.1, if $\widehat{f}^k > \lceil nR_{\max} \rceil$ for some $k \in M$ in the initial state variable $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$, then the same component of the state variable $(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1})$ at the final decision epoch always satisfies $f^k_{n+1} > 0$, irrespective of the decisions that we take in the intermediate decision epochs. Thus, noting the boundary condition in the dynamic program in (2.13), we have $V_{n+1}(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1}) = -\infty$ irrespective of our decisions, which implies that $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}}) = -\infty$. In other words, we can immediately deduce that $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}}) = -\infty$, whenever $\widehat{f}^k > \lceil nR_{\max} \rceil$ for some $k \in M$. We do not need to compute $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ explicitly whenever $\widehat{f}^k > nR_{\max}$ for some $k \in M$. On the other hand, by Lemma 2.6.1, if $\widehat{h}^k \geq \lceil \Delta(\rho, n)\, V_{\max} \rceil$ for some $k \in M$ in the initial state variable $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$, then the same component of the state variable $(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1})$ at the final decision epoch always satisfies $h^k_{n+1} \geq 0$, again, irrespective of the decisions that we take in the intermediate decision epochs. Thus, since the value function $V_{n+1}(\cdot,\cdot)$ only depends on the signs of the components of the state variable, as long as $\widehat{h}^k \geq \lceil \Delta(\rho, n)\, V_{\max} \rceil$ in the initial state variable, the value function $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ does not depend on the specific value of $\widehat{h}^k$. In other words, if we have $\widehat{h}^k > \lceil \Delta(\rho, n)\, V_{\max} \rceil$ for some $k \in M$ in the initial state variable $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$, then we can bump the value of this component of the state variable down to $\lceil \Delta(\rho, n)\, V_{\max} \rceil$ without changing the value function $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$. Therefore, we do not need to compute $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ explicitly either whenever $\widehat{h}^k > \lceil \Delta(\rho, n)\, V_{\max} \rceil$ for some $k \in M$. In this case, we do not need to compute the value function $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ when $\widehat{f}^k > \lceil nR_{\max} \rceil$ or $h^k > \lceil \Delta(\rho, n)\, V_{\max} \rceil$ for some $k \in M$.

Also, since $r_i\, v_i^k \geq R_{\min}$, if $0 < \widehat{f}^k < \lfloor R_{\min} \rfloor$ for some $k \in M$ in the initial state variable $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$, then offering any of the products at any decision epoch sets the value of this component of the state variable to zero at the subsequent decision epochs. However, if $\widehat{f}^k = \lfloor R_{\min} \rfloor$ for some $k \in M$, then offering any of the products at any decision epoch also sets the value of this component of the state variable to zero at the subsequent decision epochs. Noting that the value function $V_{n+1}(\cdot, \cdot)$ only depends on the signs of the components of the state variable, if $0 < \widehat{f}^k < \lfloor R_{\min} \rfloor$ for some $k \in M$, then we can bump the value of this component of the state variable up to $\lfloor R_{\min} \rfloor$ without changing the value function $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$. In other words, we do not need to compute $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ explicitly whenever $0 < \widehat{f}^k < \lfloor R_{\min} \rfloor$ for some $k \in M$. Using a similar argument, we do not need to compute $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ explicitly whenever $0 < \widehat{h}^k < \lfloor V_{\min} \rfloor$ for some $k \in M$. So, we do not need to compute the value function $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ when $0 < \widehat{f}^k < \lfloor R_{\min} \rfloor$ or $0 < \widehat{h}^k < \lfloor V_{\min} \rfloor$ for some $k \in M$.

Putting the discussion in the previous two paragraphs together, we only need to compute the value function $V_1(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}})$ for the values of the initial state variable $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{h}}) \in \textsc{Dom}_+^m \times \textsc{Dom}_+^m$ that satisfies $\widehat{f}^k \in \{0\} \cup [\lfloor R_{\min} \rfloor, \lceil nR_{\max} \rceil]$ and $\widehat{h}^k \in \{0\} \cup [\lfloor V_{\min} \rfloor, \lceil \Delta(\rho, n)\, V_{\max} \rceil]$ for all $k \in M$. Once we compute the value function at these values of the state variable, we can immediately deduce the value function at other values of the state variable. This discussion also indicates that there exists an optimal solution $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$ to the problem in Step 1 of the APPROX algorithm that satisfies $\widetilde{f}^k \in \{0\} \cup [\lfloor R_{\min} \rfloor, \lceil n R_{\max} \rceil]$ and $\widetilde{h}^k \in \{0\} \cup [\lfloor V_{\min} \rfloor, \lceil \Delta(\rho, n)\, V_{\max} \rceil]$ for all $k \in M$. In particular, if $\widetilde{f}^k > \lceil n R_{\max} \rceil$ for some $k \in M$, then $V_1(\widetilde{f}, \widetilde{h}) = -\infty$ by our earlier discussion, so $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}}) \notin \widetilde{\mathcal{P}}$, which indicates that $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$ is not feasible to the problem in Step 1. On the other hand, an optimal solution $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$ to the problem in Step 1 satisfies $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}}) \in \widetilde{\mathcal{P}}$ so that $V_1(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}}) = 0$. If we have $\widetilde{h}^k \geq \lceil \Delta(\rho, n)\, V_{\max} \rceil$ for some $k \in M$, then we, by our earlier discussion, can bump the value of $\widetilde{h}^k$ down to $\lceil \Delta(\rho, n)\, V_{\max} \rceil$ without changing the value of $V_1(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$ from zero. Therefore, the solution that we obtain in this way is still feasible to the problem in Step 1. Furthermore, since the objective function of this problem is decreasing in h^k, the solution that we obtain in this way is also an optimal solution. Similarly if we have $0 < \widetilde{f}^k < \lfloor R_{\min} \rfloor$ for some $k \in M$, then we can bump the value of $\widetilde{f}^k$ up to $\lfloor R_{\min} \rfloor$ without changing the value of $V_1(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$ from zero. Therefore, the solution that we obtain in this way is still feasible to the problem in Step 1. Furthermore, the objective function of this problem is increasing in f^k, indicating that the solution that we obtain in this way is also an optimal solution. Lastly, using a similar argument, if $0 < \widetilde{h}^k < \lfloor V_{\min} \rfloor$ for some $k \in M$, then we can bump the value of $\widetilde{h}^k$ down to zero and still obtain an optimal solution to the problem in Step 1. So, there exists a finite number of possible solutions to the problem in Step 1 of the APPROX algorithm and we can solve the dynamic program in (2.13) to check whether each one of these solutions is feasible. In the next section, we use this observation to give our FPTAS.

2.7 Fully Polynomial-Time Approximation Scheme

At the end of Section 2.5, we discuss that if we execute the APPROX algorithm with $\rho = \epsilon/(8n)$ for some $\epsilon \in (0,1)$, then we obtain a solution to problem (2.1) that provides an expected revenue deviating from the optimal expected revenue by no more than a factor of $1-\epsilon$. Next, we discuss that if we execute the APPROX algorithm with $\rho = \epsilon/(8n)$ for some $\epsilon \in (0,1)$, then the running time is polynomial in input size, when the number of stages is fixed. In this way, we obtain our FPTAS. We know that there exists an optimal solution $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}})$ to the problem in Step 1 of the APPROX algorithm that satisfies $\widetilde{f}^k \in \{0\} \cup [\lfloor R_{\min} \rfloor, \lceil n R_{\max} \rceil]$ and $\widetilde{h}^k \in \{0\} \cup [\lfloor V_{\min} \rfloor, \lceil \Delta(\rho, n)\, V_{\max} \rceil]$ for all $k \in M$. Noting the definition of DOM, the number of possible values of $(\widetilde{\boldsymbol{f}}, \widetilde{\boldsymbol{h}}) \in \text{DOM}_+^m \times \text{DOM}_+^m$ that lie in these intervals is given by

$$O\left(\left(\frac{\log(n\frac{R_{\max}}{R_{\min}})}{\log(1+\rho)}\right)^m \times \left(\frac{\log(\Delta(\rho,n)\,\frac{V_{\max}}{V_{\min}})}{\log(1+\rho)}\right)^m\right) = O\left(\left(\frac{\log(n\frac{R_{\max}}{R_{\min}}) \times \log(\Delta(\rho,n)\frac{V_{\max}}{V_{\min}})}{\rho^2}\right)^m\right). \quad (2.14)$$

To check whether a value of $(\boldsymbol{f}, \boldsymbol{h}) \in \text{DOM}_+^m \times \text{DOM}_+^m$ is feasible to the problem in Step 1 of the APPROX algorithm, we can use the value function $V_1(\boldsymbol{f}, \boldsymbol{h})$. We know that we only need to compute the value function $V_1(\boldsymbol{f}, \boldsymbol{h})$ for the values of the initial state variable $(\boldsymbol{f}, \boldsymbol{h}) \in \text{DOM}_+^m \times \text{DOM}_+^m$ that satisfies $f^k \in \{0\} \cup [\lfloor R_{\min} \rfloor, \lceil n R_{\max} \rceil]$ and $h^k \in \{0\} \cup [\lfloor V_{\min} \rfloor, \lceil \Delta(\rho, n)\, V_{\max} \rceil]$ for all $k \in M$. Each component of the state variable in the dynamic program in (2.13) decreases as we move from one decision epoch to the next. Therefore, if a component of a state variable turns negative, then it never turns positive in a subsequent decision epoch. Since $V_{n+1}(\cdot,\cdot)$ only depends on the signs of the components of the state variable, the number of possible values for the state variable at each decision epoch is also given by the expression in (2.14).

There are $O(n)$ decision epochs in the dynamic program in (2.13). We can compute the value functions $\{V_i(\cdot,\cdot) : i \in N\}$ starting from the last decision epoch and moving backwards over the decision epochs. Computation of the value function at a particular state takes $O(m)$ operations, since there are m stages in which we can offer a product. Thus, noting the number of possible values for the state variable in (2.14), we can execute Step 1 of the APPROX algorithm in $O(mn(\log(n\frac{R_{\max}}{R_{\min}}))^m (\log(\Delta(\rho,n)\frac{V_{\max}}{V_{\min}}))^m/\rho^{2m})$ operations. On the other hand, to execute Step 2 of the APPROX algorithm, once we obtain an optimal solution $(\widetilde{\boldsymbol{f}},\widetilde{\boldsymbol{h}})$ to the problem in Step 1, we can follow the optimal state and action trajectory in the dynamic program in (2.13). In particular, letting $(\widetilde{\boldsymbol{f}}_1,\widetilde{\boldsymbol{h}}_1) = (\widetilde{\boldsymbol{f}},\widetilde{\boldsymbol{h}})$, we can compute $\widetilde{\boldsymbol{x}}_i$ and $(\widetilde{\boldsymbol{f}}_i,\widetilde{\boldsymbol{h}}_i)$ recursively as $\widetilde{\boldsymbol{x}}_i = \arg\max\{V_{i+1}(\lceil \widetilde{\boldsymbol{f}}_i - \sum_{k\in M} \boldsymbol{e}^k\, r_i\, v_i^k\, x_i^k \rceil, \lfloor \widetilde{\boldsymbol{h}}_i - \sum_{k\in M} \boldsymbol{e}^k\, v_i^k\, x_i^k \rfloor) : \sum_{k\in M} x_i^k \leq 1,\ \boldsymbol{x}_i \in \{0,1\}^m\}$ with $\widetilde{\boldsymbol{f}}_{i+1} = \lceil \widetilde{\boldsymbol{f}}_i - \sum_{k\in M} \boldsymbol{e}^k\, r_i\, v_i^k\, x_i^k \rceil$ and $\widetilde{\boldsymbol{h}}_{i+1} = \lfloor \widetilde{\boldsymbol{h}}_i - \sum_{k\in M} \boldsymbol{e}^k\, v_i^k\, x_i^k \rfloor$ for all $i \in N$. In this case, letting $\widetilde{S}^k = \{i \in N : \widetilde{x}_i^k = 1\}$ for all $k \in M$, the solution $(\widetilde{S}^1,\ldots,\widetilde{S}^m) \in \mathcal{F}$ satisfies $\Phi_{n+1}^k(\widetilde{f}^k,\widetilde{S}^k) \leq 0$ and $\Gamma_{n+1}^k(\widetilde{f}^k,\widetilde{S}^k) \geq 0$ for all $k \in M$. The number of operations required to execute Step 2 of the APPROX algorithm is dominated by that required to execute Step 1. In the next theorem, we build on this discussion to give an FPTAS for problem (2.1).

Theorem 2.7.1 *For any $\epsilon \in (0,1)$, we can find a solution to problem* (2.1) *such that the expected revenue from this solution deviates from the optimal objective value of problem* (2.1) *by at most a factor of $1-\epsilon$ and the number of operations required to obtain this solution is $O(mn^{2m+1}(\log(n\frac{R_{\max}}{R_{\min}}))^m (\log(n\frac{V_{\max}}{V_{\min}}))^m/\epsilon^{2m})$.*

Proof. Consider executing the APPROX algorithm with $\rho = \epsilon/(8n)$. By the discussion at the end Section 2.5, this algorithm returns a solution such that the expected revenue from this solution deviates from the optimal objective value of problem (2.1) by at most a factor of $1-\epsilon$. On the other hand, by the discussion right before the theorem, we can execute the APPROX algorithm in $O(mn(\log(n\frac{R_{\max}}{R_{\min}}))^m\,(\log(\Delta(\rho,n)\frac{V_{\max}}{V_{\min}}))^m/\rho^{2m})$ operations. Noting that $\exp(x/2)\le 1+x$ for all $x\in(0,1)$ and $2\rho n=\epsilon/4<1$, we have $\Delta(\rho,n)=((1+\rho)^n-1)/\rho\le(\exp(\rho n)-1)/\rho=(1+2\rho n-1)/\rho=2n$. In this case, replacing ρ with $\epsilon/(8n)$ and $\Delta(\rho,n)$ with $2n$, we obtain the number of operations to execute the APPROX algorithm with $\rho=\epsilon/(8n)$. ■

The number of operations in Theorem 2.7.1 is polynomial in the input size and $1/\epsilon$ when the number of stages m is fixed, yielding an FPTAS for problem (2.1) for fixed number of stages. The state variable in the dynamic program in (2.13) takes values in $\text{DOM}_+^m\times\text{DOM}_+^m$. It turns out we can formulate an equivalent dynamic program, where the state variable takes values in $\text{DOM}_+^{m-1}\times\text{DOM}_+^m$. Using the latter dynamic program improves the number of operations in our FPTAS. To formulate the equivalent dynamic program, we choose one stage arbitrarily. We choose the first stage in the discussion that follows. We partition the vector $\boldsymbol{f}=(f^1,\dots,f^m)$ into the scalar f^1 and the vector $\boldsymbol{f}^{-1}=(f^2,\dots,f^m)$. Therefore, we can write $\boldsymbol{f}=(f^1,\boldsymbol{f}^{-1})$. It is not difficult to use induction over the products to show that $\Phi^k_{n+1}(f^k,S^k)$ is increasing in f^k. In particular, since $\Phi^k_1(f^k,S^k)=f^k$, $\Phi^k_1(f^k,S^k)$ is increasing f^k. If we assume that $\Phi^k_i(f^k,S^k)$ is increasing in f^k and note that $\lceil x\rceil$ in increasing in x, then (2.6) implies that $\Phi^k_{i+1}(f^k,S^k)$ is increasing in f^k as well, completing the induction argument. Therefore, if $\Phi^k_{i+1}(\widehat{f^k},S^k)\le 0$, then we have $\Phi^k_{i+1}(f^k,S^k)\le 0$ for all $f^k\le\widehat{f^k}$. Similarly, if $\Phi^k_{i+1}(\widehat{f^k},S^k)>0$, then we have $\Phi^k_{i+1}(f^k,S^k)>0$ for all $f^k>\widehat{f^k}$.

In this case, consider a fixed value of $(\boldsymbol{f}^{-1},\boldsymbol{h})\in\text{DOM}_+^{m-1}\times\text{DOM}_+^m$. Noting the definition of $\widetilde{\mathcal{P}}$ in (2.7), depending on the value of $(\boldsymbol{f}^{-1},\boldsymbol{h})\in\text{DOM}_+^{m-1}\times\text{DOM}_+^m$, there exists a threshold $T(\boldsymbol{f}^{-1}, \boldsymbol{h})$ such that we have $((f^1, \boldsymbol{f}^{-1}), \boldsymbol{h}) \in \widetilde{\mathcal{P}}$ for all $f^1 \in \text{DOM}_+$ that satisfies $f^1 \leq T(\boldsymbol{f}^{-1}, \boldsymbol{h})$, whereas we have $((f^1, \boldsymbol{f}^{-1}), \boldsymbol{h}) \notin \widetilde{\mathcal{P}}$ for all $f^1 \in \text{DOM}_+$ that satisfies $f^1 > T(\boldsymbol{f}^{-1}, \boldsymbol{h})$. In other words, we have $V_1((f^1, \boldsymbol{f}^{-1}), \boldsymbol{h}) = 0$ if $f^1 \leq T(\boldsymbol{f}^{-1}, \boldsymbol{h})$, whereas we have $V_1((f^1, \boldsymbol{f}^{-1}), \boldsymbol{h}) = -\infty$ if $f^1 > T(\boldsymbol{f}^{-1}, \boldsymbol{h})$. Thus, if we can compute the threshold $T(\boldsymbol{f}^{-1}, \boldsymbol{h})$ for all $(\boldsymbol{f}^{-1}, \boldsymbol{h}) \in \text{DOM}_+^{m-1} \times \text{DOM}_+^m$, then we do not need to solve the dynamic program in (2.13). In the rest of this section, we give a dynamic program to compute the threshold $T(\boldsymbol{f}^{-1}, \boldsymbol{h})$. In particular, letting $M^{-1} = M \setminus \{1\}$ for notational brevity, we consider the dynamic program

$$J_i(\boldsymbol{f}_i^{-1}, \boldsymbol{h}_i) = \max_{\substack{\boldsymbol{x}_i \in \{0,1\}^m : \\ \sum_{k \in M} x_i^k \leq 1}} \left\{ r_i\, v_i^1\, x_i^1 + \left\lfloor J_{i+1}\left(\left\lceil \boldsymbol{f}_i^{-1} - \sum_{k \in M^{-1}} \boldsymbol{e}^k\, r_i\, v_i^k\, x_i^k \right\rceil, \left\lfloor \boldsymbol{h}_i - \sum_{k \in M} \boldsymbol{e}^k\, v_i^k\, x_i^k \right\rfloor \right) \right\rfloor \right\}, \quad (2.15)$$

with the boundary condition that $J_{n+1}(\boldsymbol{f}_{n+1}^{-1}, \boldsymbol{h}_{n+1}) = 0$ if $f_{n+1}^k \leq 0$ for all $k \in M^{-1}$ and $h_{n+1}^k \geq 0$ for all $k \in M$. Otherwise, we have $J_{n+1}(\boldsymbol{f}_{n+1}^{-1}, \boldsymbol{h}_{n+1}) = -\infty$. In Appendix A.2, we show that we have $V_1((f^1, \boldsymbol{f}^{-1}), \boldsymbol{h}) = 0$ if $f^1 \leq \lfloor J_1(\boldsymbol{f}^{-1}, \boldsymbol{h}) \rfloor$, whereas we have $V_1((f^1, \boldsymbol{f}^{-1}), \boldsymbol{h}) = -\infty$ if $f^1 > \lfloor J_1(\boldsymbol{f}^{-1}, \boldsymbol{h}) \rfloor$. Therefore, we can use $\lfloor J_1(\boldsymbol{f}^{-1}, \boldsymbol{h}) \rfloor$ as the threshold $T(\boldsymbol{f}^{-1}, \boldsymbol{h})$, preventing the need to compute $V_1(\boldsymbol{f}, \boldsymbol{h})$ for all $(\boldsymbol{f}, \boldsymbol{h}) \in \text{DOM}_+^m \times \text{DOM}_+^m$.

Working with the dynamic program in (2.15), we can use precisely the same argument earlier in this section to give an FPTAS for problem (2.1). This FPTAS provides a solution such that the expected revenue from this solution deviates from the optimal objective value of problem (2.1) by at most a factor of $1 - \epsilon$ and the number of operations required to obtain this solution is $O(mn^{2m} (\log(n \frac{R_{\max}}{R_{\min}}))^{m-1} (\log(n \frac{V_{\max}}{V_{\min}}))^m / \epsilon^{2m-1})$. Note that although working with the dynamic program in (2.15) allows us to obtain a more efficient FPTAS, the dynamic program in (2.13) is substantially more interpretable than the one in (2.15). Therefore, we chose to use the dynamic program in (2.13) in our initial presentation of our FPTAS.

Prior to our work, [9] consider assortment optimization problems under a mixture of multinomial logit models. The authors develop an FPTAS by using the connections of their problem to the knapsack problem and aligning the cumulative capacity consumptions to a geometric grid. Due to the multiple stages in the choice process in our assortment optimization problem, we need to be careful to characterize the error resulting from aligning the cumulative capacity consumptions to a geometric grid, as in Proposition 2.4.1. By Theorem 2.5.2, for a fixed grid size ρ, the error is exponential in the number of products. As in Theorem 2.7.1, choosing the grid size ρ such that it differs from the precision ϵ by a factor of n, we get our FPTAS.

Lastly, although our assortment optimization problem is NP-hard, we can provide some structure for the form of the optimal solution. In particular, we can show that the union of the optimal sets to offer in the different stages is nested by revenue, including a certain number of products with the largest revenues. In other words, there exists an optimal solution $(\widehat{S}^1, \ldots, \widehat{S}^m)$ to problem (2.1) such that $\cup_{k \in M} \widehat{S}^k = \{i \in N : r_i \geq \widehat{\zeta}\}$ for some constant $\widehat{\zeta}$. While this result intuitively suggests that we should give more priority to offering the products with larger revenues, it does not allow us to find the optimal solution efficiently, since this result does not characterize the stage in which each product should be offered. We discuss this result in Appendix A.3.

2.8 Constraints on the Offered Sets of Products

In this section, we consider two types of constraints on the sets of products that we can offer to the customers. First, in online retail, for example, we may display the search results to a customer sequentially through multiple webpages. If there is limited space on the webpage, then it may be desirable to limit the number of products offered in each stage. We refer to this type of constraints as cardinality constraints *within* stages. Note that not all online retail applications require cardinality constraints within stages. In particular, it may be at our discretion to decide how many products to display on each page, in which case, it may not be necessary to limit the number of products that we offer in each stage. Second, in scheduling healthcare appointments over the phone, for example, to gently guide the patient through the choice process, we may offer sets of appointment slots sequentially. To avoid overwhelming the patient with a large number of options, it may be desirable to limit the total number of appointment slots offered over all stages. We refer to this type of constraints as cardinality constraints *across* stages.

Cardinality Constraints within Stages. We let C^k be the maximum number of products that we can offer in stage k. Therefore, the feasible sets of products that we can offer over all stages are $\mathcal{F} = \{(S^1, \ldots, S^m) : S^k \subseteq N \;\forall\, k \in M,\; |S^k| \leq C^k \;\forall\, k \in M,\; S^k \cap S^\ell = \varnothing \;\forall\, k \neq \ell\}$. The development in Sections 2.3, 2.4 and 2.5 does not change at all, as long as we use this definition of $\mathcal{F}$ under cardinality constraints within stages. All we need to do is to interpret all occurrences of $\mathcal{F}$ as the one under cardinality constraints within stages. We slightly modify the dynamic program in (2.13) that we use to check whether a fixed value of $(\boldsymbol{f}, \boldsymbol{h}) \in \textsc{Dom}_+^m \times \textsc{Dom}_+^m$ satisfies $(\boldsymbol{f}, \boldsymbol{h}) \in \widetilde{\mathcal{P}}$. In particular, we let c_i^k be the number of products among $\{1, \ldots, i-1\}$ that we offer in stage k. Defining the vector $\boldsymbol{c}_i = (c_i^1, \ldots, c_i^m)$, we use the dynamic program

$$V_i(\boldsymbol{f}_i, \boldsymbol{h}_i, \boldsymbol{c}_i) \;=\; \max_{\substack{\boldsymbol{x}_i \in \{0,1\}^m : \\ \sum_{k \in M} x_i^k \leq 1}} \left\{ V_{i+1}\left(\left\lceil \boldsymbol{f}_i - \sum_{k \in M} \boldsymbol{e}^k\, r_i\, v_i^k\, x_i^k \right\rceil, \left\lfloor \boldsymbol{h}_i - \sum_{k \in M} \boldsymbol{e}^k\, v_i^k\, x_i^k \right\rfloor, \boldsymbol{c}_i + \sum_{k \in M} \boldsymbol{e}^k\, x_i^k \right) \right\},$$

with the boundary condition that $V_{n+1}(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1}, \boldsymbol{c}_{n+1}) = 0$ if $f_{n+1}^k \leq 0$, $h_{n+1}^k \geq 0$ and $c_{n+1}^k \leq C^k$ for all $k \in M$. Otherwise, we have $V_{n+1}(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1}, \boldsymbol{c}_{n+1}) = -\infty$. In this case, we have $V_1(\boldsymbol{f}, \boldsymbol{h}, \boldsymbol{0}) = 0$ if and only if $(\boldsymbol{f}, \boldsymbol{h}) \in \widetilde{\mathcal{P}}$, where $\boldsymbol{0} \in \Re_+^m$ is a vector of all zeros. We can slightly modify the discussion in Section 2.7 to construct our FPTAS. In particular, using the dynamic program above, under cardinality constraints within stages, we can execute the APPROX algorithm in $O(mn^{m+1}(\log(n\frac{R_{\max}}{R_{\min}}))^m\,(\log(\Delta(\rho,n)\frac{V_{\max}}{V_{\min}}))^m/\rho^{2m})$ operations. Choosing $\rho = \epsilon/(8n)$, we obtain an FPTAS with a running time of $O(mn^{3m+1}(\log(n\frac{R_{\max}}{R_{\min}}))^m\,(\log(n\frac{V_{\max}}{V_{\min}}))^m/\epsilon^{2m})$ to obtain a $(1-\epsilon)$-approximate solution. Lastly, using the same approach in the dynamic program in (2.15), we can reduce this running time to $O(mn^{3m}(\log(n\frac{R_{\max}}{R_{\min}}))^{m-1}\,(\log(n\frac{V_{\max}}{V_{\min}}))^m/\epsilon^{2m-1})$.

Cardinality Constraints across Stages. We let C be the maximum total number of products that we can offer over all stages. Therefore, the feasible sets of products that we can offer over all stages are $\mathcal{F} = \{(S^1,\ldots,S^m) : S^k \subseteq N\ \forall\, k \in M,\ |\cup_{k\in M} S^k| \leq C,\ S^k \cap S^\ell = \emptyset\ \forall\, k \neq \ell\}$. Once again, as long as we use the definition of $\mathcal{F}$ under cardinality constraints across stages, the development in Sections 2.3, 2.4 and 2.5 does not change at all. We slightly modify the dynamic program in (2.13) that we use to check whether a fixed value of $(\boldsymbol{f}, \boldsymbol{h}) \in \text{DOM}_+^m \times \text{DOM}_+^m$ satisfies $(\boldsymbol{f}, \boldsymbol{h}) \in \widetilde{\mathcal{P}}$. We use c_i to denote the total number of products among $\{1,\ldots,i-1\}$ that we offer in any of the stages. In this case, we use the dynamic program

$$V_i(\boldsymbol{f}_i, \boldsymbol{h}_i, c_i) = \max_{\substack{\boldsymbol{x}_i \in \{0,1\}^m : \\ \sum_{k\in M} x_i^k \leq 1}} \left\{ V_{i+1}\left(\left\lceil \boldsymbol{f}_i - \sum_{k\in M} \boldsymbol{e}^k\, r_i\, v_i^k\, x_i^k \right\rceil, \left\lfloor \boldsymbol{h}_i - \sum_{k\in M} \boldsymbol{e}^k\, v_i^k\, x_i^k \right\rfloor, c_i + \sum_{k\in M} x_i^k \right) \right\},$$

with the boundary condition that $V_{n+1}(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1}, c_{n+1}) = 0$ if $f_{n+1}^k \leq 0$ and $h_{n+1}^k \geq 0$ for all $k \in M$ and $c_{n+1} \leq C$. Otherwise, we have $V_{n+1}(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1}, c_{n+1}) = -\infty$. Thus, we have $V_1(\boldsymbol{f}, \boldsymbol{h}, 0) = 0$ if and only if

$(\boldsymbol{f},\boldsymbol{h}) \in \widetilde{\mathcal{P}}$. We can construct our FPTAS by slightly modifying the discussion in Section 2.7. We can use the dynamic program above to execute the APPROX algorithm in $O(mn^2(\log(n\frac{R_{\max}}{R_{\min}}))^m\,(\log(\Delta(\rho,n)\frac{V_{\max}}{V_{\min}}))^m/\rho^{2m})$ operations under cardinality constraints across stages. To obtain an FPTAS, we choose $\rho = \epsilon/(8n)$, in which case, to obtain a $(1-\epsilon)$-approximate solution, our FPTAS has a running time of $O(mn^{2m+2}(\log(n\frac{R_{\max}}{R_{\min}}))^m\,(\log(n\frac{V_{\max}}{V_{\min}}))^m/\epsilon^{2m})$. We can reduce this running time to $O(mn^{2m+1}(\log(n\frac{R_{\max}}{R_{\min}}))^{m-1}\,(\log(n\frac{V_{\max}}{V_{\min}}))^m/\epsilon^{2m-1})$ by using the same approach in the dynamic program in (2.15). It is also not too difficult to combine the discussion in this paragraph with the one in the previous paragraph to limit the number of products offered in each stage, as well as the total number of products offered over all stages, in which case, we have joint cardinality constraints within and across stages.

Space Constraints across Stages. Naturally, we can consider the case where each product occupies a certain amount of space and we limit the total space consumption of the products offered in each stage or over all stages. We refer to these types of constraints as space constraints within or across stages. We can extend the discussion in this section to space constraints across stages, but the extension to space constraints within stages appears to be difficult. In particular, under space constraints across stages, we let w_i be the space consumption of product i and T be the limit on the total space consumption of the products offered over all stages. The development in Sections 2.3, 2.4 and 2.5 still does not change at all. Under space constraints across stages, in the dynamic program in (2.13), the value function $V_i(\boldsymbol{f}_i,\boldsymbol{h}_i)$ would correspond to the minimum total space consumption for the products in $\{i,\ldots,n\}$ to ensure that $\Phi^k_{n+1}(f^k,S^k) \leq 0$ and $\Gamma_{n+1}(h^k,S^k) \geq 0$ for all $k \in M$, given that the decisions that we make for the products in $\{1,\ldots,i-1\}$ satisfy $f^k_i = \Phi^k_i(f^k,S^k)$ and $h^k_i = \Gamma^k_i(h^k,S^k)$ for all $k \in M$;

see [9]. Therefore, we have

$$V_i(\boldsymbol{f}_i, \boldsymbol{h}_i) \;=\; \min_{\substack{\boldsymbol{x}_i \,\in\, \{0,1\}^m \,:\\ \sum_{k \in M} x_i^k \,\leq\, 1}} \left\{ w_i \sum_{k \in M} x_i^k + V_{i+1}\Big(\Big\lceil \boldsymbol{f}_i - \sum_{k \in M} \boldsymbol{e}^k \, r_i \, v_i^k \, x_i^k \Big\rceil, \Big\lfloor \boldsymbol{h}_i - \sum_{k \in M} \boldsymbol{e}^k \, v_i^k \, x_i^k \Big\rfloor \Big) \right\},$$

with the boundary condition that $V_{n+1}(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1}) = 0$ if $f_{n+1}^k \leq 0$ and $h_{n+1}^k \geq 0$ for all $k \in M$. Otherwise, we have $V_{n+1}(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1}) = +\infty$. In this case, we have $V_1(\boldsymbol{f}, \boldsymbol{h}, 0) \leq T$ if and only if $(\boldsymbol{f}, \boldsymbol{h}) \in \widetilde{\mathcal{P}}$. The number of possible values for the state variable in the dynamic program above is the same as that for the dynamic program in (2.13). Thus, the number of possible values for the state variable in the dynamic program above is also given by the expression in (2.14), in which case, we can solve the dynamic program above in $O(mn(\log(n\frac{R_{\max}}{R_{\min}}))^m \, (\log(\Delta(\rho, n)\frac{V_{\max}}{V_{\min}}))^m/\rho^{2m})$ operations. Using the same discussion in Section 2.7 and earlier in this section, under space constraints across stages, we can choose $\rho = \epsilon/(8n)$ to construct an FPTAS with a running time of $O(mn^{2m+1}(\log(n\frac{R_{\max}}{R_{\min}}))^m \, (\log(n\frac{V_{\max}}{V_{\min}}))^m/\epsilon^{2m})$. We cannot reduce this running time by using the approach that we use in the dynamic program in (2.15), since we cannot characterize the value function above by using a threshold on one component of the state variable, as is done right before the dynamic program in (2.15). Note that the running time of our FPTAS under space constraints across stages is slightly worse than that under cardinality constraints across stages.

Space Constraints within Stages. Extending our approach to space constraints within stages appears to be difficult. Under space constraints within stages, similar to our approach under cardinality constraints within stages, we need an additional m-dimensional state variable in our dynamic program, which keeps track of the cumulative space consumption of the products offered in each stage. To obtain an FPTAS with a running time that is polynomial in the input size, we need to discretize the components of the additional state variable by using a geometric grid, but if we use such a discretization, then we cannot guarantee that we satisfy the constraints on the space consumptions of the products offered in each stage. It is simple to construct an FPTAS to obtain a solution that satisfies the constraints on the space consumptions with a multiplicative error of $1 + \epsilon$ in running time that is polynomial in $1/\epsilon$, but a feasible solution, by its definition, must satisfy the hard space constraints. In the previous paragraph, under space constraints across stages, we do not have to use an additional state variable in our dynamic program. In particular, under space constraints across stages, we need to keep track of the total space consumption of the products offered over all stages, which is a scalar. In this case, we can "overload" the value function so that the value of the value function, itself, corresponds to the cumulative space consumption of the products offered over all stages. Under space constraints within stages, however, we need to keep track of the space consumption of the products offered in each stage separately, which is an m-dimensional quantity, preventing us from "overloading" the value function.

2.9 Numerical Experiments

In this section, we give two sets of numerical experiments to test the effectiveness of our FTPAS. In the first set of numerical experiments, we work with randomly generated test problems. In the second set of numerical experiments, we use the data coming from a survey on the appointment slot choices of the patients in a clinic. Throughout, our goal is to understand how the practical performance of our FPTAS compares with its theoretical guarantee. We begin by formulating a linear program that provides an upper bound on the optimal expected revenue in problem (2.1). In this case, we can compare the upper bound on the optimal expected revenue with the expected revenue from the solution obtained by our FPTAS to assess the ex post optimality gap of the solution. Following the linear program, we give our two sets of numerical experiments.

2.9.1 Upper Bound on the Optimal Expected Revenue

We construct a linear program that we can use to obtain an upper bound on the optimal expected revenue in problem (2.1). All test problems in our numerical experiments have two stages. So, for notational brevity, we give our linear program for the case with two stages, but we discuss the extension to more than two stages at the end of this section. To formulate our linear program, we use the decision variable $x_i^k \in \{0,1\}$, where $x_i^k = 1$ if and only if we offer product i in stage k. Noting that we have $m = 2$ stages, we write problem (2.1) equivalently as

$$\widehat{Z} = \max_{\{\boldsymbol{x}_i \,:\, i \in N\} \in \{0,1\}^{n\times m}} \left\{ \frac{\sum_{i\in N} r_i\, v_i^1\, x_i^1}{1+\sum_{i\in N} v_i^1\, x_i^1} + \frac{1}{1+\sum_{i\in N} v_i^1\, x_i^1}\, \frac{\sum_{i\in N} r_i\, v_i^2\, x_i^2}{1+\sum_{i\in N} v_i^2\, x_i^2} \;:\; \sum_{k\in M} x_i^k \le 1 \;\forall\, i \in N \right\}. \quad (2.16)$$

Our linear program is based on guessing the values of $\sum_{i\in N} v_i^1\, x_i^1$ and $\sum_{i\in N} r_i\, v_i^2\, x_i^2 / (1+\sum_{i\in N} v_i^2\, x_i^2)$ above in an optimal solution.

Using $\{\widehat{\boldsymbol{x}}_i : i \in N\}$ to denote an optimal solution to problem (2.16), we let the intervals $[\underline{v}, \overline{v}]$ and $[\underline{\sigma}, \overline{\sigma}]$ be such that $\sum_{i\in N} v_i^1\, \widehat{x}_i^1 \in [\underline{v}, \overline{v}]$ and $\frac{\sum_{i\in N} r_i\, v_i^2\, \widehat{x}_i^2}{1+\sum_{i\in N} v_i^2\, \widehat{x}_i^2} \in [\underline{\sigma}, \overline{\sigma}]$.

Consider the problem

$$\widehat{Z}(\underline{\nu},\overline{\nu},\underline{\sigma},\overline{\sigma}) = \max_{\{\boldsymbol{x}_i \,:\, i\in N\}\in[0,1]^{n\times m}} \left\{ \frac{\sum_{i\in N} r_i\, v_i^1\, x_i^1}{1+\underline{\nu}} + \frac{1}{1+\underline{\nu}}\,\overline{\sigma} \;:\; \sum_{k\in M} x_i^k \le 1 \;\forall\, i\in N, \right.$$
$$\left. \sum_{i\in N} v_i^1\, x_i^1 \le \overline{\nu},\; \frac{\sum_{i\in N} r_i\, v_i^2\, x_i^2}{1+\sum_{i\in N} v_i^2\, x_i^2} \ge \underline{\sigma} \right\}. \quad (2.17)$$

In the problem above, the values of $\underline{\nu}, \overline{\nu}, \underline{\sigma}$ and $\overline{\sigma}$ are fixed. Therefore, the objective along with the first and second constraints are linear in the decision variables. We can write the third constraint as $\sum_{i\in N}(r_i - \underline{\sigma})\, v_i^2\, x_i^2 \ge \underline{\sigma}$, so that the third constraint is linear in the decision variables as well. Since the decision variable x_i^k takes values in the interval $[0,1]$ for all $i \in N$, $k \in M$, problem (2.17) is a linear program. We proceed to argue that the optimal objective value of the linear program in (2.17) is an upper bound on the optimal objective value of problem (2.16). Letting $\{\widehat{\boldsymbol{x}}_i : i \in N\}$ be an optimal solution to problem (2.16), by the definition of the intervals $[\underline{\nu},\overline{\nu}]$ and $[\underline{\sigma},\overline{\sigma}]$, we have $\sum_{i\in N} v_i^1\, \widehat{x}_i^1 \in [\underline{\nu},\overline{\nu}]$ and $\frac{\sum_{i\in N} r_i\, v_i^2\, \widehat{x}_i^2}{1+\sum_{i\in N} v_i^2\, \widehat{x}_i^2} \in [\underline{\sigma},\overline{\sigma}]$. Since $\{\widehat{\boldsymbol{x}}_i : i \in N\}$ is an optimal solution to problem (2.16), we also have $\sum_{k\in M} \widehat{x}_i^k \le 1$ for all $i \in N$. Therefore, the solution $\{\widehat{\boldsymbol{x}}_i : i \in N\}$ is feasible to problem (2.17). Since $\{\widehat{\boldsymbol{x}}_i : i \in N\}$ is an optimal solution to problem (2.16), noting that $\sum_{i\in N} v_i^1\, \widehat{x}_i^1 \ge \underline{\nu}$ and $\frac{\sum_{i\in N} r_i\, v_i^2\, \widehat{x}_i^2}{1+\sum_{i\in N} v_i^2\, \widehat{x}_i^2} \le \overline{\sigma}$, the optimal objective value of problem (2.16) satisfies

$$\widehat{Z} = \frac{\sum_{i\in N} r_i\, v_i^1\, \widehat{x}_i^1}{1+\sum_{i\in N} v_i^1\, \widehat{x}_i^1} + \frac{1}{1+\sum_{i\in N} v_i^1\, \widehat{x}_i^1}\, \frac{\sum_{i\in N} r_i\, v_i^2\, \widehat{x}_i^2}{1+\sum_{i\in N} v_i^2\, \widehat{x}_i^2} \le \frac{\sum_{i\in N} r_i\, v_i^1\, \widehat{x}_i^1}{1+\underline{\nu}} + \frac{1}{1+\underline{\nu}}\,\overline{\sigma}.$$

The expression on the right side above is the objective value of the linear program in (2.17) evaluated at the solution $\{\widehat{\boldsymbol{x}}_i : i \in N\}$. Therefore, there exists a feasible solution to the linear program in (2.17) with the corresponding objective value that is no less than the optimal objective value of problem (2.16). Therefore, the optimal objective value $\widehat{Z}(\underline{\nu},\overline{\nu},\underline{\sigma},\overline{\sigma})$ of the linear program in (2.17) is no less than the optimal objective value of problem (2.16), as desired. We cannot use the linear program in (2.17) immediately to obtain an upper bound on the optimal objective value of problem (2.16) since we cannot come up with the intervals $[\underline{\nu}, \overline{\nu}]$ and $[\underline{\sigma}, \overline{\sigma}]$ without knowing an optimal solution to problem (2.16). To get around this difficulty, we solve the linear program in (2.17) for multiple guesses for the intervals that can potentially include the values $\sum_{i \in N} v_i^1 \widehat{x}_i^1$ and $\frac{\sum_{i \in N} r_i v_i^2 \widehat{x}_i^2}{1+\sum_{i \in N} v_i^2 \widehat{x}_i^2}$.

Let $V_{\max} = \max\{v_i^1 : i \in N\}$ and $r_{\max} = \max\{r_i : i \in N\}$. So, the value of $\sum_{i \in N} v_i^1 \widehat{x}_i^1$ cannot exceed $nV_{\max}$. Viewing $\frac{\sum_{i \in N} r_i v_i^2 \widehat{x}_i^2}{1+\sum_{i \in N} v_i^2 \widehat{x}_i^2}$ as the weighted average of the revenues of the products, its value cannot exceed $r_{\max}$. We partition the interval $[0, nV_{\max}]$ by using the $K+1$ points $0 = \nu^0 \leq \nu^1 \leq \ldots \leq \nu^K = nV^{\max}$. Similarly, we partition the interval $[0, r_{\max}]$ by using the $L+1$ points $0 = \sigma^0 \leq \sigma^1 \leq \ldots \leq \sigma^L = r^{\max}$. Even if we do not know an optimal solution $\{\widehat{\boldsymbol{x}}_i : i \in N\}$ to problem (2.16), the value $\sum_{i \in N} v_i^1 \widehat{x}_i^1$ lies in one of the intervals $\{[\nu^{k-1}, \nu^k] : k = 1, \ldots, K\}$, whereas the value $\frac{\sum_{i \in N} r_i v_i^2 \widehat{x}_i^2}{1+\sum_{i \in N} v_i^2 \widehat{x}_i^2}$ lies in one of the intervals $\{[\sigma^{\ell-1}, \sigma^\ell] : \ell = 1, \ldots, L\}$. Thus, if we solve the linear program in (2.17) with $[\underline{\nu}, \overline{\nu}] = [\nu^{k-1}, \nu^k]$ and $[\underline{\sigma}, \overline{\sigma}] = [\sigma^{\ell-1}, \sigma^\ell]$ to compute $\widehat{Z}(\nu^{k-1}, \nu^k, \sigma^{\ell-1}, \sigma^\ell)$ for all $k = 1, \ldots, K$, $\ell = 1, \ldots, L$, then $\max\{\widehat{Z}(\nu^{k-1}, \nu^k, \sigma^{\ell-1}, \sigma^\ell) : k = 1, \ldots, K, \ell = 1, \ldots, L\}$ is an upper bound on the optimal objective value of problem (2.16).

We obtain an upper bound on the optimal objective value of problem (2.16) for any choice of the points $0 = \nu^0 \leq \nu^1 \leq \ldots \leq \nu^K = nV^{\max}$ and $0 = \sigma^0 \leq \sigma^1 \leq \ldots \leq \sigma^L = r^{\max}$. In our numerical experiments, we choose these points such that $\nu^k - \nu^{k-1} = \sigma^\ell - \sigma^{\ell-1} = 0.01$. Using our approach, we obtain reasonably tight upper bounds for our test problems, but a theoretical characterization of the tightness of the upper bounds requires characterizing the integrality gap of a multiple knapsack problem, which is difficult. We can extend our approach to the case where there are more than two stages in the choice process of the customers. This extension requires guessing the intervals that can potentially include the values $\sum_{i \in N} v_i^k \widehat{x}_i^k$ for all $k = 1, \ldots, m-1$, $\sum_{i \in N} r_i \, v_i^k \, \widehat{x}_i^k$ for all $k = 2, \ldots, m-1$ and $\frac{\sum_{i \in N} r_i \, v_i^m \, \widehat{x}_i^m}{1+\sum_{i \in N} v_i^m \, \widehat{x}_i^m}$. Therefore, the number of operations to compute an upper bound increases exponentially with the number of stages.

2.9.2 Randomly Generated Test Problems

In this section, we work with a large number of test problems that are randomly generated and test the performance of our FPTAS on these test problems.

Experimental Setup. We have $n = 18$ products and $m = 2$ stages in all of our test problems. The preference weight of a product is the same in both stages. Using v_i to denote the preference weight of product i, to generate the preference weights of the products, we sample θ_i from the uniform distribution over $[1, 10]$ for all $i \in N$ and set the preference weight of product i as $v_i = (1 - P_0) \, \theta_i / (P_0 \, \sum_{j \in N} \theta_j)$, where P_0 is a parameter that we vary. So, if we offer all products in a particular stage, then the probability of no purchase is $1/(1 + \sum_{i \in N} v_i) = 1/(1 + (1 - P_0)/P_0) = P_0$. Thus, the parameter P_0 controls the likelihood that a customer does not make a purchase in a stage. To generate the revenues of the products, for all $i \in N$, we set $r_i = 0.3$ or $r_i = 1$ with equal probabilities. We also tested our FPTAS with revenues of the products uniformly generated over a bounded interval and its performance was even better. After generating the revenues $\{r_i : i \in N\}$ and preference weights $\{v_i : i \in N\}$ as described, we use two approaches to finalize their values. In the first approach, we simply leave the generated revenues and preference weights as they are. So, there is no relationship between the revenue and preference weight of a product. In the second approach, we reindex the revenues $\{r_i : i \in N\}$ and preference weights $\{v_i : i \in N\}$ so that $r_1 \geq r_2 \geq \ldots \geq r_n$ and $v_1 \leq v_2 \leq \ldots \leq v_n$. Thus, the more expensive products have smaller preference weights. We use $T \in \{N, O\}$ to denote the approach to finalize the revenues and preference weights, where N corresponds to no relationship between the revenues and preference weights and O corresponds to ordering the revenues and preference weights. Varying $P_0 \in \{0.05, 0.1, 0.2, 0.3\}$ and $T \in \{N, O\}$, we obtain eight parameter settings. In each parameter setting, we generate 50 individual test problems by using the approach just discussed.

Benchmark. As a benchmark, we use a neighborhood search heuristic. In this heuristic, we represent a solution by using the vector $\boldsymbol{z} = (z_1, \ldots, z_n) \in \{0, 1, 2\}^n$, where we have $z_i = 0$ if we do not offer product i, $z_i = 1$ if we offer product i in stage 1 and $z_i = 2$ if we offer product i in stage 2. We define the neighborhood of the solution $\boldsymbol{z}$ as $\mathcal{N}(\boldsymbol{z}) = \{\boldsymbol{y} \in \{0, 1, 2\}^n : \|\boldsymbol{y} - \boldsymbol{z}\|_0 = 1\}$, which is the set of all solutions that differ from the solution $\boldsymbol{z}$ in one component. In the neighborhood search heuristic, at the first iteration, we have the solution $\boldsymbol{z}^1 = (0, \ldots, 0) \in \Re^n_+$, which does not offer any of the products. At iteration ℓ, given that we have the solution $\boldsymbol{z}^\ell$, we check the expected revenue provided by each solution in the neighborhood $\mathcal{N}(\boldsymbol{z}^\ell)$ and find the best solution in the neighborhood. If the expected revenue provided by the best solution in the neighborhood is larger than the expected revenue provided by the solution $\boldsymbol{z}^\ell$, then the solution $\boldsymbol{z}^{\ell+1}$ that we have at iteration $\ell + 1$ is the best solution in the neighborhood. Otherwise, we stop.

Numerical Results. In our numerical experiments, we use our FPTAS to obtain solutions with performance guarantees of $\epsilon = 1/4$ and $\epsilon = 1/2$.

Therefore, the expected revenues from the solutions obtained by our FPTAS are guaranteed to be at least 75% and 50% of the optimal expected revenue. By the discussion at the end of Section 2.5, to obtain such performance guarantees, we need to execute the APPROX algorithm with $\rho = \epsilon/(8n)$. We give our main numerical results in Table 2.1. In this table, the first column shows the parameter setting by using the pair (P_0, T), where P_0 and T are as discussed in our experimental setup. There are three blocks of four columns in the rest of the table. The first block of columns focuses on the performance of our FPTAS with $\epsilon = 1/4$, corresponding to a 75% performance guarantee. Recall that we generate 50 test problems in each parameter setting. For each test problem, we use our FPTAS to obtain an approximate solution. For each test problem, we also use the linear program in (2.17) to obtain an upper bound on the optimal expected revenue. In the first block of columns in Table 2.1, the first column shows the average percent gap between the upper bound on the optimal expected revenue and the expected revenue from the solution obtained by our FPTAS, where the average is computed over the 50 test problems in a parameter setting. In particular, for test problem k, letting Rev^k be the expected revenue from the solution obtained by our FPTAS and Upp^k be the upper bound on the optimal expected revenue, the first column shows the average of the data $\{100 \times (\mathsf{Upp}^k - \mathsf{Rev}^k)/\mathsf{Upp}^k : k = 1, \ldots, 50\}$. The second, third and fourth columns show the maximum, 75th percentile and 95th percentile of the same data. The second and third blocks of columns in Table 2.1 have the same interpretation as the first block, but the second block focuses on the performance of the neighborhood search heuristic, whereas the third block focuses on the performance of our FPTAS with $\epsilon = 1/2$.

We make three observations in the results in Table 2.1. First, our FPTAS

Par.	FPTAS with $\epsilon = 1/4$				Neighborhood Search Heuristic				FPTAS with $\epsilon = 1/2$			
(P_0, T)	Avg.	Max.	75th	95th	Avg.	Max.	75th	95th	Avg.	Max.	75th	95th
$(N, 0.05)$	0.47%	1.39%	0.63%	1.02%	0.73%	11.59%	0.67%	1.15%	0.47%	1.39%	0.63%	1.02%
$(O, 0.05)$	0.56%	1.77%	0.73%	1.17%	1.34%	12.75%	0.75%	6.69%	0.56%	1.77%	0.73%	1.17%
$(N, 0.1)$	0.71%	1.69%	0.82%	1.12%	2.22%	17.93%	1.01%	14.73%	0.71%	1.69%	0.82%	1.12%
$(O, 0.1)$	0.88%	2.34%	0.92%	2.00%	2.51%	16.24%	2.34%	10.43%	0.88%	2.34%	0.92%	2.00%
$(N, 0.2)$	1.08%	1.76%	1.21%	1.67%	2.61%	13.94%	2.54%	9.59%	1.08%	1.76%	1.21%	1.67%
$(O, 0.2)$	1.39%	2.73%	1.72%	1.95%	2.23%	10.58%	1.76%	6.87%	1.39%	2.73%	1.72%	1.95%
$(N, 0.3)$	1.75%	3.59%	1.95%	3.06%	2.35%	8.33%	2.89%	6.08%	1.75%	3.59%	1.95%	3.06%
$(O, 0.3)$	1.77%	3.54%	2.10%	3.06%	2.15%	7.43%	2.59%	4.98%	1.77%	3.54%	2.10%	3.06%
Avg.	1.08%	2.35%	1.26%	1.88%	2.02%	12.35%	1.82%	7.57%	1.08%	2.35%	1.26%	1.88%

Table 2.1: Performance of our FPTAS and the neighborhood search heuristic.

is able to obtain high quality solutions and its practical performance can be substantially better than its theoretical performance guarantee. When we execute our FPTAS with $\epsilon = 1/4$ corresponding to a performance guarantee of 75%, the average optimality gap of the solutions that we obtain is no larger than 1.08%. Second, the performance of the neighborhood search heuristic noticeably lags behind that of our FPTAS. Also, the heuristic can be unreliable. In particular, there are test problems where we have a gap of as much as 17.93% between the upper bound on the optimal expected revenue and the expected revenue from the solution obtained by the heuristic. Third, the performance of our FPTAS with $\epsilon = 1/4$ and $\epsilon = 1/2$ is indistinguishable up to the two decimal digits that we report, but we occasionally get different solutions with $\epsilon = 1/4$ and $\epsilon = 1/2$. Intuitively, our theoretical analysis accumulates the error in the dynamic program in (2.13) rather conservatively. Thus, even when we use a large value of ϵ corresponding to a crude performance guarantee in our FPTAS, we may still get solutions with high quality, but of course, we cannot give a priori guarantees beyond 75% and 50%, unless we tighten our theoretical analysis further, which appears to be difficult.

Another interesting question is the benefit in the expected revenue that we obtain by letting the customers make choices in two stages. To answer this question, we compute the optimal solution when we offer products in only one of the two stages, but not in the other stage. Since the preference weights of the products in the two stages are the same, this approach is equivalent to computing the optimal set of products to offer in the first stage when we do not offer any products in the second stage. In this case, the expected revenue takes the same form as that under the standard multinomial logit model and there exist efficient algorithms to find the optimal solution; see [41]. For all of the test problems in our experimental setup, we computed the optimal solution when we offer products only in the first stage. We give our numerical results in Table 2.2. The first column in this table shows the parameter setting by using the pair (P_0, T). For test problem k, we let Rev^k be the expected revenue from the solution obtained by our FPTAS with $\epsilon = 1/4$ and One^k be the expected revenue from the optimal solution that we obtain when we offer products only in the first stage. In this case, the second, third, fourth and fifth columns in Table 2.2 show the average, maximum, 75th percentile and 95th percentile of the data $\{100 \times (\mathsf{Rev}^k - \mathsf{One}^k)/\mathsf{Rev}^k : k = 1, \ldots, 50\}$, providing summary statistics for the percent improvement in the expected revenue when we offer products in two stages. Over all of our test problems, offering products in two stages provides an average improvement of 14.53% in the expected revenue. There are test problems where the improvement reaches 39.27%.

The CPU time for the neighborhood search heuristic is on the order of milliseconds. If we use our FPTAS with $\epsilon = 1/2$, then the average CPU time per test problem is 1.34 minutes, whereas if we use our FPTAS with $\epsilon = 1/4$, then the average CPU time per test problem increases by about a factor of ten. In our theoretical running time analysis, decreasing ϵ by a factor of two increases the running time by a factor of eight. The additional CPU time is spent on allocating

Par.	Performance Gap			
(P_0, T)	Avg.	Max.	75th	95th
$(N, 0.05)$	7.71%	32.53%	9.21%	11.99%
$(O, 0.05)$	10.29%	39.27%	10.16%	26.59%
$(N, 0.1)$	12.53%	36.14%	12.43%	25.96%
$(O, 0.1)$	15.72%	36.33%	18.88%	35.14%
$(N, 0.2)$	15.26%	31.95%	17.58%	27.45%
$(O, 0.2)$	20.00%	34.59%	28.68%	34.11%
$(N, 0.3)$	16.61%	29.22%	22.95%	26.01%
$(O, 0.3)$	18.11%	30.08%	24.74%	29.20%
Avg.	14.53%	33.76%	18.08%	27.06%

Table 2.2: Performance improvement when we make offers in two stages rather than in one stage.

memory, which is not accounted for in our theoretical running time analysis. Ultimately, our FPTAS has longer CPU times, but it can provide significantly better solutions than the neighborhood search heuristic. Also, the CPU times for our FPTAS are reasonable if we compute solutions in an offline fashion, which is the case in many applications. Lastly, our FPTAS can continue to provide high solution quality with even larger values of ϵ. In particular, we repeated our numerical experiments with $\epsilon = 3/4$ and the performance of our FPTAS on all our test problems remained almost unchanged, whereas its average CPU time per test problem was 0.19 minutes.

2.9.3 Test Problems for Appointment Slot Choices

In this section, we build on the survey conducted by [14] regarding the appointment slot choices of the patients visiting Farrell Community Health Center in New York City. In their work, [14] use the data provided by the survey to fit a multinomial logit model with a single stage. In our numerical experiments, we augment the data provided by their survey to use it under the multinomial logit model with two stages.

Experimental Setup. We particularly focus on one question in the survey. The question describes a number of symptoms that the patient is hypothetically going through, including heavy cough and sharp chest pain. There are six possible days and three possible time blocks on each day, resulting in 18 possible appointment slots. Among the 18 possible appointment slots, the question offers the patient a randomly chosen set of six appointment slots, along with the option of seeking care elsewhere. The patient picks one of the offered alternatives. In the survey, the patients are offered one set of appointment slots, all in one stage. To use the data provided by the survey under the multinomial logit model with two stages, we artificially augment the data as follows. We randomly split the set of six appointment slots offered to each patient into two partitions, each containing three appointment slots, so that the first partition is offered in the first stage, whereas the second partition is offered in the second stage. So, for each patient, we have the two sets of appointment slots offered in the two stages, along with the choice of the patient. Using maximum likelihood estimation, we fit a multinomial logit model with two stages. [14] focus on the appointment slot choices among the different days, but their survey also includes more granular data on the appointment slot choices among the different time blocks on each day, which is what we use. Also, since we augmented the data, we caution the reader against comparing the results of our numerical experiments with the current operations of the clinic.

In the appointment scheduling setting, a possible objective is to maximize the probability that a patient schedules an appointment, but for purposes of quality and continuity of care, it is preferable to get the patients to schedule earlier appointments. We use two revenue structures in our numerical experiments. In the first revenue structure, we set $r_i = 1$ for all $i \in N$. In this case, we maximize the probability that a patient schedules an appointment without making a distinction between scheduling earlier or later appointments. In the second revenue structure, we use N_1 to denote the appointment slots in the first three days and N_2 to denote the appointment slots in the last three days. We set $r_i = 1$ for all $i \in N_1$ and $r_i = 0.3$ for all $i \in N_2$. In this case, we put a larger weight on scheduling earlier appointments. We use $R \in \{U, E\}$ to denote the revenue structure, where U corresponds to having a uniform revenue of one from each appointment and E corresponds to putting a larger weight on scheduling earlier appointments. Also, recall that we augment the data provided by the survey by randomly splitting the set of appointment slots offered to each patient into two partitions. Using six different random seeds to carry out the splitting process, we obtain six different datasets. We fit a multinomial logit model with two stages to each one of the datasets, yielding six multinomial logit models. The approach that we use to fit the multinomial logit model is similar to the one in [42]. Letting ℓ denote the multinomial logit model that we obtain, varying $R \in \{U, E\}$ and $\ell \in \{1, \ldots, 6\}$, we obtain 12 test problems.

Numerical Results. We give our numerical results in Table 2.3. In this table, the first column shows the parameter setting for each of the 12 test problems by using the pair (R, ℓ) with $R \in \{U, E\}$ and $\ell \in \{1, \ldots, 6\}$. The second column focuses on the performance of our FPTAS with $\epsilon = 1/4$, corresponding to a 75% performance guarantee. In particular, the second column shows the percent gap between the upper bound on the optimal expected revenue and the expected revenue from the solution obtained by our FPTAS. The third column focuses on the expected revenues that we obtain when we offer appointment slots only in the first stage, but not in the second stage. In particular, the third column shows the percent gap between the upper bound on the optimal expected revenue and the expected revenue from the optimal solution under the constraint that we offer appointment slots only in the first stage. We executed our FPTAS with $\epsilon = 1/2$ and $\epsilon = 3/4$, as well as the neighborhood search heuristic. Using $\epsilon = 1/2$ or $\epsilon = 3/4$ for our FPTAS did not change the performance of our FPTAS. In 10 out of the 12 test problems, the performance of the neighborhood search heuristic was identical to that of our FPTAS. In the remaining two test problems, the expected revenue obtained by the neighborhood search heuristic lagged behind that obtained by our FPTAS, but by less than 0.01%. The results in Table 2.3 indicate that our FPTAS can obtain solutions with high quality and its performance in practice can be significantly better than what is predicted by its theoretical performance guarantee. In the table, the solutions obtained by our FPTAS have optimality gaps no larger than 0.79%. On the other hand, offering appointment slots only in the first stage may incur losses in the expected revenues by more than 10%. Closing this section, we note that the goal of the numerical experiments that we give in this section is to test the quality of the solutions obtained by our FPTAS and the benefits provided by taking advantage of the second stage in the choice process. In particular, since the data that we use is based on a survey that offers the sets of appointment slots altogether in one stage and we artificially augment the data to fit a multinomial logit model with two stages, our numerical experiments do not shed light into the benefits of sequential offerings in a clinical setting. Testing the clinical benefits of sequential offerings is outside our scope, which requires a survey where we collect data by actually offering the sets of appointment slots in multiple stages.

Par. (R, ℓ)	FPTAS $\epsilon = 1/4$	Single Stage
$(U, 1)$	0.64%	10.15%
$(E, 1)$	0.74%	10.90%
$(U, 2)$	0.66%	10.24%
$(E, 2)$	0.75%	10.98%
$(U, 3)$	0.65%	10.23%
$(E, 3)$	0.77%	11.00%
Avg.	0.70%	10.58%

Par. (R, ℓ)	FPTAS $\epsilon = 1/4$	Single Stage
$(U, 4)$	0.65%	10.28%
$(E, 4)$	0.75%	11.03%
$(U, 5)$	0.66%	10.36%
$(E, 5)$	0.79%	11.12%
$(U, 6)$	0.63%	10.12%
$(E, 6)$	0.75%	10.91%
Avg.	0.71%	10.64%

Table 2.3: Performance of our FPTAS and the approach that offers appointment slots only in the first stage.

2.10 Conclusions

In this chapter, we studied assortment optimization problems under the multinomial logit model, where the choice process of the customer takes place in multiple stages. There are several directions for future work. The running time of our FPTAS depends on $R_{\max}/R_{\min}$ and $V_{\max}/V_{\min}$. It would be interesting to see whether one can develop a strongly polynomial time algorithm, whose running time does not depend on these quantities, possibly by building on the fact that the union of the optimal assortments offered in all stages is nested by revenue. Also, the running time of our FPTAS is polynomial in the number of products but exponential in the number of stages. A useful line of research is to develop algorithms with running times polynomial in the number of stages. Our effort in this regard has been unfruitful so far and designing algorithms with running times polynomial in the number of stages appears to need a new line of attack. Lastly, as discussed in Section 2.8, our approach does not immediately extend to space constraints within each stage. It is interesting to study the approximation schemes under space constraints within stages.

CHAPTER 3

MULTINOMIAL LOGIT MODEL WITH IMPATIENT CUSTOMERS

3.1 Introduction

A common assumption in traditional revenue management models is that each customer arrives into the system with the intention to purchase a particular product. If this product is available for purchase, then the customer purchases it. Otherwise, the customer leaves the system without a purchase. In many settings, however, the customers do not arrive into the system with the intention to purchase a particular product. Instead, they observe the assortment of available products and choose and substitute within the offered assortment, based on the features and prices of the offered products. In this case, the demand for a product depends on the availability of other products, along with their features and prices. There has been a recent surge in revenue management to use discrete choice models to capture the fact that the customers choose and substitute among the available products. A large portion of these models works under the assumption that the customers view the whole assortment of products offered to them simultaneously, but it is not difficult to give examples where the customers incrementally view the assortment of offered products and they make a purchase decision before viewing all offered products. When purchasing products in online retail, for example, a customer may view the assortment of offered products in multiple webpages. When booking a healthcare appointment on the phone, the patient may be offered appointment slots gradually until she makes a choice. In either one of these examples, the customer may make a purchase or leave without a purchase before viewing all of the offered products. When the customers incrementally view the assortment of offered products, the question is not only what assortment of products to offer but also in which sequence to offer the products.

We give a variant of the multinomial logit model where the customers incrementally view the assortment of offered products in multiple stages. We study assortment optimization and pricing problems under this choice model. In the choice model, each customer has a different patience level sampled from a distribution, which determines the maximum number of stages for which she is willing to view the assortment of products. In each stage, if the utility of a product viewed so far is larger than the utility of the outside option, then the customer purchases this product. Otherwise, the customer views the assortment of products in the next stage, as long as her patience level allows her to do so. The customers are impatient for two reasons. First, a customer purchases a product in the current stage as soon as its utility exceeds a minimum acceptable utility captured by the outside option, even though there may be a product with a larger utility in a subsequent stage. Second, a customer may run out of patience and leave without viewing the whole assortment.

Main Contributions: Our main contributions are in the formulation of the choice model as well as solving the corresponding assortment optimization and pricing problems.

Multinomial Logit Model with Impatient Customers. We give a new variant of the multinomial logit model with impatient customers that has a closed-form expression for the choice probabilities. The choice model is based on random utilities. A customer is characterized by the utilities she attaches with the products, a minimum acceptable utility and a patience level. We refer to the minimum acceptable utility as the utility of the outside option. The utilities of the products and the outside option are independent and have the Gumbel distribution with the same scale parameter. The patience level of the customer has a general distribution over the support $\{1, \ldots, m\}$. The customer incrementally views the assortment of offered products in multiple stages. In each stage, if the customer finds a product with utility exceeding the utility of the outside option, then she purchases the product. If the customer cannot find such a product before she runs out of patience, then the customer leaves without a purchase. Since the patience level of a customer is at most m, we need to choose the assortments offered over m stages. We give a closed-form expression for the choice probability of a product under any assortment. This choice probability is significantly different from the one under the standard multinomial logit model.

Assortment Optimization. In assortment optimization, each product has a fixed revenue and the goal is to find a revenue-maximizing sequence of assortments to offer. We give an efficient algorithm for the assortment optimization problem by using the following steps. First, we show that there exists a revenue-ordered optimal solution. That is, letting n be the number of products and r_i be the revenue of product i, indexing the products so that $r_1 \geq r_2 \geq \ldots \geq r_n$, the optimal assortment to offer in stage k is of the form $\{j_k^* + 1, \ldots, j_{k+1}^*\}$ for some $0 = j_1^* \leq j_2^* \leq \ldots \leq j_m^* \leq j_{m+1}$. Second, knowing the optimality of a revenue-ordered solution does not immediately yield an efficient algorithm since the number of possible choices of $0 = j_1^* \leq j_2^* \leq \ldots \leq j_m^* \leq j_{m+1}$ increases exponentially with the number of stages. We give a dynamic program that finds the best sequence of revenue-ordered assortments to offer. Under the standard multinomial logit model, the expected revenue can be expressed as a fraction where both the numerator and denominator are linear in the decision variables. In our case, the denominator is quadratic in the decision variables and the optimality of revenue-ordered assortments does not follow from existing results.

Joint Pricing and Assortment Optimization. In the pricing setting, the mean utility of a product depends on its price. Following the standard assumption in the pricing literature, we assume that the products have the same price sensitivity, which is reasonable when the customers make a choice among the products in the same category. We begin with the case where the sequence of offered assortments is fixed and the goal is to find the revenue-maximizing prices. The expected revenue is not concave in the prices, but we give a reformulation of the pricing problem where the decision variables are the probabilities that a customer reaches different stages. We show that the expected revenue is concave in these decision variables and we can recover the optimal prices by using optimal probabilities that a customer reaches different stages. Once we know how to solve the pricing problem for a fixed assortment, we consider the case where both the sequence of offered assortments and prices are decision variables. We give a greedy search heuristic that is guaranteed to obtain at least 50% of the optimal expected revenue. This heuristic exploits the fact that we can compute the revenue-maximizing prices for a fixed assortment.

Space Constraints. We consider the assortment optimization problem when each product occupies a certain amount of space and there is a constraint on the total space consumption of the offered products. We give a fully polynomial-time approximation scheme (FPTAS). We show that we can improve the running time of our FPTAS where there is a constraint on the total number of offered products. We also give an exact algorithm when the number of stages is fixed. A constraint on the total space consumption or the total number of offered products may arise, for example, when we want to avoid overwhelming a patient with too many appointment slot options. Similarly, offering a product may require some form of capital investment, which could be in the form of a copyright, so we may want to limit the total investment. We close the chapter with computational experiments, where we use a dataset from Expedia to demonstrate that our choice model with multiple stages can predict the purchases better, against the standard multinomial logit model benchmark. Also, we test the practical performance of our greedy search heuristic and FPTAS.

To our knowledge, this chapter is the first variant of the multinomial logit model that allows viewing the offered products incrementally, while ensuring that the choice model is consistent with the random utility framework. In many online retail settings, the products are offered in multiple webpages, but the number of products included in a webpage is at the discretion of the retailer, since the products are simply presented as a list, as in, for example, Amazon. Our unconstrained and constrained assortment problems, as well as our joint pricing and assortment optimization problem, find applications in such settings. Our choice model is motivated by the satisficing behavior of customers especially when purchasing leisure products, such as airline tickets and hotel rooms, where the customer proceeds to purchasing a product once the utility of the product exceeds a minimum acceptable utility. Naturally, such a choice model may not be appropriate in all settings. Lastly, we do not need to dynamically adjust the assortment or prices based on the knowledge that a customer reaches a particular stage. We can interpret the assortment or prices at stage k as decisions made conditional on the fact that a customer reaches stage k. Our work still leaves several open questions. The list in each webpage may not provide any flexibility, so we may consider a constraint on the number of products offered in each stage. Our efforts to address this problem was unproductive. Also, we give an exact algorithm when there is a constraint on the total number of products offered in all stages and the number of stages is fixed, but we do not know the complexity of the problem when the number of stages is an input.

Literature Review: There is recent assortment optimization and pricing work, where customers view only a portion of the offered assortment either due to search behavior or consideration sets. [20] build a product display optimization model where the customers decide on the number of webpages to view based on an exogenous distribution and choose within the entire assortment in these webpages according to a general choice model. [45] consider a model where the customers focus on a portion of the products by trading off their expected utility from the purchase with the search effort, but they do not view the assortment incrementally. [8] focus on a product ranking problem where the customers build a consideration set as a function of the search cost. The authors focus on maximizing the market share or welfare, but not expected revenue. [3] study an assortment problem where each customer views a random number of webpages and makes a choice within these webpages according to the multinomial logit model. The customers do not view the products sequentially. [2] focus on an assortment setting where each product is included in the consideration set of a customer with a fixed probability. [15] use a choice model based on preference lists, where the preference list of each customer is short, corresponding to the case with small consideration sets. In all papers in this paragraph so far, the assortment optimization problems are NP-hard and the authors propose approximation methods. [35] study a pricing problem based on a cascade model for viewing the products. The sequence of offered products is fixed and the authors do not consider finding the optimal sequence.

There is significant literature on assortment optimization under the standard multinomial logit model. [20] and [41] show that it is optimal to offer a revenue-ordered assortment under the standard multinomial logit model, but as discussed earlier, their approaches do not extend to our setting. [36], [43], [26] and [40] study the problem under various constraints on the offered assortment. [6], [33] and [37] consider the problem under a mixture of multinomial logit models. [30] work on a multi-stage assortment problem but their choice model is a heuristic extension of the multinomial logit model, which is not compatible with the random utility framework. Also, their problem is computationally intractable. [18] study a two-stage multinomial logit model, where the products that can be offered in each of the two stages are fixed a priori.

For pricing under the standard multinomial logit model, [39] show that the expected revenue is concave in the product market shares. [25] and [28] show that the optimal prices possess the constant mark-up property, where the optimal price of each product exceeds its marginal cost by a constant. [46] show that both of these results hold under all generalized extreme value models. Similar to our work, all of these papers assume that the products have the same price sensitivity. We limit our discussion to the work under the multinomial logit model, but we refer to [13], [7], [23], [1], [5], [9] and [29] for representative approaches under other choice models.

Organization: In Section 3.2, we derive the choice probabilities under our choice model. In Section 3.3, we consider the unconstrained assortment problem. In Section 3.4, we study the joint pricing and assortment problem. In Section 3.5, we consider the case where there is a constraint on the total space consumption of the offered products. In Section 3.6, we give computational experiments.

3.2 Multinomial Logit Model with Impatient Customers

The set of products is $\mathcal{N} = \{1, \ldots, n\}$ and the set of stages is $\mathcal{M} = \{1, \ldots, m\}$. We use $(S_1, \ldots, S_m)$ to denote the sequence of assortments that we offer over all m stages, where $S_k \subseteq \mathcal{N}$ is the assortment that we offer in stage k. The assortments that we offer in different stages are disjoint, so we have $S_k \cap S_\ell = \emptyset$ for all $k \neq \ell$. The utility of product i is given by the random variable U_i, which has the Gumbel distribution with location-scale parameters $(\mu_i, 1)$. Letting $v_i = e^{\mu_i}$ for notational brevity, we refer to v_i as the preference weight of product i. The utility of the outside option is given by the random variable U_0, which has the Gumbel distribution with location-scale parameters $(0, 1)$. The patience level of a customer is given by the random variable Y taking values in $\mathcal{M}$, where a customer with patience level k is willing to view the assortments in the first k stages. For notational brevity, we let $\lambda_k = \mathbb{P}\{Y \geq k\}$. Note that $1 = \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m > 0$. The random variables $\{U_i : i \in \mathcal{N}\}$, U_0 and Y are independent. In our choice model, an arriving customer is characterized by the utilities she associates with the different products and the outside option, along with her patience level, all sampled from their corresponding distributions.

As stated in the introduction, the utility of the outside option corresponds to the minimum acceptable utility for the customer. A customer chooses among the products by sequentially viewing the assortments in each stage. If the largest utility for a product in stage k exceeds the utility of the outside option, then the customer purchases this product. Otherwise, the customer moves on to stage $k+1$. If stage $k+1$ is beyond the patience level of the customer, then the customer leaves without a purchase, whereas if stage $k+1$ is within the patience level of the customer, then the customer views the products in stage $k+1$. The customers are impatient for two reasons. First, if the customer finds a product in stage k with utility exceeding the utility of the outside option, then she purchases this product right away, even though there may be a product in a subsequent stage with a larger utility. Second, due to her patience level, a customer does not necessarily view all assortments in all stages. As a function of the assortments $(S_1, \ldots, S_m)$, let $\phi_i^k(S_1, \ldots, S_m)$ be the probability that a customer chooses product $i \in S_k$. In the next theorem, we give an expression for this choice probability. Below and throughout the chapter, we let $V(S) = \sum_{i \in S} v_i$.

Theorem 3.2.1 (Choice Probabilities) *If we offer assortments $(S_1, \ldots, S_m)$ over m stages, then a customer purchases product $i \in S_k$ with probability*

$$\phi_i^k(S_1, \ldots, S_m) = \frac{\lambda_k \, v_i}{(1 + \sum_{\ell=1}^{k-1} V(S_\ell))\,(1 + \sum_{\ell=1}^{k} V(S_\ell))}.$$

Proof: Letting X_1 and X_2 be independent random variables having the Gumbel distribution with location-scale parameters $(\mu_1, 1)$ and $(\mu_2, 1)$, the proof uses three properties of Gumbel random variables. First, the random variable $\max\{X_1, X_2\}$ has the Gumbel distribution with location-scale parameters $(\log(e^{\mu_1} + e^{\mu_2}), 1)$. Second, we have $\mathbb{P}\{X_1 \geq X_2\} = e^{\mu_1}/(e^{\mu_1} + e^{\mu_2})$. Third, using $\mathbf{1}(\cdot)$ to denote the indicator function, the random variables $\max\{X_1, X_2\}$ and $\mathbf{1}(X_1 \geq X_2)$ are independent. For a customer to purchase product $i \in S_k$, her patience level must be at least k, the utility of the outside option must exceed the utilities of all products in stages $1, \ldots, k-1$ and the utility of product i must exceed both the utility of the outside option and the utilities of all other products offered in stage k. Thus, we have

$$\begin{aligned}\phi_i^k(S_1, \ldots, S_k) &= \mathbb{P}\Big\{Y \geq k\Big\} \times \mathbb{P}\Big\{U_0 \geq \max_{j \in S_1 \cup \ldots \cup S_{k-1}} U_j,\ U_i \geq \max\Big\{U_0, \max_{j \in S_k \setminus \{i\}} U_j\Big\}\Big\} \\ &= \lambda_k \times \mathbb{P}\Big\{U_0 \geq \max_{j \in S_1 \cup \ldots \cup S_{k-1}} U_j\Big\} \times \mathbb{P}\Big\{U_i \geq \max\Big\{U_0, \max_{j \in S_k \setminus \{i\}} U_j\Big\} \,\Big|\, U_0 \geq \max_{j \in S_1 \cup \ldots \cup S_{k-1}} U_j\Big\}.\end{aligned} \tag{3.1}$$

Letting $\widehat{U}_{k-1} = \max_{j \in S_1 \cup \ldots \cup S_{k-1}} U_k$ and $\widetilde{U}_k = \max_{j \in S_k \setminus \{i\}} U_j$, by the first property, $\widehat{U}_{k-1}$ and $\widetilde{U}_k$ are Gumbel with location-scale parameters $(\log \sum_{j \in S_1 \cup \ldots \cup S_{k-1}} e^{\mu_j}, 1)$ and $(\log \sum_{j \in S_k \setminus \{i\}} e^{\mu_j}, 1)$.

Noting that random variables U_i, U_0, $\widehat{U}_{k-1}$ and $\widetilde{U}_k$ are independent, considering the second probability on the right side of (3.1), we have

$$\begin{aligned}\mathbb{P}\Big\{U_i \geq \max\{U_0, \widetilde{U}_k\} \,\Big|\, U_0 \geq \widehat{U}_{k-1}\Big\} &= \mathbb{P}\Big\{U_i \geq \max\{U_0, \widetilde{U}_k, \widehat{U}_{k-1}\} \,\Big|\, U_0 \geq \widehat{U}_{k-1}\Big\} \\ &= \mathbb{P}\Big\{U_i \geq \max\{U_0, \widehat{U}_{k-1}\},\ U_i \geq \widetilde{U}_k \,\Big|\, U_0 \geq \widehat{U}_{k-1}\Big\} \\ &= \mathbb{P}\Big\{U_i \geq \max\{U_0, \widehat{U}_{k-1}\} \,\Big|\, U_0 \geq \widehat{U}_{k-1}\Big\} \\ &\qquad \times \mathbb{P}\Big\{U_i \geq \widetilde{U}_k \,\Big|\, U_i \geq \max\{U_0, \widehat{U}_{k-1}\},\ U_0 \geq \widehat{U}_{k-1}\Big\} \\ &\overset{(a)}{=} \mathbb{P}\Big\{U_i \geq \max\{U_0, \widehat{U}_{k-1}\}\Big\} \times \mathbb{P}\Big\{U_i \geq \widetilde{U}_k \,\Big|\, U_i \geq \max\{U_0, \widehat{U}_{k-1}\}\Big\} \\ &= \mathbb{P}\Big\{U_i \geq \max\{U_0, \widehat{U}_{k-1}\},\ U_i \geq \widetilde{U}_k\Big\} \\ &= \mathbb{P}\Big\{U_i \geq \max\{U_0, \widetilde{U}_k, \widehat{U}_{k-1}\}\Big\} \overset{(b)}{=} \frac{e^{\mu_i}}{1 + \sum_{j \in S_1 \cup \ldots \cup S_k} e^{\mu_j}} = \frac{v_i}{1 + \sum_{\ell=1}^k V(S_\ell)},\end{aligned} \tag{3.2}$$

where (a) uses the fact that $\max\{U_0, \widehat{U}_{k-1}\}$ and $\mathbf{1}(U_0 \geq \widehat{U}_{k-1})$ are independent, as well as U_i, U_0, $\widehat{U}_{k-1}$ and $\widetilde{U}_k$ are independent, whereas (b) uses the first and second properties above.

Considering the first probability on the right side of (3.1), using the second property and the fact that $v_i = e^{\mu_i}$, we get $\mathbb{P}\{U_0 \geq \widehat{U}_{k-1}\} = 1/(1 + \sum_{j \in S_1 \cup \ldots \cup S_{k-1}} e^{\mu_j}) = 1/(1 + \sum_{\ell=1}^{k-1} V(S_\ell))$. ■

By Theorem 3.2.1, for a given sequence of assortments $(S_1, \ldots, S_m)$, we have a closed-form expression for the probability that a customer chooses each one of the products. We can interpret our choice model with impatient customers as a version of the multinomial logit model with multiple customer types, where the customer types differ in their patience levels, but not in the distributions of the utilities that they associate with the products. Next, we focus on the assortment optimization problem under this choice model and show that we can efficiently compute the sequence of assortments to offer so that we maximize the expected revenue.

3.3 Unconstrained Assortment Optimization

We use $r_i > 0$ to denote the revenue of product i. Letting $W(S) = \sum_{i \in S} r_i v_i$, if we offer assortments $(S_1, \ldots, S_m)$ over m stages, then the expected revenue from a customer is

$$\begin{aligned}
\Pi(S_1, \ldots, S_m) &= \sum_{k \in \mathcal{M}} \sum_{i \in S_k} r_i \, \phi_i^k(S_1, \ldots, S_m) \\
&= \sum_{k \in \mathcal{M}} \sum_{i \in S_k} \frac{\lambda_k \, r_i \, v_i}{(1 + \sum_{\ell=1}^{k-1} V(S_\ell))\,(1 + \sum_{\ell=1}^{k} V(S_\ell))} \\
&= \sum_{k \in \mathcal{M}} \frac{\lambda_k \, W(S_k)}{(1 + \sum_{\ell=1}^{k-1} V(S_\ell))\,(1 + \sum_{\ell=1}^{k} V(S_\ell))}.
\end{aligned} \tag{3.3}$$

The assortments offered over m stages are disjoint, so the set of feasible solutions is $\mathcal{F} = \{(S_1, \ldots, S_m) : S_k \subseteq \mathcal{M} \; \forall \, k \in \mathcal{M}, \; S_k \cap S_\ell = \emptyset \; \forall \, k \neq \ell\}$. We want to solve the problem

$$\max_{(S_1, \ldots, S_m) \in \mathcal{F}} \Pi(S_1, \ldots, S_m). \tag{ASSORTMENT}$$

In this section, we show that there exists an optimal solution to the ASSORTMENT problem that is revenue-ordered. In particular, we index the products in the order of decreasing revenues so that $r_1 \geq r_2 \geq \ldots \geq r_n$. In this case, there exists an optimal solution $(S_1^*, \ldots, S_m^*)$ such that $S_k^* = \{j_k^* + 1, \ldots, j_{k+1}^*\}$ for $j_1^*, \ldots, j_{m+1}^*$ that satisfy $0 = j_1^* \leq j_2^* \leq \ldots \leq j_{m+1}^*$. Thus, the assortment offered in each stage follows the order of the revenues of the products. Noting $j_1^* = 0$, the choice of the products $j_2^*, \ldots, j_{m+1}^*$ determines an optimal solution to the ASSORTMENT problem. Knowing that there exists an optimal solution that is revenue-ordered reduces the number of possible optimal solutions to $O(n^m)$, which is polynomial in n but exponential in m. We find a revenue-ordered solution that maximizes the expected revenue by solving a dynamic program in $O(mn^2)$ operations. Therefore, we can solve the ASSORTMENT problem efficiently.

For two solutions $(S_1, \dots, S_m)$ and $(T_1, \dots, T_m)$, we say that the solution $(S_1, \dots, S_m)$ dominates the solution $(T_1, \dots, T_m)$ if $|S_1| = |T_1| \dots |S_k| = |T_k|$ and $|S_{k+1}| > |T_{k+1}|$ for some $k \in \mathcal{M}$. Intuitively speaking, a dominating solution offers an assortment with a larger cardinality in an earlier stage. If there are multiple optimal solutions for the ASSORTMENT problem, then we choose an optimal solution non-dominated by any other optimal solution. Note that if $(S_1^*, \dots, S_m^*)$ is a non-dominated optimal solution, then we must have $S_1^* \neq \emptyset, \dots, S_k^* \neq \emptyset$ and $S_{k+1}^* = \emptyset, \dots, S_m^* = \emptyset$ for some $k \in \mathcal{M}$. In particular, if $S_\ell^* = \emptyset$ and $S_{\ell+1}^* \neq \emptyset$, then the solution $(S_1^*, \dots, S_{\ell+1}^*, S_\ell^*, \dots, S_m^*)$ dominates the solution $(S_1^*, \dots, S_m^*)$. Also, using the fact that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m$, noting the expected revenue in (3.3), it is simple to check that the expected revenue from the solution $(S_1^*, \dots, S_{\ell+1}^*, S_\ell^*, \dots, S_m^*)$ is at least as large as the one from the solution $(S_1^*, \dots, S_m^*)$.

To show that there exists a revenue-ordered optimal solution for the ASSORTMENT problem, we construct revenue thresholds for each stage such that if the revenue of a product falls within the thresholds corresponding to stage k, then it is optimal to offer the product in stage k. In particular, the next theorem is the main result of this section. The proof of the theorem is based on an intermediate lemma, which we give after the theorem.

Theorem 3.3.1 (Optimal Revenue-Ordered Assortments) *There exists an optimal solution $(S_1^*, \dots, S_m^*)$ to the* ASSORTMENT *problem such that $S_k^* = \{i \in \mathcal{N} : t_{k+1}^* \leq r_i < t_k^*\}$ for some revenue thresholds $t_1^*, \dots, t_{m+1}^*$ that satisfy $+\infty = t_1^* \geq t_2^* \geq \dots \geq t_{m+1}^*$.*

To construct the revenue thresholds given in the theorem above, letting $R_k(S_1, \ldots, S_m) = \frac{\lambda_k\, W(S_k)}{(1+\sum_{\ell=1}^{k-1} V(S_\ell))\,(1+\sum_{\ell=1}^{k} V(S_\ell))}$, we define

$$t_k(S_1, \ldots, S_m) \;=\; \frac{R_{k-1}(S_1, \ldots, S_m) + R_k(S_1, \ldots, S_m)}{\frac{\lambda_{k-1}}{1+\sum_{\ell=1}^{k-2} V(S_\ell)} - \frac{\lambda_k}{1+\sum_{\ell=1}^{k} V(S_\ell)}} \qquad \forall\, k \in \mathcal{M} \setminus \{1\},$$

$$t_{m+1}(S_1, \ldots, S_m) \;=\; \frac{R_m(S_1, \ldots, S_m)}{\frac{\lambda_m}{1+\sum_{\ell=1}^{m-1} V(S_\ell)}}.$$

We follow the convention that $t_1(S_1, \ldots, S_m) = +\infty$. In the next lemma, we quantify the change in expected revenue when we move a product from one stage to another.

Lemma 3.3.2 (Product Exchange) *For each sequence of assortments $(S_1, \ldots, S_m) \in \mathcal{F}$ offered over m stages, we have the identities*

- $\Pi(S_1, \ldots, S_{k-1} \cup \{i\}, S_k \setminus \{i\}, \ldots, S_m) - \Pi(S_1, \ldots, S_m)$
$$= \frac{\frac{\lambda_{k-1}}{1+\sum_{\ell=1}^{k-2} V(S_\ell)} - \frac{\lambda_k}{1+\sum_{\ell=1}^{k} V(S_\ell)}}{1 + \sum_{\ell=1}^{k-1} V(S_\ell) + v_i}\, v_i\, (r_i - t_k(S_1, \ldots, S_m)),$$
- $\Pi(S_1, \ldots, S_k \setminus \{i\}, S_{k+1} \cup \{i\} \ldots, S_m) - \Pi(S_1, \ldots, S_m)$
$$= \frac{\frac{\lambda_k}{1+\sum_{\ell=1}^{k-1} V(S_\ell)} - \frac{\lambda_{k+1}}{1+\sum_{\ell=1}^{k+1} V(S_\ell)}}{1 + \sum_{\ell=1}^{k} V(S_\ell) - v_i}\, v_i\, (t_{k+1}(S_1, \ldots, S_m) - r_i),$$
- $\Pi(S_1, \ldots, S_{m-1}, S_m \setminus \{i\}) - \Pi(S_1, \ldots, S_m)$
$$= \frac{\frac{\lambda_m}{1+\sum_{\ell=1}^{m-1} V(S_\ell)}}{1 + \sum_{\ell=1}^{m} V(S_\ell) - v_i}\, v_i\, (t_{m+1}(S_1, \ldots, S_m) - r_i).$$

The proof of the lemma follows directly by evaluating the changes in the expected revenue by using (3.3). We give the proof in Appendix B.1. Noting that $\lambda_k \geq \lambda_{k+1}$ and $\sum_{\ell=1}^{k} V(S_\ell) \leq \sum_{\ell=1}^{k+1} V(S_\ell)$, by the lemma above, we can compare r_i only with $t_k(S_1, \ldots, S_m)$ or $t_{k+1}(S_1, \ldots, S_m)$ to decide whether moving product i from stage k to stage $k-1$ or $k+1$ provides an improvement in the expected revenue. Below is the proof of Theorem 3.3.1.

Proof of Theorem 3.3.1: Let $(S_1^*, \ldots, S_m^*)$ be a non-dominated optimal solution to the ASSORTMENT problem. By our earlier discussion, there exists $\ell \in \mathcal{M}$ such that $S_1^* \neq \emptyset, \ldots, S_\ell^* \neq \emptyset, S_{\ell+1}^* = \emptyset, \ldots, S_m^* = \emptyset$. Thus, a customer does not make a purchase after stage ℓ. In this case, if we consider the ASSORTMENT problem with only ℓ stages, then $(S_1^*, \ldots, S_\ell^*)$ must be a non-dominated optimal solution to this problem. For all $k = 1, \ldots, \ell$, we let $t_k^* = t_k(S_1^*, \ldots, S_\ell^*)$ and focus on the ASSORTMENT problem with ℓ stages, where the set of stages is $\mathcal{L} = \{1, \ldots, \ell\}$.

We claim that $S_k^* \subseteq \{i \in \mathcal{N} : t_{k+1}^* \leq r_i < t_k^*\}$ for all $k \in \mathcal{L}$. In particular, if $i \in S_k^*$ and $r_i \geq t_k^*$, then the first part of Lemma 3.3.2 implies that moving product i from assortment S_k^* to S_{k-1}^* does not degrade the expected revenue from the solution $(S_1^*, \ldots, S_\ell^*)$, which contradicts the fact that $(S_1^*, \ldots, S_\ell^*)$ is a non-dominated optimal solution. For $k = 1$, we cannot have $r_i \geq t_1^*$ since $t_1^* = +\infty$. If $i \in S_k^*$ and $r_i < t_{k+1}^*$, then the second part of Lemma 3.3.2 implies that moving product i from assortment S_k^* to S_{k+1}^* strictly increases the expected revenue from the solution $(S_1^*, \ldots, S_\ell^*)$, which contradicts the fact that $(S_1^*, \ldots, S_\ell^*)$ is an optimal solution. For $k = \ell$, if $i \in S_\ell^*$ and $r_i < t_{\ell+1}^*$, then the third part of Lemma 3.3.2 implies that removing product i from assortment S_ℓ^* strictly improves the expected revenue from the solution $(S_1^*, \ldots, S_\ell^*)$. Thus, the claim follows. Also, by the preceding discussion, if $i \in S_k^*$, then $t_{k+1}^* \leq r_i < t_k^*$. For all $k \in \mathcal{L}$, assortment $S_k^* \neq \emptyset$ includes at least one product. So, we get $t_{k+1}^* \leq t_k^*$ for all $k \in \mathcal{L}$.

Next, we claim that $S_k^* \supseteq \{i \in \mathcal{N} : t_{k+1}^* \leq r_i < t_k^*\}$ for all $k \in \mathcal{L}$. In particular, if $t_{k+1}^* \leq r_i < t_k^*$ and $i \notin S_k^*$ for some $k \in \mathcal{L}$, then it must be the case that $r_i \notin S_q^*$ for all $q \in \mathcal{L}$, because by the discussion in the previous paragraph, we have $S_q^* \subseteq \{i \in \mathcal{N} : t_{q+1}^* \leq r_i < t_q^*\}$ for all $q \in \mathcal{L}$ and $+\infty = t_1^* \geq t_2^* \geq \ldots \geq t_{\ell+1}^*$. In this case, using the fact that $r_i \geq t_{k+1}^* \geq t_{\ell+1}^*$, replacing the preference weight of product

i in the third part of Lemma 3.3.2 with $-v_i$, we notice that adding product i to assortment S^*_ℓ does not degrade the expected revenue from the solution $(S^*_1, \ldots, S^*_\ell)$, which contradicts the fact that $(S^*_1, \ldots, S^*_\ell)$ is a non-dominated optimal solution to the ASSORTMENT problem with ℓ stages. Thus, the claim follows. By the discussion so far, we have $S^*_k = \{i \in \mathcal{N} : t^*_{k+1} \leq r_i < t^*_k\}$ for all $k \in \mathcal{L}$ and $+\infty = t^*_1 \geq t^*_2 \geq \ldots \geq t^*_{\ell+1}$. For the problem with m stages, noting that $S^*_k = \emptyset$ for all $k = \ell + 1, \ldots, m$, we set $t^*_k = t^*_{\ell+1}$ for all $k = \ell + 2, \ldots, m + 1$, so we have $S^*_k = \{i \in \mathcal{N} : t^*_{k+1} \leq r_i < t^*_k\}$ for all $k \in \mathcal{M}$ and $+\infty = t^*_1 \geq t^*_2 \geq \ldots \geq t^*_{m+1}$. ■

By Theorem 3.3.1, we can simply consider solutions $(S_1, \ldots, S_m)$ of the form $S_k = \{j_k + 1, \ldots, j_{k+1}\}$ for $j_1, \ldots, j_{m+1}$ that satisfy $0 = j_1 \leq j_2 \leq \ldots \leq j_{m+1}$. Note that if we offer assortments of this form, then we have $S_1 \cup \ldots \cup S_{k-1} = \{1, \ldots, j_k\}$ and $\sum_{\ell=1}^{k-1} V(S_\ell) = V(\{1, \ldots, j_k\})$. Therefore, we can solve a dynamic program to decide which assortments of this form to offer in each stage so that we maximize the expected revenue. The decision epochs correspond to the stages. The state variable at decision epoch k is the value of j such that the assortments $S_1, \ldots, S_k$ offered in the previous stages satisfy $S_1 \cup \ldots \cup S_{k-1} = \{1, \ldots, j\}$. The action at decision epoch k is the value of p such that the assortment offered in stage k is $\{j + 1, \ldots, p\}$. To find an assortment to offer in each stage to maximize the expected revenue, we can solve the dynamic program

$$J_k(j) = \max_{p \in \{j, \ldots, n\}} \left\{ \frac{\lambda_k \, W(\{j+1, \ldots, p\})}{(1 + V(\{1, \ldots, j\}))\,(1 + V(\{1, \ldots, p\}))} + J_{k+1}(p) \right\}$$

with the boundary condition that $J_{m+1}(\cdot) = 0$. There are m decision epochs, n possible states and n possible actions, so we can solve the dynamic program in $O(mn^2)$ operations.

3.4 Joint Pricing and Assortment Optimization

We consider the joint pricing and assortment optimization problem, where we choose the assortment of products to offer in each stage, as well as the prices of the products.

3.4.1 Optimal Prices under Fixed Assortments

In this section, we study the case where the assortments $(S_1, \ldots, S_m)$ offered over m stages are fixed and we choose the prices of the products to maximize the expected revenue. Although the expected revenue is not concave in the prices, we give a convex program to solve the pricing problem. Building on our results in this section, we will give a heuristic for the joint pricing and assortment optimization problem that is guaranteed to obtain at least 50% of the optimal expected revenue. We use p_i to denote the price for product i. For fixed parameters α_i and β, if we charge the price p_i for product i, then the utility of product i has the Gumbel distribution with location-scale parameters $(\alpha_i - \beta p_i, 1)$. The mean of a Gumbel random variable with location-scale parameters $(\mu_i, 1)$ is $\mu_i + \gamma$, where γ is the Euler constant. Therefore, the mean utility of a product linearly depends on its price. The parameter α_i captures the intrinsic mean utility of product i, whereas β captures the sensitivity of the mean utility to price. If we charge the price p_i for product i, then the preference weight of the product is given by $e^{\alpha_i - \beta p_i}$. As a function of the prices $\boldsymbol{p} = (p_1, \ldots, p_n)$ that we charge for the products, we let $V_k(\boldsymbol{p}) = \sum_{i \in S_k} e^{\alpha_i - \beta p_i}$ to capture the total preference weight of the products in stage k. Since the assortment of products offered in each stage is fixed, we do not make the dependence of $V_k(\boldsymbol{p})$ on the assortment S_k explicit. Thus, by the discussion in Section 3.2, if the prices of the products are given by $\boldsymbol{p}$, then a customer chooses product $i \in S_k$ with probability $\phi_i^k(\boldsymbol{p}) = \lambda_k \, e^{\alpha_i - \beta \, p_i} / ((1 + \sum_{\ell=1}^{k-1} V_\ell(\boldsymbol{p}))\,(1 + \sum_{\ell=1}^{k} V_\ell(\boldsymbol{p})))$. As a function of the prices $\boldsymbol{p}$, the expected revenue that we obtain from a customer is

$$\Pi(\boldsymbol{p}) \;=\; \sum_{k \in \mathcal{M}} \sum_{i \in S_k} p_i \, \phi_i^k(\boldsymbol{p}) \;=\; \sum_{k \in \mathcal{M}} \frac{\lambda_k \sum_{i \in S_k} p_i \, e^{\alpha_i - \beta \, p_i}}{(1 + \sum_{\ell=1}^{k-1} V_\ell(\boldsymbol{p}))\,(1 + \sum_{\ell=1}^{k} V_\ell(\boldsymbol{p}))}.$$

In the pricing literature, it is customary to include a marginal cost c_i for product i so that the objective function above reads $\sum_{k \in \mathcal{M}} \sum_{i \in S_k} (p_i - c_i)\, \phi_i^k(\boldsymbol{p})$. Note that including a marginal cost for product i is equivalent to shifting the price of product i by c_i and the constant α_i by $\beta \, c_i$. Our goal is to find the product prices to maximize the expected revenue, yielding the problem

$$\max_{\boldsymbol{p} \in \mathbb{R}^n} \; \Pi(\boldsymbol{p}). \qquad \text{(PRICING)}$$

Next, we show that the prices in a particular stage are the same in an optimal solution to the PRICING problem. We use this result to give a convex program to solve the PRICING problem.

Stage-Specific Optimal Prices:

Throughout the rest of section, we let $q_k(\boldsymbol{p}) = 1/(1 + \sum_{\ell=1}^{k} V_\ell(\boldsymbol{p}))$ with the convention that $q_0(\boldsymbol{p}) = 1$. By the first and second properties of Gumbel random variables discussed in the proof of Theorem 3.2.1, $q_k(\boldsymbol{p})$ is the probability that the utility of the outside option exceeds the utility of all products that are offered in the first k stages. We refer to $q_k(\boldsymbol{p})$ as the no-purchase probability over the first k stages with the understanding that this probability is the no-purchase probability for a customer with patience level exceeding k.

Theorem 3.4.1 (Stage-Specific Optimal Prices) *There exists an optimal solution $\boldsymbol{p}^*$ to the* PRICING *problem such that if $i, j \in S_k$ for some $k \in \mathcal{M}$, then $p_i^* = p_j^*$.*

Proof: Since $V_\ell(\boldsymbol{p}) = \sum_{j\in S_\ell} e^{\alpha_j-\beta p_j}$, for $j \in S_\ell$, we have $\partial q_k(\boldsymbol{p})/\partial p_j = \beta\, e^{\alpha_j-\beta p_j}\, q_k^2(\boldsymbol{p})$ if $\ell \le k$, but $\partial q_k(\boldsymbol{p})/\partial p_j = 0$ if $\ell > k$. Noting that $\phi_i^k(\boldsymbol{p}) = \lambda_k\, e^{\alpha_i-\beta p_i}\, q_k(\boldsymbol{p})\, q_{k-1}(\boldsymbol{p})$, for $i \in S_k$ and $j \in S_\ell$, we get

$$\frac{\partial \phi_i^k(\boldsymbol{p})}{\partial p_j} = \begin{cases} -\lambda_k\,\beta\, e^{\alpha_i-\beta p_i}\, q_k(\boldsymbol{p})\, q_{k-1}(\boldsymbol{p}) + \lambda_k\, e^{\alpha_i-\beta p_i}\,\beta\, e^{\alpha_i-\beta p_i}\, q_k^2(\boldsymbol{p})\, q_{k-1}(\boldsymbol{p}) & \text{if } \ell = k,\ j = i \\ \lambda_k\, e^{\alpha_i-\beta p_i}\,\beta\, e^{\alpha_j-\beta p_j}\, q_k^2(\boldsymbol{p})\, q_{k-1}(\boldsymbol{p}) & \text{if } \ell = k,\ j \ne i \\ \lambda_k\, e^{\alpha_i-\beta p_i}\Big\{\beta\, e^{\alpha_j-\beta p_j}\, q_k^2(\boldsymbol{p})\, q_{k-1}(\boldsymbol{p}) + q_k(\boldsymbol{p})\,\beta\, e^{\alpha_j-\beta p_j}\, q_{k-1}^2(\boldsymbol{p})\Big\} & \text{if } \ell < k \\ 0 & \text{otherwise} \end{cases}$$

$$= \begin{cases} \beta\, \phi_j^k(\boldsymbol{p})\, e^{\alpha_i-\beta p_i}\, q_k(\boldsymbol{p}) - \beta\, \phi_j^k(\boldsymbol{p}) & \text{if } \ell = k,\ j = i \\ \beta\, \phi_j^k(\boldsymbol{p})\, e^{\alpha_i-\beta p_i}\, q_k(\boldsymbol{p}) & \text{if } \ell = k,\ j \ne i \\ \beta\, \phi_i^k(\boldsymbol{p})\, e^{\alpha_j-\beta p_j}\, (q_k(\boldsymbol{p}) + q_{k-1}(\boldsymbol{p})) & \text{if } \ell < k \\ 0 & \text{otherwise,} \end{cases} \tag{3.4}$$

where the first two cases use the fact that $q_{k-1}(\boldsymbol{p})$ does not depend on p_j for $j \in S_k$, whereas the third case uses the fact that both $q_k(\boldsymbol{p})$ and $q_{k-1}(\boldsymbol{p})$ depend on p_j for $j \in S_\ell$ and $\ell < k$.

We have $\Pi(\boldsymbol{p}) = \sum_{k\in\mathcal{M}} \sum_{i\in S_k} p_i\, \phi_i^k(\boldsymbol{p})$. Consider a product $j \in S_\ell$. Note that if $i \in S_k$ and $k < \ell$, then $\phi_i^k(\boldsymbol{p})$ does not depend on p_j. In this case, for $j \in S_\ell$, we have

$$\frac{\partial \Pi(\boldsymbol{p})}{\partial p_j} = \phi_j^\ell(\boldsymbol{p}) + p_j\, \frac{\partial \phi_j^\ell(\boldsymbol{p})}{\partial p_j} + \sum_{i\in S_\ell, i\ne j} p_i\, \frac{\partial \phi_i^\ell(\boldsymbol{p})}{\partial p_j} + \sum_{k=\ell+1}^{m} \sum_{i\in S_k} p_i\, \frac{\partial \phi_i^k(\boldsymbol{p})}{\partial p_j}.$$

For an optimal solution $\boldsymbol{p}^*$ to the PRICING problem, the solution satisfies the first order condition $\frac{\partial\Pi(\boldsymbol{p})}{\partial p_j}\Big|_{\boldsymbol{p}=\boldsymbol{p}^*} = 0$ for all $j \in S_\ell$ and $\ell \in \mathcal{M}$. Using (3.4), we can evaluate each one of the partial derivatives on the right side of the expression above. In this case, the partial derivative $\frac{\partial\Pi(\boldsymbol{p})}{\partial p_j}$ in the first order condition is given by expression

$$\frac{\partial \Pi(\boldsymbol{p})}{\partial p_j} = \phi_j^\ell(\boldsymbol{p}) \;+\; p_j \left\{ \beta\, \phi_j^\ell(\boldsymbol{p})\, e^{\alpha_j - \beta p_j}\, q_\ell(\boldsymbol{p}) - \beta\, \phi_j^\ell(\boldsymbol{p}) \right\} \;+ \sum_{i \in S_\ell, i \neq j} p_i\, \beta\, \phi_j^\ell(\boldsymbol{p})\, e^{\alpha_i - \beta p_i}\, q_\ell(\boldsymbol{p})$$

$$+ \sum_{k=\ell+1}^{m} \sum_{i \in S_k} p_i\, \beta\, \phi_i^k(\boldsymbol{p})\, e^{\alpha_j - \beta p_j}\, (q_k(\boldsymbol{p}) + q_{k-1}(\boldsymbol{p}))$$

$$= \beta\, \phi_j^\ell(\boldsymbol{p}) \left\{ \frac{1}{\beta} - p_j + \sum_{i \in S_\ell} p_i\, e^{\alpha_i - \beta p_i}\, q_\ell(\boldsymbol{p}) \right\}$$

$$+ \beta\, e^{\alpha_j - \beta p_j} \sum_{k=\ell+1}^{m} \sum_{i \in S_k} p_i\, \phi_i^k(\boldsymbol{p})\, (q_k(\boldsymbol{p}) + q_{k-1}(\boldsymbol{p})).$$

We have $\phi_j^\ell(\boldsymbol{p}) = \lambda_\ell\, e^{\alpha_j - \beta p_j}\, q_\ell(\boldsymbol{p})\, q_{\ell-1}(\boldsymbol{p})$. In this case, setting the expression on the right side above to zero and solving for p_j, for $j \in S_\ell$, it follows that $\boldsymbol{p}^*$ satisfies the first order condition

$$p_j^* = \frac{1}{\beta} + \sum_{i \in S_\ell} p_i^*\, e^{\alpha_i - \beta p_i^*}\, q_\ell(\boldsymbol{p}^*) + \frac{1}{\lambda_\ell\, q_\ell(\boldsymbol{p}^*)\, q_{\ell-1}(\boldsymbol{p}^*)} \sum_{k=\ell+1}^{m} \sum_{i \in S_k} p_i^*\, \phi_i^k(\boldsymbol{p}^*)\, (q_k(\boldsymbol{p}^*) + q_{k-1}(\boldsymbol{p}^*)). \tag{3.5}$$

The expression on the right side above depends on the stage $\ell \in \mathcal{M}$, but not on the product $j \in S_\ell$. Thus, it follows that if we have $j, j' \in S_\ell$, then we have $p_j^* = p_{j'}^*$. ■

So, we can focus on the solutions where the products in a particular stage have the same price. If we include a marginal cost, then by the theorem above, the optimal prices of the products in a given stage would exceed their marginal costs by a constant, resulting in a constant markup. By the theorem above, letting ρ_k be the price that we charge for the products in stage k, we can use stage-specific prices $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_m)$, instead of product-specific prices $\boldsymbol{p} = (p_1, \ldots, p_n)$, as the decision variables. It is simple to give counterexamples to demonstrate that the expected revenue is not a concave function of either of $\boldsymbol{p}$ or $\boldsymbol{\rho}$. Next, we proceed to giving an equivalent formulation for the PRICING problem, which has a concave objective function and linear constraints. In our equivalent formulation, we use the no-purchase probabilities over different numbers of stages as the decision variables. We can express the expected revenue as a concave function of the no-purchase probabilities over different numbers of stages. Also, we can recover an optimal solution to the PRICING problem by using an optimal solution to the equivalent formulation. These results allow us to use standard convex optimization tools to efficiently solve the PRICING problem.

Convex Reformulation of the Pricing Problem:

Using the stage-specific prices $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_m)$, the total preference weight of the products in stage k is $\widehat{V}_k(\boldsymbol{\rho}) = \sum_{i \in S_k} e^{\alpha_i - \beta \rho_k} = e^{-\beta \rho_k} \sum_{i \in S_k} e^{\alpha_i}$. Noting the definition of the purchase probability for product i, if charge the stage-specific prices $\boldsymbol{\rho}$, then the probability that a customer purchases a product in stage k is $\frac{\lambda_k \sum_{i \in S_k} e^{\alpha_i - \beta \rho_k}}{(1+\sum_{\ell=1}^{k-1} \widehat{V}_\ell(\boldsymbol{p}))(1+\sum_{\ell=1}^{k} \widehat{V}_\ell(\boldsymbol{p}))} = \frac{\lambda_k \widehat{V}_k(\boldsymbol{\rho})}{(1+\sum_{\ell=1}^{k-1} \widehat{V}_\ell(\boldsymbol{p}))(1+\sum_{\ell=1}^{k} \widehat{V}_\ell(\boldsymbol{p}))}$. We refer to this quantity as the stage-specific purchase probability for stage k with the understanding that this quantity captures the total purchase probability in stage k. The idea behind our convex reformulation is to express the stage-specific purchase probabilities and stage-specific prices as functions of the no-purchase probabilities over different numbers of stages. In particular, as a function of the stage-specific prices, we write the no-purchase probability over the first k stages as $\widehat{q}_k(\boldsymbol{\rho}) = 1/(1 + \sum_{\ell=1}^{k} \widehat{V}_\ell(\boldsymbol{\rho}))$. We have $\widehat{q}_{k-1}(\boldsymbol{\rho}) - \widehat{q}_k(\boldsymbol{\rho}) = \frac{\widehat{V}_k(\boldsymbol{\rho})}{(1+\sum_{\ell=1}^{k-1} \widehat{V}_\ell(\boldsymbol{\rho}))(1+\sum_{\ell=1}^{k} \widehat{V}_\ell(\boldsymbol{\rho}))}$. Therefore, given the no-purchase probabilities $\boldsymbol{q} = (q_1, \ldots, q_m)$ over different numbers of stages, the corresponding stage-specific purchase probability for nest k is $\lambda_k\,(q_{k-1} - q_k)$. Also, since $\widehat{q}_k(\boldsymbol{\rho}) = 1/(1 + \sum_{\ell=1}^{k} \widehat{V}_\ell(\boldsymbol{\rho}))$, we get $\frac{1}{\widehat{q}_k(\boldsymbol{\rho})} - \frac{1}{\widehat{q}_{k-1}(\boldsymbol{\rho})} = \widehat{V}_k(\boldsymbol{\rho}) = e^{-\beta \rho_k} \sum_{i \in S_k} e^{\alpha_i}$. Thus, solving for ρ_k in the equality $\frac{1}{\widehat{q}_k(\boldsymbol{\rho})} - \frac{1}{\widehat{q}_{k-1}(\boldsymbol{\rho})} = e^{-\beta \rho_k} \sum_{i \in S_k} e^{\alpha_i}$, given the no-purchase probabilities $\boldsymbol{q} = (q_1, \ldots, q_m)$ over different numbers of stages, the corresponding stage-specific price for stage k is

$$\rho_k(\boldsymbol{q}) = \frac{1}{\beta}\left\{ \log\left(\sum_{i \in S_k} e^{\alpha_i} \right) - \log\left(\frac{1}{q_k} - \frac{1}{q_{k-1}} \right) \right\}.$$

Thus, for given no-purchase probabilities $\boldsymbol{q} = (q_1, \ldots, q_m)$, the customer makes a purchase in stage k with probability $\lambda_k\,(q_{k-1} - q_k)$. If she does so, then the price of the purchased product is $\rho_k(\boldsymbol{q})$.

By the discussion in the paragraph above, we can express the expected revenue as a function of the no-purchase probabilities. In particular, we have

$$\widehat{\Pi}(\boldsymbol{q}) \;=\; \sum_{k \in \mathcal{M}} \lambda_k\,(q_{k-1} - q_k)\,\rho_k(\boldsymbol{q}) \;=\; \sum_{k \in \mathcal{M}} \frac{\lambda_k}{\beta}\,(q_{k-1} - q_k)\left\{ \log\left(\sum_{i \in S_k} e^{\alpha_i} \right) - \log\left(\frac{1}{q_k} - \frac{1}{q_{k-1}} \right) \right\}. \tag{3.6}$$

In the next theorem, we show that the expected revenue function above is concave in the no-purchase probabilities and we can recover the optimal prices by using a maximizer of $\widehat{\Pi}(\boldsymbol{q})$.

Theorem 3.4.2 (Reformulation of Pricing Problem) *The expected revenue* $\widehat{\Pi}(\boldsymbol{q})$ *in* (3.6) *is a concave function of* $\boldsymbol{q}$. *Furthermore, letting* $\boldsymbol{q}^*$ *be an optimal solution to the problem*

$$\max_{\boldsymbol{q} \in \mathbb{R}^m} \left\{ \widehat{\Pi}(\boldsymbol{q}) \;:\; q_{k-1} \ge q_k \quad \forall\, k \in \mathcal{M} \right\} \tag{3.7}$$

with the convention that $q_0 = 1$, *if we set* $\rho_k^* = \rho_k(\boldsymbol{q}^*)$ *for all* $k \in \mathcal{M}$, *then* $\boldsymbol{\rho}^*$ *are optimal stage-specific prices to charge in the* PRICING *problem.*

Proof: To show that $\widehat{\Pi}(\boldsymbol{q})$ is a concave function of $\boldsymbol{q}$, noting that $\sum_{k \in \mathcal{M}} \lambda_k\,(q_{k-1} - q_k)\,\log \sum_{i \in S_k} e^{\alpha_i}$ is linear in $\boldsymbol{q}$, it is enough to argue that $\sum_{k \in \mathcal{M}} \lambda_k\,(q_{k-1} - q_k)\,\log(\frac{1}{q_k} - \frac{1}{q_{k-1}})$ is convex in $\boldsymbol{q}$. We have $\sum_{k \in \mathcal{M}} \lambda_k\,(q_{k-1} - q_k)\,\log(\frac{1}{q_k} - \frac{1}{q_{k-1}}) = \sum_{k \in \mathcal{M}} \lambda_k\,(q_{k-1} -$

$q_k) \log \frac{q_{k-1}-q_k}{q_{k-1}} - \sum_{k\in\mathcal{M}} \lambda_k \, (q_{k-1} - q_k) \log q_k$. First, we show that $\sum_{k\in\mathcal{M}} \lambda_k \, (q_{k-1} - q_k) \log \frac{q_{k-1}-q_k}{q_{k-1}}$ is convex in $\boldsymbol{q}$. The relative entropy function $x \log(x/y)$ is convex in $(x, y) \in \mathbb{R}^2_+$. Since composition of a convex function with an affine function preserves its convexity, $(q_{k-1} - q_k) \log \frac{q_{k-1}-q_k}{q_{k-1}}$ is convex in $\boldsymbol{q}$. Thus, $\sum_{k\in\mathcal{M}} \lambda_k \, (q_{k-1} - q_k) \log \frac{q_{k-1}-q_k}{q_{k-1}}$ is convex in $\boldsymbol{q}$. Second, we show that $-\sum_{k\in\mathcal{M}} \lambda_k \, (q_{k-1} - q_k) \log q_k$ is convex in $\boldsymbol{q}$. Noting that $q_0 = 1$ and rearranging the terms in the sum, we have

$$
\begin{aligned}
&- \sum_{k\in\mathcal{M}} \lambda_k \, (q_{k-1} - q_k) \, \log q_k \\
&= - \lambda_1 \, \log q_1 + \sum_{k=1}^{m-1} q_k \, (\lambda_k \, \log q_k - \lambda_{k+1} \, \log q_{k+1}) + \lambda_m \, q_m \, \log q_m \\
&= - \lambda_1 \, \log q_1 + \sum_{k=1}^{m-1} \lambda_{k+1} \, q_k (\log q_k - \log q_{k+1}) + \sum_{k=1}^{m-1} (\lambda_k - \lambda_{k+1}) \, q_k \, \log q_k + \lambda_m \, q_m \, \log q_m.
\end{aligned}
$$

So, since $x \log(x/y)$ and $x \log x$ are convex in $(x, y) \in \mathbb{R}^2_+$ and $\lambda_k \geq \lambda_{k+1}$, $-\sum_{k\in\mathcal{M}} \lambda_k \, (q_{k-1} - q_k) \log q_k$ is convex in $\boldsymbol{q}$. The second part of the theorem holds by the discussion before the theorem. ■

Problem (3.7) has a concave objective function and linear constraints. Therefore, we can solve this problem efficiently by using standard convex optimization tools.

Monotonicity of Optimal Prices:

In Section 3.3, we show that there exists a revenue-ordered optimal solution to the ASSORTMENT problem, where the revenues of the products in stage k are larger than those of the products in stage $k + 1$. An analogue of this results holds for the PRICING problem as long as all customers have the same patience level. In the next theorem, we compare the optimal prices in different stages. If all customers have the same patience level, then this result implies that the optimal prices for the products in stage k are larger than those for the products in stage $k + 1$.

Theorem 3.4.3 (Monotone Prices) *There exist optimal stage-specific prices* $\boldsymbol{\rho}^* = (\rho_1^*, \ldots, \rho_m^*)$ *in the* PRICING *problem such that* $\lambda_k\, \rho_k^* \geq \lambda_{k+1}\, \rho_{k+1}^*$ *for all* $k = 1, \ldots, m-1$.

Proof: Under stage-specific prices $\boldsymbol{\rho}^*$, the no-purchase probability over the first k stages is $\widehat{q}_k(\boldsymbol{\rho}^*) = \frac{1}{1+\sum_{\ell=1}^{k} \widehat{V}_\ell(\boldsymbol{\rho}^*)}$. The probability that a customer makes a purchase in stage k is

$$\begin{aligned}&\frac{\lambda_k\, \widehat{V}_k(\boldsymbol{\rho}^*)}{(1+\sum_{\ell=1}^{k-1} \widehat{V}_\ell(\boldsymbol{\rho}^*))\,(1+\sum_{\ell=1}^{k} \widehat{V}_\ell(\boldsymbol{\rho}^*))}\\ =&\frac{\lambda_k}{1+\sum_{\ell=1}^{k-1} \widehat{V}_\ell(\boldsymbol{\rho}^*)} - \frac{\lambda_k}{1+\sum_{\ell=1}^{k} \widehat{V}_\ell(\boldsymbol{\rho}^*)} = \lambda_k\, (\widehat{q}_{k-1}(\boldsymbol{\rho}^*) - \widehat{q}_k(\boldsymbol{\rho}^*)).\end{aligned}$$

Also, the total preference weight of the products in stage k is $\widehat{V}_k(\boldsymbol{\rho}^*) = \sum_{i\in S_k} e^{\alpha_i - \beta\rho_k^*}$, which we equivalently write as $\frac{1}{\widehat{q}_k(\boldsymbol{\rho}^*)} - \frac{1}{\widehat{q}_{k-1}(\boldsymbol{\rho}^*)}$.

Letting $q_k^* = \widehat{q}_k(\boldsymbol{\rho}^*)$ for notational brevity, since $\sum_{i\in S_k} \phi_i^k(\boldsymbol{p})$ is the probability that a customer makes a purchase in stage k under prices $\boldsymbol{p}$, for product $j \in S_\ell$, we write (3.5) as

$$\rho_\ell^* = \frac{1}{\beta} + \rho_\ell^* \Big(\frac{1}{q_\ell^*} - \frac{1}{q_{\ell-1}^*}\Big) q_\ell^* + \frac{1}{\lambda_\ell\, q_\ell^*\, q_{\ell-1}^*} \sum_{k=\ell+1}^{m} \rho_k^*\, \lambda_k\, (q_{k-1}^* - q_k^*)\, (q_{k-1}^* + q_k^*),$$

which is the first order condition that needs to be satisfied by the stage-specific prices. We can derive the first order condition above by expressing the expected revenue as a function of the stage-specific prices and differentiating, but using (3.5) provides a short-cut. Arranging the terms and letting $Q_{\ell+1}^* = \sum_{k=\ell+1}^{m} \rho_k^*\, \lambda_k\, (q_{k-1}^* - q_k^*)\, (q_{k-1}^* + q_k^*)$, the first order condition above is

$$\frac{q_\ell^*}{q_{\ell-1}^*}\, \rho_\ell^* = \frac{1}{\beta} + \frac{1}{\lambda_\ell\, q_\ell^*\, q_{\ell-1}^*}\, Q_{\ell+1}^*. \tag{3.8}$$

For stage $\ell+1$, this first order condition is $\frac{q_{\ell+1}^*}{q_\ell^*}\, \rho_{\ell+1}^* = \frac{1}{\beta} + \frac{1}{\lambda_{\ell+1}\, q_{\ell+1}^*\, q_\ell^*}\, Q_{\ell+2}^*$, which, again rearranging the terms, yields $\frac{1}{\lambda_\ell\, q_\ell^*\, q_{\ell-1}^*}\, Q_{\ell+2}^* = \frac{\lambda_{\ell+1}\, (q_{\ell+1}^*)^2}{\lambda_\ell\, q_\ell^*\, q_{\ell-1}^*}\, \rho_{\ell+1}^* - \frac{\lambda_{\ell+1}\, q_{\ell+1}^*}{\beta\, \lambda_\ell\, q_{\ell-1}^*}$.

Noting the definition of $Q^*_{\ell+1}$, we have $Q^*_{\ell+1} = \rho^*_{\ell+1}\,\lambda_{\ell+1}\,((q^*_\ell)^2 - (q^*_{\ell+1})^2) + Q^*_{\ell+2}$. Therefore, using this expression for $Q^*_{\ell+1}$ in (3.8), we obtain

$$\begin{aligned}
\frac{q^*_\ell}{q^*_{\ell-1}}\,\rho^*_\ell &= \frac{1}{\beta} + \frac{1}{\lambda_\ell\,q^*_\ell\,q^*_{\ell-1}}\Big\{\rho^*_{\ell+1}\,\lambda_{\ell+1}\,((q^*_\ell)^2 - (q^*_{\ell+1})^2) + Q^*_{\ell+2}\Big\}\\
&\overset{(a)}{=} \frac{1}{\beta} + \frac{1}{\lambda_\ell\,q^*_\ell\,q^*_{\ell-1}}\,\rho^*_{\ell+1}\,\lambda_{\ell+1}\,((q^*_\ell)^2 - (q^*_{\ell+1})^2) + \frac{\lambda_{\ell+1}\,(q^*_{\ell+1})^2}{\lambda_\ell\,q^*_\ell\,q^*_{\ell-1}}\,\rho^*_{\ell+1} - \frac{\lambda_{\ell+1}\,q^*_{\ell+1}}{\beta\,\lambda_\ell\,q^*_{\ell-1}}\\
&= \left(1 - \frac{\lambda_{\ell+1}\,q^*_{\ell+1}}{\lambda_\ell\,q^*_{\ell-1}}\right)\frac{1}{\beta} + \frac{\lambda_{\ell+1}\,q^*_\ell}{\lambda_\ell\,q^*_{\ell-1}}\,\rho^*_{\ell+1},
\end{aligned}$$

where (a) holds because $\frac{1}{\lambda_\ell\,q^*_\ell\,q^*_{\ell-1}}\,Q^*_{\ell+2} = \frac{\lambda_{\ell+1}\,(q^*_{\ell+1})^2}{\lambda_\ell\,q^*_\ell\,q^*_{\ell-1}}\,\rho^*_{\ell+1} - \frac{\lambda_{\ell+1}\,q^*_{\ell+1}}{\beta\,\lambda_\ell\,q^*_{\ell-1}}$. The chain of equalities yields $\frac{q^*_\ell}{\lambda_\ell\,q^*_{\ell-1}}(\lambda_\ell\,\rho^*_\ell - \lambda_{\ell+1}\,\rho^*_{\ell+1}) = (1 - \frac{\lambda_{\ell+1}\,q^*_{\ell+1}}{\lambda_\ell\,q^*_{\ell-1}})\frac{1}{\beta}$. Since $q^*_{\ell+1} \leq q^*_{\ell-1}$ and $\lambda_{\ell+1} \leq \lambda_\ell$, we get $\lambda_\ell\,\rho^*_\ell \geq \lambda_{\ell+1}\,\rho^*_{\ell+1}$. ■

By the theorem above, if $\lambda_1 = \ldots = \lambda_m = 1$, then we have $\rho^*_k \geq \rho^*_{k+1}$.

3.4.2 Optimal Assortments and Prices

In this section, we study the case where both the assortments $(S_1, \ldots, S_m)$ offered over m stages and the prices charged for the products are decision variables. By the discussion in the previous section, for any fixed sequence of assortments, it is optimal to charge stage-specific prices. Therefore, it is enough to focus on stage-specific prices when both the sequence of assortments to offer and the prices to charge are decision variables, but to simplify the proofs of our results, we go back to using product-specific prices. As a function of the product-specific prices $\boldsymbol{p} = (p_1, \ldots, p_n)$, we let $V(\boldsymbol{p}, S) = \sum_{i \in S} e^{\alpha_i - \beta\,p_i}$ to capture the total preference weight of the products in S. Noting the discussion in Section 3.2, if we charge the prices $\boldsymbol{p}$ and offer the assortments $(S_1, \ldots, S_m)$, then a customer purchases product $i \in S_k$ with probability $\lambda_k\,e^{\alpha_i - \beta\,p_i} / ((1 + \sum_{\ell=1}^{k-1} V(\boldsymbol{p}, S_\ell))\,(1 + \sum_{\ell=1}^{k} V(\boldsymbol{p}, S_\ell)))$. So, as a function of the product-specific prices $\boldsymbol{p}$ and the assortments $(S_1, \ldots, S_m)$ over m stages, the expected revenue is

$$\Pi(\boldsymbol{p}, S_1, \ldots, S_m) = \sum_{k \in \mathcal{M}} \frac{\lambda_k \sum_{i \in S_k} p_i \, e^{\alpha_i - \beta p_i}}{(1 + \sum_{\ell=1}^{k-1} V(\boldsymbol{p}, S_\ell))\,(1 + \sum_{\ell=1}^{k} V(\boldsymbol{p}, S_\ell))}.$$

We continue using $\mathcal{F}$ to denote the set of feasible assortments that we can offer over m stages, ensuring that the assortments offered over different stages are disjoint. Our goal is to find the assortment to offer in each stage and the prices to charge for the products to maximize the expected revenue, yielding the problem

$$\max_{(\boldsymbol{p}, S_1, \ldots, S_m) \in \mathbb{R}^n \times \mathcal{F}} \Pi(\boldsymbol{p}, S_1, \ldots, S_m). \qquad \text{(PRICING-ASSORTMENT)}$$

The problem above involves both continuous and discrete decision variables. For this problem, we give a greedy search heuristic that is guaranteed to obtain at least 50% of the optimal expected revenue. In our computational experiments, we compare the performance of this heuristic with an efficiently-computable upper bound on the optimal expected revenue and demonstrate that the practical performance of the heuristic is within 1.5% of the optimal expected revenue, on average. The idea behind our heuristic is to start with a sequence of assortments to offer over all m stages. Given the current sequence of assortments on hand, we check all neighbors of the current sequence of assortments for an appropriately defined neighborhood. For each sequence of assortments in the neighborhood of the current sequence, we compute the corresponding optimal prices to charge. By the discussion in the previous section, we can solve a convex program to compute the optimal prices for a given sequence of assortments. Among all sequences of assortments that we check in the neighborhood and the corresponding optimal prices, we choose the one that provides the largest expected revenue. We repeat the process starting from the best sequence of assortments we choose. In our greedy search heuristic, we define the neighborhood of the sequence of assortments $(S_1, \ldots, S_m)$ as all sequences of assortments that are obtained by moving one product from one stage to another. In particular, we define

$$C(S_1, \ldots, S_m) = \Big\{(S_1, \ldots, S_k \setminus \{i\}, \ldots, S_\ell \cup \{i\}, \ldots, S_m) \in \mathcal{F} \\ : \ \forall\, i \in S_k,\ k, \ell \in \mathcal{M},\ k \neq \ell\Big\}.$$

Below is our greedy search heuristic. Note that $\boldsymbol{p}^*(\widehat{S}_1, \ldots, \widehat{S}_m)$ in Step 2 corresponds to the optimal prices for the sequence of assortments $(\widehat{S}_1, \ldots, \widehat{S}_m)$.

<u>Greedy Search Heuristic</u>:

- **Step 1.** Initialize the iteration counter by setting $t = 1$. Set $(S_1^t, S_2^t, \ldots, S_m^t) = (\mathcal{N}, \varnothing, \ldots, \varnothing)$ and $\text{Rev}^t = \max_{\boldsymbol{p} \in \mathbb{R}^n} \Pi(\boldsymbol{p}, S_1^t, \ldots, S_m^t)$.
- **Step 2.** For all $(\widehat{S}_1, \ldots, \widehat{S}_m) \in C(S_1^1, \ldots, S_m^t)$, set $\boldsymbol{p}^*(\widehat{S}_1, \ldots, \widehat{S}_m) = \arg\max_{\boldsymbol{p} \in \mathbb{R}^n} \Pi(\boldsymbol{p}, \widehat{S}_1, \ldots, \widehat{S}_m)$.
- **Step 3**Set $(S_1^{t+1}, \ldots, S_m^{t+1}) = \arg\max_{(\widehat{S}_1, \ldots, \widehat{S}_m) \in C(S_1^t, \ldots, S_m^t)} \Pi(\boldsymbol{p}^*(\widehat{S}_1, \ldots, \widehat{S}_m), \widehat{S}_1, \ldots, \widehat{S}_m)$ and $\text{Rev}^{t+1} = \Pi(\boldsymbol{p}^*(S_1^{t+1}, \ldots, S_m^{t+1}), S_1^{t+1}, \ldots, S_m^{t+1})$.
- **Step 4.** If $\text{Rev}^{t+1} > \text{Rev}^t$, then increase t by one and go to Step 2. Return $(S_1^{t+1}, \ldots, S_m^{t+1})$.

In Step 2, for each sequence of assortments in the neighborhood of the current solution, we find the optimal prices to charge. In Step 3, we check all sequences of assortments in the neighborhood of the current solution and pick the one that provides the largest expected revenue when we charge the optimal prices for this sequence of assortments. The sequence of assortments that we pick is $(S_1^{t+1}, \ldots, S_m^{t+1})$. The greedy search heuristic is guaranteed to stop since the expected revenue at each iteration strictly increases and there are finitely many possible sequences of assortments. In the next theorem, we give a performance guarantee for the greedy heuristic.

Theorem 3.4.4 (Performance for Greedy Search) *Letting π^* be the optimal objective value of the* PRICING-ASSORTMENT *problem, $(\widetilde{S}_1, \ldots, \widetilde{S}_m)$ be the output of the greedy heuristic and $\widetilde{\boldsymbol{p}} = \arg\max_{\boldsymbol{p} \in \mathbb{R}^m} \Pi(\boldsymbol{p}, \widetilde{S}_1, \ldots, \widetilde{S}_m)$, we have $\Pi(\widetilde{\boldsymbol{p}}, \widetilde{S}_1, \ldots, \widetilde{S}_m) \geq \frac{1}{2}\pi^*$.*

Proof: Fix prices $\boldsymbol{p}$ and sequence of assortments $(S_1, \ldots, S_m)$. For notational brevity, letting $T_k = S_1 \cup \ldots \cup S_k$ with $T_0 = \emptyset$, by the definition of the expected revenue function, we have

$$
\begin{aligned}
\Pi(\boldsymbol{p}, S_1, \ldots, S_m) &= \sum_{k \in \mathcal{M}} \frac{\lambda_k \, \sum_{i \in S_k} p_i \, e^{\alpha_i - \beta\, p_i}}{(1 + \sum_{\ell=1}^{k-1} V(\boldsymbol{p}, S_\ell))\,(1 + \sum_{\ell=1}^{k} V(\boldsymbol{p}, S_\ell))} \\
&= \sum_{k \in \mathcal{M}} \frac{\lambda_k \sum_{i \in T_k} p_i \, e^{\alpha_i - \beta\, p_i} - \lambda_k \sum_{i \in T_{k-1}} p_i \, e^{\alpha_i - \beta\, p_i}}{(1 + V(\boldsymbol{p}, T_{k-1}))\,(1 + V(\boldsymbol{p}, T_k))} \\
&\stackrel{(a)}{=} \sum_{k=1}^{m-1} \frac{\sum_{i \in T_k} p_i \, e^{\alpha_i - \beta\, p_i}}{1 + V(\boldsymbol{p}, T_k)} \left\{ \frac{\lambda_k}{1 + V(\boldsymbol{p}, T_{k-1})} - \frac{\lambda_{k+1}}{1 + V(\boldsymbol{p}, T_{k+1})} \right\} \\
&\qquad\qquad + \frac{\lambda_m \sum_{i \in T_m} p_i \, e^{\alpha_i - \beta\, p_i}}{(1 + V(\boldsymbol{p}, T_{m-1}))\,(1 + V(\boldsymbol{p}, T_m))} \\
&\stackrel{(b)}{\leq} \max_{\boldsymbol{p} \in \mathbb{R}^n} \max_{S \subseteq \mathcal{N}} \left\{ \frac{\sum_{i \in S} p_i \, e^{\alpha_i - \beta\, p_i}}{1 + V(\boldsymbol{p}, S)} \right\} \left(\sum_{k=1}^{m-1} \left\{ \frac{\lambda_k}{1 + V(\boldsymbol{p}, T_{k-1})} - \frac{\lambda_{k+1}}{1 + V(\boldsymbol{p}, T_{k+1})} \right\} \right. \\
&\qquad\qquad \left. + \frac{\lambda_m}{1 + V(\boldsymbol{p}, T_{m-1})} \right) \\
&\stackrel{(c)}{=} \max_{\boldsymbol{p} \in \mathbb{R}^n} \max_{S \subseteq \mathcal{N}} \left\{ \frac{\sum_{i \in S} p_i \, e^{\alpha_i - \beta\, p_i}}{1 + V(\boldsymbol{p}, S)} \right\} \left(\lambda_1 + \sum_{k=2}^{m} \lambda_k \left\{ \frac{1}{1 + V(\boldsymbol{p}, T_{k-1})} - \frac{1}{1 + V(\boldsymbol{p}, T_k)} \right\} \right) \\
&\stackrel{(d)}{\leq} \max_{\boldsymbol{p} \in \mathbb{R}^n} \max_{S \subseteq \mathcal{N}} \left\{ \frac{\sum_{i \in S} p_i \, e^{\alpha_i - \beta\, p_i}}{1 + V(\boldsymbol{p}, S)} \right\} \left(1 + \sum_{k=2}^{m} \left\{ \frac{1}{1 + V(\boldsymbol{p}, T_{k-1})} - \frac{1}{1 + V(\boldsymbol{p}, T_k)} \right\} \right) \\
&= \max_{\boldsymbol{p} \in \mathbb{R}^n} \max_{S \subseteq \mathcal{N}} \left\{ \frac{\sum_{i \in S} p_i \, e^{\alpha_i - \beta\, p_i}}{1 + V(\boldsymbol{p}, S)} \right\} \left(1 + \frac{1}{1 + V(\boldsymbol{p}, T_1)} - \frac{1}{1 + V(\boldsymbol{p}, T_m)} \right) \\
&\leq 2 \max_{\boldsymbol{p} \in \mathbb{R}^n} \max_{S \subseteq \mathcal{N}} \left\{ \frac{\sum_{i \in S} p_i \, e^{\alpha_i - \beta\, p_i}}{1 + V(\boldsymbol{p}, S)} \right\} \stackrel{(e)}{=} 2 \max_{\boldsymbol{p} \in \mathbb{R}^n} \max_{S \subseteq \mathcal{N}} \; \Pi(\boldsymbol{p}, S, \emptyset, \ldots, \emptyset),
\end{aligned}
$$

where (a) and (c) follow by arranging the terms, (b) holds since $T_k \subseteq \mathcal{N}$, along with $\lambda_k \geq \lambda_{k+1}$ and $V(\boldsymbol{p}, T_{k-1}) \leq V(\boldsymbol{p}, T_k)$, (d) holds since $\lambda_k \leq 1$ and (e) holds by the expected revenue function.

The last expression above is the optimal expected revenue when we offer products in the first stage, but not in the later stages. We continue the chain of inequalities above as

$$2 \max_{\boldsymbol{p} \in \mathbb{R}^n} \max_{S \subseteq \mathcal{N}} \Pi(\boldsymbol{p}, S, \emptyset, \ldots, \emptyset) \overset{(a)}{=} 2 \max_{\boldsymbol{p} \in \mathbb{R}^n} \Pi(\boldsymbol{p}, \mathcal{N}, \emptyset, \ldots, \emptyset)$$
$$\overset{(b)}{=} 2\,\mathrm{REV}^1 \overset{(c)}{\leq} 2\,\Pi(\widetilde{\boldsymbol{p}}, \widetilde{S}_1, \ldots, \widetilde{S}_m),$$

where (a) uses the fact that if the customers choose according to the standard multinomial logit model with a single stage, then it is optimal to offer all products in the joint pricing and assortment optimization problem, (b) holds since we have $(S_1^1, \ldots, S_m^1) = (\mathcal{N}, \emptyset, \ldots, \emptyset)$ at the first iteration of the greedy search heuristic and (c) follows as the expected revenue at subsequent iterations of the greedy search heuristic can only increase. The two chains of inequalities above hold for all prices $\boldsymbol{p}$ and assortments $(S_1, \ldots, S_m)$, in which case, combining the two chains of inequalities, it follows that $\pi^* = \max_{(\boldsymbol{p}, S_1, \ldots, S_m) \in \mathbb{R}^n \times \mathcal{F}} \Pi(\boldsymbol{p}, S_1, \ldots, S_m) \leq 2\,\Pi(\widetilde{\boldsymbol{p}}, \widetilde{S}_1, \ldots, \widetilde{S}_m)$. ■

3.5 Assortment Optimization under a Space Constraint

We consider the assortment problem when each product occupies a certain amount of space and there is a limit on the total space consumption of the products offered in all stages. As in Section 3.3, the revenue of product i is r_i and we index the products such that $r_1 \geq r_2 \geq \ldots \geq r_n$. The space consumption of product i is c_i. We let $C(S) = \sum_{i \in S} c_i$. The total amount of space available is b.

Noting the expected revenue function in the ASSORTMENT problem, we want to solve

$$\max_{(S_1,\ldots,S_m)\in\mathcal{F}} \left\{ \sum_{k\in\mathcal{M}} \frac{\lambda_k\, W(S_k)}{(1+\sum_{q=1}^{k-1} V(S_q))\,(1+\sum_{q=1}^{k} V(S_q))} \;:\; \sum_{k\in\mathcal{M}} C(S_k) \le b \right\}. \tag{CAPACITATED}$$

The CAPACITATED problem is NP-hard even with a single stage [36], so we focus on developing an FPTAS. In Appendix B.2, we build on Lemma 3.3.2 to show that there exists an optimal solution $(S_1^*,\ldots,S_m^*)$ such that $S_k^* \subseteq \{j_k^*+1,\ldots,j_{k+1}^*\}$ for $j_1^*,\ldots,j_m^*$ that satisfy $0 = j_1^* \le j_2^* \le \ldots \le j_m^* \le j_{m+1}^* = n$. This result does not immediately yield an efficient algorithm since the optimal assortment S_k^* in stage k may omit products in $\{j_k^*+1,\ldots,j_{k+1}^*\}$.

To give an FPTAS for the CAPACITATED problem, we fix $\epsilon \in (0,1)$ and proceed in two parts.

<u>Part 1</u>:

For each $j,\ell \in \{0,\ldots,n\}$ with $j \le \ell$, we will construct a collection of candidate assortments $\textsc{Cand}(j,\ell)$ that satisfies the following two properties.

- **(Correct Product Interval)** For each $\widehat{S} \in \textsc{Cand}(j,\ell)$, we have $\widehat{S} \subseteq \{j+1,\ldots,\ell\}$. Thus, a candidate assortment in $\textsc{Cand}(j,\ell)$ can only include the products in $\{j+1,\ldots,\ell\}$.
- **(Limited Degredation)** For each $S \subseteq \{j+1,\ldots,\ell\}$, there exists $\widehat{S} \in \textsc{Cand}(j,\ell)$ such that $W(\widehat{S}) \ge (1-\epsilon/4)\, W(S)$, $V(\widehat{S}) \le (1+\epsilon/4)\, V(S)$ and $C(\widehat{S}) \le C(S)$.

Intuitively speaking, in the CAPACITATED problem, we prefer $S \subseteq \mathcal{N}$ with larger $W(S)$, smaller $V(S)$ and smaller $C(S)$. By the second property, for any assortment $S \subseteq \{j+1,\ldots,\ell\}$, there exists a candidate assortment $\widehat{S} \in \textsc{Cand}(j,\ell)$ that is almost as preferable. Letting $v_{\min} = \min\{v_i : i \in \mathcal{N}\}$, $v_{\max} = \max\{v_i : i \in \mathcal{N}\}$, $w_{\min} = \min\{v_i r_i : i \in \mathcal{N}\}$ and $w_{\max} = \max\{v_i r_i : i \in \mathcal{N}\}$, we will construct all collections $\{\textsc{Cand}(j,\ell) : j,\ell \in \{0,\ldots,n\},\ j \le \ell\}$ in $O(\frac{n^4}{\epsilon^2} \log(\frac{nw_{\max}}{w_{\min}}) \log(\frac{nv_{\max}}{v_{\min}}))$ operations.

Part 2:

By using the collection of candidate assortments $\textsc{Cand}(j,\ell)$, we solve an approximate version of the CAPACITATED problem. In particular, we consider the the problem

$$\widetilde{z} = \max_{(S_1,\ldots,S_m,j_1,\ldots,j_m)} \left\{ \sum_{k\in\mathcal{M}} \frac{\lambda_k\, W(S_k)}{(1+\sum_{q=1}^{k-1} V(S_q))\,(1+\sum_{q=1}^{k} V(S_q))} \right.$$
$$\left. : S_k \in \textsc{Cand}(j_k, j_{k+1})\ \ \forall\, k \in \mathcal{M},\ j_k \le j_{k+1}\ \ \forall\, k \in \mathcal{M},\ \ \sum_{k\in\mathcal{M}} C(S_k) \le b \right\}, \tag{3.9}$$

where we follow the convention that $j_{m+1} = n$. Comparing the problem above with the CAPACITATED problem, we choose the assortment S_k in $\textsc{Cand}(j_k, j_{k+1})$ above. Also, in the problem above, we do not explicitly impose the constraint that $S_k \cap S_q = \varnothing$ for $k \neq q$, but the constraint $S_k \in \textsc{Cand}(j_k, j_{k+1})$ for all $k \in \mathcal{M}$, along with $j_k \le j_{k+1}$ for all $k \in \mathcal{M}$, ensures that $S_k \cap S_q = \varnothing$ for $k \neq q$. We will obtain a $(1-\frac{\epsilon}{4})$-approximate solution to problem (3.9) in $O(\frac{n^4 m^3}{\epsilon^4} \log(\frac{nw_{\max}}{w_{\min}}) \log(\frac{nw_{\max}\,(1 \vee nv_{\max})^2}{\lambda_m\, w_{\min}}) \log^2(\frac{nv_{\max}}{v_{\min}}))$ operations, where we use $a \vee b = \max\{a,b\}$. In this case, by the two parts, we get the following result.

Theorem 3.5.1 (FPTAS under a Space Constraint) *For each $\epsilon \in (0,1)$, we can obtain a $(1-\epsilon)$-approximate solution to the* CAPACITATED *problem in the number of operations*

$$O\left(\frac{n^4 m^3}{\epsilon^4} \log\left(\frac{nw_{\max}}{w_{\min}}\right) \log\left(\frac{nw_{\max}\,(1 \vee nv_{\max})^2}{\lambda_m\, w_{\min}}\right) \log^2\left(\frac{nv_{\max}}{v_{\min}}\right)\right).$$

Proof. We execute the two parts above. Let $(S_1^*, \ldots, S_m^*)$ be an optimal solution to the CAPACITATED problem. We know that there exist $0 = j_1^* \leq j_2^* \leq \ldots \leq j_m^* \leq j_{m+1}^* = n$ such that $S_k^* \subseteq \{j_k^* + 1, \ldots, j_{k+1}^*\}$ for all $k \in \mathcal{M}$. By the second property in Part 1, for each $k \in \mathcal{M}$, there exists $\widehat{S}_k \in \text{CAND}(j_k^*, j_{k+1}^*)$ be such that $W(\widehat{S}_k) \geq (1 - \epsilon/4)\, W(S_k^*)$, $V(\widehat{S}_k) \leq (1 + \epsilon/4)\, V(S_k^*)$ and $C(\widehat{S}_k) \leq C(S_k^*)$. Since $\sum_{k \in \mathcal{M}} C(S_k^*) \leq b$ and $C(\widehat{S}_k) \leq C(S_k^*)$ for all $k \in \mathcal{M}$, the solution $(\widehat{S}_1, \ldots, \widehat{S}_m, j_1^*, \ldots, j_m^*)$ is feasible to (3.9). Executing Part 2, we have a $(1 - \frac{\epsilon}{4})$-approximate solution to problem (3.9) as well, which we denote by $(\widetilde{S}_1, \ldots, \widetilde{S}_m, \widetilde{j}_1, \ldots, \widetilde{j}_m)$. So, we get

$$
\begin{aligned}
&\sum_{k \in \mathcal{M}} \frac{\lambda_k\, W(\widetilde{S}_k)}{(1 + \sum_{q=1}^{k-1} V(\widetilde{S}_q))\,(1 + \sum_{q=1}^{k} V(\widetilde{S}_q))} \\
&\qquad \stackrel{(a)}{\geq} \left(1 - \frac{\epsilon}{4}\right) \sum_{k \in \mathcal{M}} \frac{\lambda_k\, W(\widehat{S}_k)}{(1 + \sum_{q=1}^{k-1} V(\widehat{S}_q))\,(1 + \sum_{q=1}^{k} V(\widehat{S}_q))} \\
&\qquad \stackrel{(b)}{\geq} \left(1 - \frac{\epsilon}{4}\right) \sum_{k \in \mathcal{M}} \frac{(1 - \frac{\epsilon}{4})\, \lambda_k\, W(S_k^*)}{(1 + \sum_{q=1}^{k-1} (1 + \frac{\epsilon}{4}) V(S_q^*))\,(1 + \sum_{q=1}^{k} (1 + \frac{\epsilon}{4}) V(S_q^*))} \\
&\qquad \geq \frac{(1 - \frac{\epsilon}{4})^2}{(1 + \frac{\epsilon}{4})^2} \sum_{k \in \mathcal{M}} \frac{\lambda_k\, W(S_k^*)}{(1 + \sum_{q=1}^{k-1} V(S_q^*))\,(1 + \sum_{q=1}^{k} V(S_q^*))}
\end{aligned}
$$

In the chain of inequalities above, (a) holds because $(\widetilde{S}_1, \ldots, \widetilde{S}_m, \widetilde{j}_1, \ldots, \widetilde{j}_m)$ is a $(1 - \frac{\epsilon}{4})$-approximate solution to problem (3.9), whereas $(\widehat{S}_1, \ldots, \widehat{S}_m, j_1^*, \ldots, j_m^*)$ is only a feasible solution to problem (3.9), whereas (b) holds because $W(\widehat{S}_k) \geq (1 - \epsilon/4)\, W(S_k^*)$ and $V(\widehat{S}_k) \leq (1 + \epsilon/4)\, V(S_k^*)$. For $\epsilon \in (0, 1)$, we have $\frac{(1-\epsilon/4)^2}{(1+\epsilon/4)^2} \geq (1 - \frac{\epsilon}{4})^4 \geq 1 - \epsilon$. In this case, letting z^* be the optimal objective value of the CAPACITATED problem and noting that $(S_1^*, \ldots, S_m^*)$ is the optimal solution to the CAPACITATED problem, the chain of inequalities above implies that the solution $(\widetilde{S}_1, \ldots, \widetilde{S}_m)$ provides an objective value of at least $(1 - \epsilon)\, z^*$ to the CAPACITATED problem, so this solution is a $(1 - \epsilon)$-approximate solution to the CAPACITATED problem. The number of operations to obtain the solution $(\widetilde{S}_1, \ldots, \widetilde{S}_m)$ is equal to the number of operations to execute the two parts, which, by the discussion right before the theorem, is given by the expression in the theorem. ■

The number of operations in Theorem 3.5.1 is polynomial in the input size and $1/\epsilon$, giving an FPTAS. In the next two sections, we discuss how to execute the two parts.

3.5.1 Constructing Collections of Candidate Assortments

We focus on executing Part 1. For each $j, \ell \in \{0, \ldots, n\}$ with $j \le \ell$, we separately construct the collection of candidate assortments $\text{CAND}(j, \ell)$. Therefore, we fix j, ℓ throughout this section. Intuitively speaking, to construct the collection of candidate assortments $\text{CAND}(j, \ell)$, we use a geometric grid to guess the values of $W(S)$ and $V(S)$ for each possible assortment $S \subseteq \{j+1, \ldots, \ell\}$. For each guess for the values of $W(S)$ and $V(S)$, we use a dynamic program to find an assortment $\widehat{S}$ such that $W(\widehat{S})$ and $V(\widehat{S})$ are not too far from the guess and the capacity consumption of $\widehat{S}$ is as small as possible. The dynamic program that we use is, in spirit, similar to the one that is used for solving the knapsack problem. In particular, for fixed $\rho > 0$, we define the geometric grid $\text{DOM} = \{(1+\rho)^r : r \in \mathbb{Z}\} \cup \{0\}$. We define the round down operator "$\lfloor \cdot \rfloor$" that rounds its argument down to the closest point in DOM when the argument is positive. That is, if $a \ge 0$, then we have $\lfloor a \rfloor = \max\{b \in \text{DOM} : b \le a\}$. If $a < 0$, then we follow the convention that $\lfloor a \rfloor = 0$. Similarly, we define the round up operator "$\lceil \cdot \rceil$" that rounds its argument up to the closest point in DOM when the argument is positive. That is, if $a \ge 0$, then we have $\lceil a \rceil = \min\{b \in \text{DOM} : b \ge a\}$. If $a < 0$, then we follow the convention that $\lceil a \rceil = -\infty$. For given $(x, y) \in \text{DOM}^2$ and (j, ℓ), consider finding the smallest possible capacity consumption of any assortment $S \subseteq \{j+1, \ldots, \ell\}$ that satisfies $W(S) \geq x$ and $V(S) \leq y$. For this purpose, we use the dynamic program

$$\Theta_i^\ell(x, y) \;=\; \min_{u_i \in \{0,1\}} \Big\{ c_i\, u_i + \Theta_{i+1}^\ell(\lfloor x - v_i\, r_i\, u_i \rfloor \,,\, \lceil y - v_i\, u_i \rceil) \Big\}, \tag{3.10}$$

where we use the boundary condition that $\Theta_{\ell+1}^\ell(x, y) = 0$ if $x \leq 0$ and $y \geq 0$. If, on the other hand, $x > 0$ or $y < 0$, then we have $\Theta_{\ell+1}^\ell(x, y) = +\infty$.

In the dynamic program above, the decision epochs correspond to products. The action at decision epoch i corresponds to whether we offer product i. If we drop the round down and up operators on the right side of (3.10), then $\Theta_{j+1}^\ell(x, y)$ gives the smallest possible capacity consumption of any assortment $S \subseteq \{j+1, \ldots, \ell\}$ that satisfies $W(S) \geq x$ and $V(S) \leq y$. If there is no assortment $S \subseteq \{j+1, \ldots, \ell\}$ such that $W(S) \geq x$ and $V(S) \leq y$, then $\Theta_{j+1}^\ell(x, y) = +\infty$. With the round down and up operators on the right side of (3.10), this dynamic program is only an approximation. We consider $(x, y) \in \textsc{Dom}^2$, so the number of possible values of the state variable is countable but not yet finite. Shortly in this section, we put natural upper and lower bounds on the two components of the state variable, in which case, the number of possible values for the state variable becomes finite. Thus, we will solve the dynamic program in (3.10) in finite time.

To construct the collection of candidate assortments $\textsc{Cand}(j, \ell)$, we compute the value functions $\{\Theta_i^\ell(x, y) \,:\, (x, y) \in \textsc{Dom}^2,\; i = 1, \ldots, \ell+1\}$ by using the dynamic program in (3.10). Once we compute the value functions, for each $(x, y) \in \textsc{Dom}^2$, starting with state (x, y) and decision epoch $j+1$, we follow the sequence of optimal state-action pairs in the dynamic program in (3.10). In this way, we obtain an assortment $\widehat{S}_{x,y}$ for each $(x, y) \in \textsc{Dom}^2$, which we use as one of the candidate assortments in the collection $\textsc{Cand}(j, \ell)$. To be specific, for each $(x, y) \in \textsc{Dom}^2$, if $\Theta_{j+1}^\ell(x, y) < +\infty$, then we construct the assortment $\widehat{S}_{x,y}$ by using the following algorithm. Throughout this section, we refer to this algorithm as the candidate construction algorithm.

<u>Candidate Construction</u>:

Step 1. Set $i = j+1$, $\widehat{x}_i = x$ and $\widehat{y}_i = y$.

Step 2. Set

$$\widehat{u}_i = \arg\min_{u_i \in \{0,1\}} \Big\{ c_i\, u_i + \Theta^{\ell}_{i+1}(\lfloor \widehat{x}_i - v_i\, r_i\, u_i \rfloor\,,\, \lceil \widehat{y}_i - v_i\, u_i \rceil) \Big\}.$$

Step 3. Set

$$\widehat{x}_{i+1} = \lfloor \widehat{x}_i - v_i\, r_i\, \widehat{u}_i \rfloor \quad \text{and} \quad \widehat{y}_{i+1} = \lceil \widehat{y}_i - v_i\, \widehat{u}_i \rceil.$$

Increase i by one. If $i < \ell + 1$, then go to Step 2.

Step 4. Return $\widehat{S}_{x,y} = \{i \in \{j+1, \ldots, \ell\} : \widehat{u}_i = 1\}$.

In the next lemma, we show useful properties of the assortment $\widehat{S}_{x,y}$ obtained by the algorithm above. In particular, if there exists an assortment $S \subseteq \{j + 1, \ldots, \ell\}$ with $W(S) \geq x$ and $V(S) \leq y$, then we must have $\Theta^{\ell}_{j+1}(x, y) < +\infty$, in which case, we do execute the candidate construction algorithm to obtain the assortment $\widehat{S}_{x,y}$. Also, considering the assortment $S \subseteq \{j+1, \ldots, \ell\}$ with $W(S) \geq x$ and $V(S) \leq y$, the output of the candidate construction algorithm $\widehat{S}_{x,y}$ satisfies $W(\widehat{S}_{x,y}) \geq \frac{1}{(1+\rho)^n}\, x$, $V(\widehat{S}_{x,y}) \leq (1+\rho)^n\, y$ and $C(\widehat{S}_{x,y}) \leq C(S)$. Recall that we prefer an assortment S with larger $W(S)$, smaller $V(S)$ and smaller $C(S)$. Thus, if ρ is small, then the candidate assortment $\widehat{S}_{x,y}$ is almost as preferable as the assortment S. The proof of the lemma is in Appendix B.3.

Lemma 3.5.2 (Candidate Assortments) *If there exists an assortment* $S \subseteq \{j+1,\ldots,\ell\}$ *such that* $W(S) \geq x$ *and* $V(S) \leq y$*, then we have* $\Theta^{\ell}_{j+1}(x,y) < +\infty$*,* $W(\widehat{S}_{x,y}) \geq \frac{1}{(1+\rho)^n}\,x$*,* $V(\widehat{S}_{x,y}) \leq (1+\rho)^n\,y$ *and* $C(\widehat{S}_{x,y}) \leq C(S)$*.*

Next, we discuss how to use the lemma and the candidate construction algorithm to execute Part 1. Given $\epsilon \in (0,1)$, we set the size of the geometric grid as $\rho = \frac{1}{8\,(n+1)}\,\epsilon$. For all $S \neq \emptyset$, $W(S) \in [w_{\min}, n\,w_{\max}]$ and $V(S) \in [v_{\min}, n\,v_{\max}]$, so we construct the collection $\textsc{Cand}(j,\ell)$ as

$$\textsc{Cand}(j,\ell) = \Big\{\widehat{S}_{x,y} \;:\; (x,y) \in \textsc{Dom}^2,\\ x \in [\lfloor w_{\min} \rfloor, \lceil n\,w_{\max} \rceil] \cup \{0\},\; y \in [\lfloor v_{\min} \rfloor, \lceil n\,v_{\max} \rceil] \cup \{0\}\Big\}. \quad (3.11)$$

Noting that $\widehat{S}_{x,y} \subseteq \{j+1,\ldots,\ell\}$, we have $\widehat{S} \subseteq \{j+1,\ldots,\ell\}$ for all $\widehat{S} \in \textsc{Cand}(j,\ell)$. Thus, the collection of candidate assortments $\textsc{Cand}(j,\ell)$ given above satisfies the correct product interval property in Part 1. It remains to argue that the collection of candidate assortments $\textsc{Cand}(j,\ell)$ given above satisfies the limited degradation property in Part 1 as well. Given an assortment $S \subseteq \{j+1,\ldots,\ell\}$, let $(x,y) \in \textsc{Dom}^2$ be such that $x \in [\lfloor w_{\min} \rfloor, \lceil n\,w_{\max} \rceil] \cup \{0\}$, $y \in [\lfloor v_{\min} \rfloor, \lceil n\,v_{\max} \rceil] \cup \{0\}$, $x \leq W(S) \leq (1+\rho)\,x$ and $y/(1+\rho) \leq V(S) \leq y$. Since we have $W(S) \in [w_{\min}, n\,w_{\max}] \cup \{0\}$ and $V(S) \in [v_{\min}, n\,v_{\max}] \cup \{0\}$, there always exists such $(x,y) \in \textsc{Dom}^2$. In this case, since $W(S) \geq x$ and $V(S) \leq y$, by Lemma 3.5.2, the candidate assortment $\widehat{S}_{x,y}$ satisfies $W(\widehat{S}_{x,y}) \geq \frac{1}{(1+\rho)^n}\,x$, $V(\widehat{S}_{x,y}) \leq (1+\rho)^n\,y$ and $C(\widehat{S}_{x,y}) \leq C(S)$. Also, noting the fact that $W(S) \leq (1+\rho)\,x$ and $V(S) \geq y/(1+\rho)$, the last two inequalities yield $W(\widehat{S}_{x,y}) \geq \frac{1}{(1+\rho)^{n+1}}\,W(S)$ and $V(\widehat{S}_{x,y}) \leq (1+\rho)^{n+1}\,V(S)$. For all $\delta \in [0,1/2]$ and $n \in \mathbb{Z}_+$, we have the standard inequalities $(1+\delta/n)^n \leq$

$\exp(\delta) \le 1 + 2\delta$. Thus, since $\epsilon/8 \le 1/2$, we get

$$W(\widehat{S}_{x,y}) \;\ge\; \frac{1}{(1+\rho)^{n+1}}\, W(S) \;=\; \frac{1}{\left(1+\frac{\epsilon}{8\,(n+1)}\right)^{n+1}}\, W(S)$$
$$\ge\; \frac{1}{1+\epsilon/4}\, W(S) \;\ge\; (1-\epsilon/4)\, W(S),$$
$$V(\widehat{S}_{x,y}) \le (1+\rho)^{n+1}\, V(S) = \left(1+\frac{\epsilon}{8\,(n+1)}\right)^{n+1} V(S) \le (1+\epsilon/4)\, V(S).$$

Thus, given an assortment $S \subseteq \{j+1,\ldots,\ell\}$, there exists $\widehat{S}_{x,y} \in \textsc{Cand}(j,\ell)$ such that $W(\widehat{S}_{x,y}) \ge (1-\epsilon/4)\, W(S)$, $V(\widehat{S}_{x,y}) \le (1+\epsilon/4)\, V(S)$ and $C(\widehat{S}_{x,y}) \le C(S)$. So, the collection of candidate assortments $\textsc{Cand}(j,\ell)$ in (3.11) satisfies the limited degradation property in Part 1.

We close this section by arguing that we can construct all collections of candidate assortments $\{\textsc{Cand}(j,\ell) : j,\ell \in \{0,\ldots,n\},\ j \le \ell\}$ in $O(\frac{n^4}{\epsilon^2}\log(\frac{n w_{\max}}{w_{\min}})\log(\frac{n v_{\max}}{v_{\min}}))$ operations. By (3.11), we need the value functions $\Theta_i^\ell(x,y)$ through the dynamic program in (3.10) for $(x,y) \in \textsc{Dom}^2$ such that $x \in [\lfloor w_{\min}\rfloor, \lceil n\, w_{\max}\rceil] \cup \{0\}$, $y \in [\lfloor v_{\min}\rfloor, \lceil n\, v_{\max}\rceil] \cup \{0\}$ and $i \in \mathcal{N}$, $\ell \in \{0,\ldots,n\}$ with $i \le \ell+1$. Thus, the largest values of x and y in the state variable (x,y) are, respectively, $\lceil n\, w_{\max}\rceil$ and $\lceil n\, v_{\max}\rceil$. Since $\lfloor a-b\rfloor \le a$ and $\lceil a-b\rceil \le a$ for $a \in \textsc{Dom}$ and $a,b \in \mathbb{R}_+$, from one decision epoch to another, the values of x and y in the state variable (x,y) in (3.10) goes down. Also, the boundary condition in (3.10) depends only on the sign of x and y. Thus, if the value of the state variable x goes below $\lfloor w_{\min}\rfloor$ but still strictly positive, then without loss of generality, we can bump the value of the state variable x up to $\lfloor w_{\min}\rfloor$, because offering any of the products would immediately turn the value of the state variable x to negative. Similarly, if the value of the state variable y goes below $\lfloor v_{\min}\rfloor$ but still strictly positive, then we can bump the value of the state variable y up to $\lfloor v_{\min}\rfloor$. Lastly, once the value of x and y in the state variable (x,y) turns negative, we do not need to keep their exact values since each component of the state variable can only go down and the boundary condition at state (x, y) with $y < 0$ always yields a value function of $+\infty$. So, since $\rho = \frac{1}{8\,(n+1)}\epsilon$, for each decision epoch, the number of state variables that we need to consider is

$$O\left(\frac{\log(\frac{nw_{\max}}{w_{\min}})}{\log(1+\rho)}\,\frac{\log(\frac{nv_{\max}}{v_{\min}})}{\log(1+\rho)}\right) \;=\; O\left(\frac{n^2}{\epsilon^2}\log\left(\frac{nw_{\max}}{w_{\min}}\right)\log\left(\frac{nv_{\max}}{v_{\min}}\right)\right).$$

We need to compute the value function $\Theta_i^\ell(x, y)$ for all values of the state variable (x, y), $i \in \mathcal{N}$ and $\ell \in \{0, \ldots, n\}$ with $i \leq \ell + 1$. Therefore, to construct the collections of candidate assortments $\{\textsc{Cand}(j, \ell) \,:\, j, \ell \in \{0, \ldots, n\}, j \leq \ell\}$, we can compute $\Theta_i^\ell(x, y)$ for all values of the state variable (x, y), $i \in \mathcal{N}$ and $\ell \in \{0, \ldots, n\}$ with $i \leq \ell + 1$ in $O(\frac{n^4}{\epsilon^2}\,\log(\frac{nw_{\max}}{w_{\min}})\,\log(\frac{nv_{\max}}{v_{\min}}))$ operations. The number of other operations to construct the collections of candidate assortments is negligible.

In the collection of candidate assortments $\textsc{Cand}(j, \ell)$, we have one assortment for each $(x, y) \in \textsc{Dom}^2$ such that $x \in [\lfloor w_{\min} \rfloor, \lceil n\, w_{\max} \rceil] \cup \{0\}$ and $y \in [\lfloor v_{\min} \rfloor, \lceil n\, v_{\max} \rceil] \cup \{0\}$. Therefore, each of the collection of candidate assortments $\textsc{Cand}(j, \ell)$ includes $O\Big(\frac{\log(\frac{nw_{\max}}{w_{\min}})}{\log(1+\rho)}\,\frac{\log(\frac{nv_{\max}}{v_{\min}})}{\log(1+\rho)}\Big) = O(\frac{n^2}{\epsilon^2}\,\log(\frac{nw_{\max}}{w_{\min}})\,\log(\frac{nv_{\max}}{v_{\min}}))$ assortments in it. This observation becomes useful in the next section when we discuss the number of operations to execute Part 2.

3.5.2 Combining Candidate Assortments

We focus on executing Part 2, which obtains an approximate solution to problem (3.9). We can solve problem (3.9) by using dynamic programming. The decision epochs are the stages. At decision epoch k, the action is the candidate assortment S_k offered, whereas the state variable keeps j_k such that $S_{k-1} \subseteq \{j_{k-1} + 1, \ldots, j_k\}$, the accumulated value of $\sum_{q=1}^{k-1} V(S_q)$ and a target expected revenue to generate from the future stages. So, we consider the dynamic program

$$\Psi_k(j,u,z) = \min_{\substack{(\ell,S)\,:\,\ell\in\{j,\ldots,n\},\\ S\,\in\,\textsc{Cand}(j,\ell)}} \left\{ C(S) + \Psi_{k+1}\Big(\ell\,,\,\lceil u + V(S)\rceil\,,\,\Big\lceil z - \frac{\lambda_k\,W(S)}{(1+u)\,(1+u+V(S))}\Big\rceil\Big)\right\} \tag{3.12}$$

with the boundary condition that $\Psi_{m+1}(j,u,z) = 0$ if $z \le 0$. Otherwise, we have $\Psi_{m+1}(j,u,z) = +\infty$. If we drop the round up operators on the right side of (3.12), then $\Psi_k(j,u,z)$ gives the smallest total capacity consumption of assortments $(S_k,\ldots,S_m)$ such that $S_\ell \in \textsc{Cand}(j_\ell, j_{\ell+1})$ for some $j = j_k \le j_{k+1} \le \ldots \le j_m$ and these assortments provide an expected revenue of at least z in stages $k,\ldots,m$, when the assortments $(S_1,\ldots,S_{k-1})$ offered in the previous stages satisfy $\sum_{q=1}^{k-1} V(S_q) = u$. In this case, the optimal objective value of problem (3.9) is given by $\max\{z \in \mathbb{R} : \Psi_1(0,0,z) \le b\}$. With the round up operator, the dynamic program in (3.12) is only an approximation.

To obtain an approximate solution to problem (3.9), we use the dynamic program in (3.12) to compute the value functions $\{\Psi_k(j,u,z) : j = 0,\ldots,n,\ (u,z) \in \textsc{Dom}^2,\ k \in \mathcal{M}\}$. Approximating the optimal objective value of problem (3.9) as $\widehat{z}_{\text{APP}} = \max\{z \in \textsc{Dom} : \Psi_1(0,0,z) \le b\}$, we start with the state $(0,0,\widehat{z}_{\text{APP}})$ and follow the optimal state-action pairs in the dynamic program in (3.12). In particular, we use the following algorithm.

Candidate Stitching:

Step 1. Set $\widehat{z}_{\text{APP}} = \max\{z \in \textsc{Dom} : \Psi_1(0,0,z) \le b\}$, $k = 1$, $\widehat{j}_k = 0$, $\widehat{u}_k = 0$ and $\widehat{z}_k = \widehat{z}_{\text{APP}}$.

Step 2. Set

$$(\widehat{j}_{k+1}, \widehat{S}_k) = \underset{\substack{(\ell, S)\,:\, \ell \in \{\widehat{j}_k, \ldots, n\},\\ S \in \textsc{Cand}(\widehat{j}_k, \ell)}}{\arg\min} \Bigg\{ C(S) + \Psi_{k+1}\Bigg(\ell\,,\, \lceil \widehat{u}_k + V(S) \rceil\,,\, \Bigg\lceil \widehat{z}_k - \frac{\lambda_k\, W(S)}{(1+\widehat{u}_k)\,(1+\widehat{u}_k+V(S))} \Bigg\rceil \Bigg) \Bigg\}.$$

Step 3. Set

$$\widehat{u}_{k+1} = \lceil \widehat{u}_k + V(\widehat{S}_k) \rceil \quad \text{and} \quad \widehat{z}_{k+1} = \Bigg\lceil \widehat{z}_k - \frac{\lambda_k\, W(\widehat{S}_k)}{(1+\widehat{u}_k)\,(1+\widehat{u}_k+V(\widehat{S}_k))} \Bigg\rceil.$$

Increase k by one. If $k < m + 1$, then go to Step 2.

Step 4. Return $(\widehat{S}_1, \ldots, \widehat{S}_m)$.

Throughout this section, we refer to the algorithm above as the candidate stitching algorithm, since this algorithm stitches together a solution to problem (3.9) using the candidate assortments for different stages. In the next lemma, we show useful properties of the output $(\widehat{S}_1, \ldots, \widehat{S}_m)$ of the candidate stitching algorithm. In particular, the output of this algorithm is feasible to problem (3.9), satisfying $\sum_{k \in \mathcal{M}} C(\widehat{S}_k) \leq b$. Furthermore, using $\textsc{Rev}(S_1, \ldots, S_m)$ to denote the expected revenue from the solution $(S_1, \ldots, S_m)$, $\widetilde{z}$ to denote the optimal objective value of problem (3.9) and $\widehat{z}_{\textsc{App}} = \max\{z \in \textsc{Dom} : \Psi_1(0, 0, z) \leq b\}$ to denote our approximation to the optimal objective value of problem (3.9), we have $\textsc{Rev}(\widehat{S}_1, \ldots, \widehat{S}_m) \geq \widehat{z}_{\textsc{App}} \geq \widetilde{z}/(1+\rho)^{3(m+1)}$. Thus, the expected revenue provided by the output of the candidate stitching algorithm is at least as large as our approximation to the optimal objective value of problem (3.9). Also, if ρ is small, then our approximation to the optimal objective value of problem (3.9) is not too far from the optimal objective value of this problem. The proof of the lemma uses induction over the stages and it is in Appendix B.4.

Lemma 3.5.3 (Stitching Candidates) *Let* $(\widehat{S}_1, \ldots, \widehat{S}_m)$ *be the output of the candidate stitching algorithm,* $\widetilde{z}$ *be the optimal objective of problem* (3.9) *and* $\widehat{z}_{\text{APP}} = \max\{z \in \text{DOM} : \Psi_1(0, 0, z) \le b\}$. *We have* $\sum_{k \in \mathcal{M}} C(\widehat{S}_k) \le b$ *and* $\text{REV}(\widehat{S}_1, \ldots, \widehat{S}_m) \ge \widehat{z}_{\text{APP}} \ge \widetilde{z}/(1+\rho)^{3m+1}$.

We discuss how to use the lemma and the candidate stitching algorithm to execute Part 2. Given $\epsilon \in (0,1)$, we set the size of the geometric grid as $\rho = \frac{1}{8(3m+1)}\epsilon$. Since $\epsilon/8 \le 1/2$ and $(1+\delta/n)^n \le \exp(\delta) \le 1 + 2\delta$ for all $\delta \in [0, 1/2]$ and $n \in \mathbb{Z}_+$, by Lemma 3.5.3, we get

$$\text{REV}(\widehat{S}_1, \ldots, \widehat{S}_m) \ge \frac{1}{(1+\rho)^{3m+1}}\widetilde{z} = \frac{1}{(1+\frac{\epsilon}{8(3m+1)})^{3m+1}}\widetilde{z} \ge \frac{1}{1+\frac{\epsilon}{4}}\widetilde{z} \ge \Big(1 - \frac{\epsilon}{4}\Big)\widetilde{z},$$

so the output of the candidate stitching algorithm is a $(1 - \frac{\epsilon}{4})$-approximate solution to problem (3.9), as desired. Lastly, we argue that we can execute the candidate stitching algorithm with $\rho = \frac{1}{8(3m+1)}\epsilon$ in $O(\frac{n^4 m^3}{\epsilon^4}\log(\frac{n w_{\max}}{w_{\min}})\log(\frac{n w_{\max}(1 \vee n v_{\max})^2}{\lambda_m w_{\min}})\log^2(\frac{n v_{\max}}{v_{\min}}))$ operations. We account for the number of operations to solve the dynamic program in (3.12). The number of other operations to execute the candidate stitching algorithm is negligible. A simple lemma, given as Lemma B.5.1 in Appendix B.5, shows that if we compute $\{\widehat{u}_k : k = 1, \ldots, m+1\}$ as $\widehat{u}_{k+1} = \lceil \widehat{u}_k + V(S_k) \rceil$ with $\widehat{u}_1 = 0$ and $S_k \cap S_q = \varnothing$ for all $k \neq q$, then $\widehat{u}_k \le 2n\, v_{\max}$ for all $k \in \mathcal{M}$. Thus, the largest value of u in the state variable (j, u, z) in the dynamic program in (3.12) is $2n\, v_{\max}$. The smallest strictly positive value of u in the state variable (j, u, z) is $v_{\min}$. On the other hand, if the initial state variable (j, u, z) satisfies $z > n\, w_{\max}$, then since $\sum_{k \in \mathcal{M}} W(S_k) \le n\, w_{\max}$ for any $(S_1, \ldots, S_m)$ with $S_k \cap S_q = \varnothing$ for all $k \neq q$, no matter which assortments we offer, the final state variable (j, u, z) satisfies $z > 0$, in which case, the value function $\Psi_1(0, 0, z)$ takes the value $+\infty$. Thus, we do not need to consider the values of z that exceed $n\, w_{\max}$ in the state variable (j, u, z).

Finally, if the value of z in the state variable goes below $\lfloor \lambda_m\, w_{\min}/(1+2n\, v_{\max})^2 \rfloor$ but still strictly positive, then without of loss generality, we can bump the value of z up to $\lfloor \lambda_m\, w_{\min}/(1+2n\, v_{\max})^2 \rfloor$ since offering any non-empty candidate assortment would immediately turn the value of the state variable to negative. Thus, noting that j in the state variable (j, u, z) takes $O(n)$ possible values and $\rho = \frac{1}{8(3m+1)}\,\epsilon$, for each decision epoch, the number of state variables we need to consider in the dynamic program in (3.12) is

$$O\left(n \frac{\log(\frac{nv_{\max}}{v_{\min}})}{\log(1+\rho)}\, \frac{\log(\frac{nw_{\max}}{\lambda_m\, w_{\min}/(1+2n\, v_{\max})^2})}{\log(1+\rho)}\right) = O\left(\frac{n\, m^2}{\epsilon^2} \log\left(\frac{nv_{\max}}{v_{\min}}\right) \log\left(\frac{nw_{\max}\,(1 \vee n\, v_{\max})^2}{\lambda_m\, w_{\min}}\right)\right).$$

In decision epoch k, there are $\sum_{p=\widehat{j}^k}^{n} \text{CAND}(\widehat{j}_k, p) = O(\frac{n^3}{\epsilon^2} \log(\frac{nw_{\max}}{w_{\min}}) \log(\frac{nv_{\max}}{v_{\min}}))$ possible actions. Also, there are m decision epochs. Therefore, we can solve the dynamic program in (3.12) in $O(\frac{n^4 m^3}{\epsilon^4} \log(\frac{nw_{\max}}{w_{\min}}) \log(\frac{nw_{\max}\,(1 \vee nv_{\max})^2}{\lambda_m\, w_{\min}}) \log^2(\frac{nv_{\max}}{v_{\min}}))$ operations.

In Appendix B.6, we specialize our FPTAS to the case where there is a constraint on the total number of offered products and slightly improve its running time. For this case, we also give an exact algorithm to find the optimal assortment when the number of stages is fixed.

3.6 Computational Experiments

We provide three sets of computational experiments. First, we use a dataset publicly released by Expedia to check the ability our choice model to predict the purchase behavior of the customers. Second, we test the quality of the solutions from our greedy search heuristic for the joint assortment optimization and pricing problem. Third, we test the quality of the solutions obtained by our FTPAS under space constraints. As we discuss the second and third sets of computational experiments, we develop efficiently-computable upper bounds on the optimal expected revenue. We compare the expected revenue from the solutions obtained by our heuristic and FPTAS with the upper bounds on the optimal expected revenue.

3.6.1 Prediction Ability on the Dataset from Expedia

We use a dataset provided by Expedia as a part of a Kaggle competition; see [27]. Our goal is to test the ability of our choice model to the predict the purchases of customers.

Experimental Setup: The dataset gives the results of search queries for hotels on Expedia. In the dataset, the rows correspond to different hotels that are displayed in different search queries. The columns give information on the attributes of the displayed hotel, the results the search query and the booking decision of the customer. We preprocess the dataset to remove the values that are either missing or not interpretable, in which case, we end up with 595,965 rows and 15 columns. The first three columns in the dataset include the following information. The first column is the unique code for each query. Using this column, we can have access to all hotels displayed in a particular search query, which is the set of products among which a particular customer makes a choice. The second column is an indicator for whether the customer booked the hotel in the search query. We use this column to identify the purchase of the customer. It is possible that a customer does not book any hotels in a search query. The third column is the display position of the hotel in the search query, which becomes useful when fitting our multinomial logit model with impatient customers. The remaining 12 columns give information on the characteristics of the hotel, such as the star rating, average review score and displayed price for the hotel. After processing the dataset, 595,965 rows represent 34,561 search queries. The average number of hotels available in a search query is 17.24, with the maximum being 37. In 83% of the search queries, the customer did not make a booking. In Appendix B.7, we give our approach for preprocessing the dataset and give a detailed listing the 15 columns that we use.

To enrich our experimental setup, we using bootstrapping on the data to generate multiple datasets. In each dataset, we vary the fraction of the search queries that did not result in a booking. There are a total of 10,000 search queries in each dataset that we bootstrap. Using P_0 to denote the fraction of the search queries that did not result in a booking, we sample $P_0 \times 10,000$ search queries among the Expedia search queries that did not results in a booking. Similarly, among the Expedia search queries that resulted in a booking, we sample $(1 - P_0) \times 10,000$ search queries. Putting these two samples together, we get a dataset with 10,000 search queries in which P_0 fraction of them did not result in a booking. For each value of P_0, we repeat the bootstrapping process 50 times so that we obtain 50 different datasets. We vary P_0 over $\{0.5, 0.7, 0.9\}$. In this way, we obtain a total of 150 datasets. In our choice model, we capture the preference weight of hotel i in a search query by $v_i = \exp(\beta_0 + \sum_{\ell=1}^{12} \beta_\ell \, x_\ell)$, where $(x_1, \ldots, x_{12})$ are the values in the last 12 columns and $(\beta_0, \beta_1, \ldots, \beta_{12})$ are coefficients that we estimate from the data. So, the parameters of our choice model are the coefficients $(\beta_0, \beta_1, \ldots, \beta_{12})$, along with the patience level distribution.

We randomly split each dataset into training, validation and testing data, each of which, respectively, includes 64%, 16% and 20% of the search queries. The data provides the display position of each hotel in the search query, but fitting our choice model requires having access to the stage in which each hotel is displayed. We proceed under the assumption that each stage corresponds to b hotels in consecutive display positions and choose the best value of b by using cross-validation. In particular, we work with values of $b \in \{1, 3, 5, 10, 20\}$. For each value of b, we use maximum likelihood to fit our choice model to the training data and check the log-likelihood of our fitted choice model on the validation data. We choose the value of b that provides the largest log-likelihood on the validation data. See, for example, [42] for fitting choice models by using maximum likelihood. As a benchmark, we also fit a standard multinomial logit model to the training data. The preference weight of hotel i under this choice model is $v_i = \exp(\beta_0 + \sum_{\ell=1}^{12} \beta_\ell \, x_\ell)$. In Appendix B.8, we compare the runtimes for fitting the two choice models.

Computational Results: We use IML and SML to, respectively, refer to our multinomial logit model with impatient customers and the standard multinomial logit model. We use two performance measures to compare the two fitted choice models. The first one is the out of sample log-likelihood on the testing data. The second one is the k-hit score on the testing data. To compute the k-hit score of the fitted IML model, we use $\mathcal{T}$ to denote the set of search queries in the testing data in which the customer made a booking. For each $t \in \mathcal{T}$, we let S_t be the assortment of hotels offered in this search query and i_t be the hotel booked. Using $\phi_i(S)$ to denote the purchase probability of hotel i within assortment S under the fitted IML model, for each t, we let A_t^k be the set of k alternatives with the largest purchase probabilities, which are given by the k largest elements of $\{\phi_i(S_t) : i \in S_t\}$. If $i_t \in A_t^k$, then the hotel booked in search query t has one of the k largest choice probabilities under the fitted IML model. Thus, the k-hit score of the fitted IML model is $\frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}\mathbf{1}(i_t \in A_t^k)$. We compute the k-hit score of the fitted SML model similarly. We use $k \in \{1, 2, 3\}$. For the k-hit score, we focus only on the search queries resulting in a booking. A large fraction of the customers do not book. So, if we included the search queries not resulting in a booking in the k-hit score, then the k-hit scores would be mainly driven by the customers who do not book, but we want to test our ability to predict the specific hotel booked.

We give our computational results in Table 3.1. Each row in the table corresponds to a different value of P_0. Recall that we generate 50 datasets for each value of P_0. In the top portion, we compare the out of sample log-likelihoods of the fitted choice models. The first column shows the number of datasets out of 50, where the out of sample log-likelihood of the fitted IML model is larger than that of SML. The second and third columns, respectively, show the average out of sample log-likelihood of the fitted IML and SML models, where the average is computed over the 50 datasets. The fourth and fifth columns, respectively, show the average and standard error of the percent gaps between the out of sample log-likelihoods of the two fitted choice models, where the standard error is the standard deviation of the percent gaps over the 50 datasets divided by $\sqrt{50}$. In bottom portion, we compare the k-hit scores. The first column shows the number of datasets out of 50, where the 1-hit score of the fitted IML model is larger than that of SML. The second column shows the average 1-hit score of the fitted IML model over the 50 datasets. The fourth and fifth columns, respectively, show the average and standard error of the percent gap between the 1-hit scores of the two fitted choice models. In the rest of the

P_0	Out of Sample Log-Likelihood IML> SML	IML Like.	SML Like.	Avg. %Gap	S. Er. %Gap
0.5	50	−3899.65	−3963.64	1.64%	0.07%
0.7	50	−2701.65	−2766.36	2.40%	0.09%
0.9	47	−1145.78	−1169.93	2.12%	0.18%

	1-Hit Score				2-Hit Score				3-Hit Score			
P_0	IML> SML	IML 1-hit	Avg. %Gap.	Std. Err.	IML> SML	IML 2-hit	Avg. %Gap	Std. %Gap	IML> SML	IML 3-hit	Avg. %Gap.	Std. %Gap
0.5	36	0.25	3.81%	0.88%	37	0.39	3.48%	0.61%	36	0.50	2.04%	0.57%
0.7	42	0.24	6.61%	0.91%	42	0.37	4.24%	0.71%	32	0.48	3.11%	0.68%
0.9	34	0.22	3.04%	2.08%	35	0.36	5.80%	1.57%	40	0.47	7.74%	1.16%

Table 3.1: Comparison of the fitted IML and SML models on the dataset from Expedia.

table, we compare the 2-hit and 3-hit scores similarly.

The fitted IML model improves the out of sample log-likelihoods of the fitted SML model in 147 out of 150 datasets with an average gap of 2.05%. We expect a choice model with a larger out of sample log-likelihood to provide better prediction accuracy for the purchases of the customers, but to clearly quantify the improvements in the prediction accuracies, we turn to k-hit scores. The fitted IML model improves the 1-hit score of the fitted SML model in 112 out of 150 datasets with an average gap of 4.49%. Noting the 3-hit scores, one of the three alternatives with the largest purchase probabilities end up being the hotel booked by the customer about 50% of the time. The gaps between the k-hit scores are maintained for $k \in \{2, 3\}$, though for large values of k, the k-hit scores for both choice models will naturally be one. We can check that all gaps in the out of sample log-likelihoods and k-hit scores are statistically significant at 99% level, except for one. For $P_0 = 0.9$, the gap in the 1-hit scores is statistically significant at 90% level.

3.6.2 Joint Pricing and Assortment Optimization

For joint pricing and assortment optimization, to evaluate the quality of the solutions provided by our heuristic, we give an efficiently-computable upper bound on the optimal expected revenue.

Upper Bound: For given assortments $(S_1, \ldots, S_m)$ and no-purchase probabilities $\boldsymbol{q}$ satisfying $q_{k-1} \geq q_k$ for all $k \in \mathcal{M}$, the expected revenue is given by (3.6). Making its dependence on the assortments explicit, we use $\widehat{\Pi}(S_1, \ldots, S_m, \boldsymbol{q})$ to denote the expected revenue in (3.6). We construct an upper bound on the expected revenue by treating $\sum_{i \in S_k} e^{\alpha_i}$ in (3.6) as a continuous quantity.

Specifically, letting $T = \sum_{i \in \mathcal{N}} e^{\alpha_i}$, for each $(S_1, \ldots, S_m) \in \mathcal{F}$, we have $\sum_{k \in \mathcal{M}} \sum_{i \in S_k} e^{\alpha_i} \leq T$. In this case, using the decision variables $\boldsymbol{x} = (x_1, \ldots, x_m)$, by (3.6), we have

$$\widehat{\Pi}(S_1, \ldots, S_m, \boldsymbol{q}) \leq \frac{1}{\beta} \max_{\boldsymbol{x} \in \mathbb{R}^m_+} \left\{ \sum_{k \in \mathcal{M}} \lambda_k \, (q_{k-1} - q_k) \left\{ \log x_k - \log\left(\frac{1}{q_k} - \frac{1}{q_{k-1}} \right) \right\} \right.$$
$$\left. : \sum_{k \in \mathcal{M}} x_k \leq T \right\},$$

where we use the fact that $(\sum_{i \in S_1} e^{\alpha_i}, \ldots, \sum_{i \in S_m} e^{\alpha_i})$ is a feasible but not necessarily an optimal solution to the problem on the right side above.

Using the Lagrange multiplier $\alpha \geq 0$, we relax the constraint $\sum_{k \in \mathcal{M}} x_k \leq T$. Thus, for any $\alpha \geq 0$, we can upper bound on the optimal objective value of the problem on the right side above by

$$\frac{1}{\beta} \max_{\boldsymbol{x} \in \mathbb{R}^m_+} \left\{ \sum_{k \in \mathcal{M}} \lambda_k \, (q_{k-1} - q_k) \left\{ \log x_k - \log\left(\frac{1}{q_k} - \frac{1}{q_{k-1}} \right) \right\} - \sum_{k \in \mathcal{M}} \alpha \, x_k + \alpha \, T \right\},$$

The problem above decomposes by the stages. By the first order condition for the problem $\max_{x_k \in \mathbb{R}_+} \lambda_k \, (q_{k-1} - q_k) \, \log x_k - \alpha x_k$, the optimal value of x_k is $\lambda_k \, (q_{k-1} - q_k)/\alpha$.

Plugging the optimal value of x_k into objective function of the last problem above, the optimal objective value of this problem is

$$\frac{1}{\beta}\sum_{k\in\mathcal{M}}\lambda_k\,(q_{k-1}-q_k)\left\{\log\frac{\lambda_k\,(q_{k-1}-q_k)}{\alpha}-\log\left(\frac{1}{q_k}-\frac{1}{q_{k-1}}\right)-1\right\}+\frac{\alpha\,T}{\beta}.$$

By the discussion so far, for any $\alpha \geq 0$, the quantity above provides an upper bound on $\widehat{\Pi}(S_1,\ldots,S_m,\boldsymbol{q})$, as long as $(S_1,\ldots,S_m)\in\mathcal{F}$ and $q_{k-1}\geq q_k$ for all $k\in\mathcal{M}$. We simplify the quantity above by noting that $\log\frac{\lambda_k\,(q_{k-1}-q_k)}{\alpha}-\log(\frac{1}{q_k}-\frac{1}{q_{k-1}})-1 = \log(q_{k-1}\,q_k)+\log\frac{\lambda_k}{\alpha}-1$. Thus, we can upper bound the optimal expected revenue in the joint pricing and assortment problem as

$$\begin{aligned}&\max_{(S_1,\ldots,S_m,\boldsymbol{q})\in\mathcal{F}\times\mathbb{R}^m_+}\left\{\widehat{\Pi}(S_1,\ldots,S_m,\boldsymbol{q})\ :\ q_{k-1}\geq q_k\ \ \forall\,k\in\mathcal{M}\right\}\\ \leq\ &\frac{1}{\beta}\max_{\boldsymbol{q}\in\mathbb{R}^m_+}\left\{\sum_{k\in\mathcal{M}}\lambda_k\,(q_{k-1}-q_k)\,(\log(q_{k-1}\,q_k)+\log\tfrac{\lambda_k}{\alpha}-1)\ :\ q_{k-1}\geq q_k\ \forall\,k\in\mathcal{M}\right\}+\frac{\alpha\,T}{\beta}.\end{aligned} \tag{3.13}$$

In the problem on the right side above, intuitively speaking, only the no-purchase probabilities in two successive stages k and $k-1$ interact, which indicates that we can solve this problem by using dynamic programming. To obtain a dynamic program with a finite number of possible states, we discretize the state variable. It is never optimal to charge negative prices in the joint pricing and assortment problem, since dropping a product with a negative price always increases the expected revenue. Thus, we can lower bound the no-purchase probability in any stage as $q_k(\boldsymbol{\rho}) = 1/(1+\widehat{V}(\boldsymbol{\rho})) = 1/(1+\sum_{i\in S_k}e^{\alpha_i-\beta\rho_k}) \geq 1/(1+\sum_{i\in\mathcal{N}}e^{\alpha_i}) = \frac{1}{1+T}$. We divide the interval $[\frac{1}{1+T},1]$ into $L+1$ subintervals using $\nu_0,\ldots,\nu_{L+1}$ that satisfy $\frac{1}{1+T}=\nu_0<\nu_1<\ldots<\nu_L<\nu_{L+1}=1$. Let $G^\alpha_k(p,r)$ be an upper bound on $\lambda_k\,(q_{k-1}-q_k)\,(\log(q_{k-1}\,q_k)+\log\frac{\lambda_k}{\alpha}-1)$ for all $q_{k-1}\in[\nu_p,\nu_{p+1}]$ and $q_k\in[\nu_r,\nu_{r+1}]$. Coming up with the upper bound $G^\alpha_k(p,r)$ is not difficult. The first derivatives of $(q_{k-1}-q_k)\,\log(q_{k-1}\,q_k)$ with respect to

q_{k-1} and q_k are, respectively, negative and positive, so $(q_{k-1} - q_k) \log(q_{k-1} q_k)$ is decreasing in q_{k-1} and increasing in q_k Therefore, if $\log \frac{\lambda_k}{\alpha} - 1 \geq 0$, then we can set $G_k^\alpha(p, r) = \lambda_k (\nu_p - \nu_{r+1}) \log(\nu_p \nu_{r+1}) + \lambda_k (\nu_{p+1} - \nu_r) (\log \frac{\lambda_k}{\alpha} - 1)$. If $\log \frac{\lambda_k}{\alpha} - 1 < 0$, then we can set $G_k^\alpha(p, r) = \lambda_k (\nu_p - \nu_{r+1}) \log(\nu_p \nu_{r+1}) + \lambda_k (\nu_p - \nu_{r+1}) (\log \frac{\lambda_k}{\alpha} - 1)$. In our dynamic program, we focus on the possible intervals that can include the no-purchase probabilities $(q_1, \ldots, q_m)$. The decision epochs are the stages. The state at decision epoch k is the interval that includes q_{k-1}. The action at decision epoch k is the interval that includes q_k. Since the no-purchase probabilities in problem (3.13) satisfies $q_{k-1} \geq q_k$, we impose the constraint that the interval that includes q_k should not lie to the right of the interval that includes q_{k-1}. Thus, we consider the dynamic program

$$J_k^\alpha(p) = \max_{r \in \{0, \ldots, p\}} \left\{ G_k^\alpha(p, r) + J_{k+1}^\alpha(r) \right\}$$

with the boundary condition that $J_{m+1}^\alpha(p) = \alpha T$. Next, we show that $J_1^\alpha(L)$ is an upper bound on the optimal objective value of the pricing and assortment optimization problem.

Proposition 3.6.1 (Upper Bound for Joint Problem) *For each $\alpha \geq 0$, $\frac{1}{\beta} J_1^\alpha(L)$ is an upper bound on the optimal expected revenue in the* PRICING-ASSORTMENT *problem.*

The proof of the proposition is in Appendix B.9. Note that $\frac{1}{\beta} J_1^\alpha(L)$ is an upper bound for any $\alpha \geq 0$, so computing $\frac{1}{\beta} J_1^\alpha(L)$ for any $\alpha \geq 0$ provides an upper bound on the optimal expected revenue. To get a reasonably tight upper bound, we use a few iterations of golden ratio search to approximately solve the problem $\frac{1}{\beta} \min_{\alpha \geq 0} J_1^\alpha(L)$. In our computational experiments, we choose $\nu_0, \ldots, \nu_{L+1}$ such that $\nu_{p+1} - \nu_p$ is approximately equal to 0.001 for all $p = 0, \ldots, L$.

Experimental Setup: We randomly generate a large number of test problems and compare the expected revenue from the solution obtained by our greedy search heuristic with the upper bound on the optimal expected revenue. In all of our test problems, the number of products is $n = 20$ and the price sensitivity is $\beta = 1$. Working with other values for the price sensitivity is equivalent to scaling the prices of the products with the same constant. We use the following approach to come up with the parameters $\{\alpha_i : i \in \mathcal{N}\}$. We have C product clusters. We randomly assign each product to a cluster. If products i and j are in the same cluster, then the values of α_i and α_j are close. In particular, cluster c has the centroid γ_c. We set the centroid of cluster c as $\gamma_c = c - 0.5$ for all $c = 1, \ldots, C$. If product i belongs to cluster c, then we generate κ_i from the normal distribution with mean γ_c and standard deviation σ, where σ is a parameter that we vary. We set $\alpha_i = \kappa_i - \Delta$, where we have $\Delta = \log \sum_{i \in \mathcal{N}} e^{\kappa_i} - \log 9$. In this case, if we offer all products in the first stage and charge a price of zero for them, then a customer leaves without a purchase with probability 0.1. Using the random variable Y to capture the patience level of a customer, the probability mass function of Y is $\mathbb{P}\{Y = k\} = e^{a \times k} / \sum_{\ell \in \mathcal{M}} e^{a \times \ell}$, where a is another parameter that we vary. Negative and positive values for a yield, respectively, left and right skewed distributions. If $a = +\infty$, then $Y = m$ with probability one, so the customers always leave after the last stage.

Recalling that m is the number of stages, varying $m \in \{6, 8, 10\}$, $C \in \{3, 5\}$, $\sigma \in \{0.5, 1.0\}$ and $a \in \{+\infty, 0.5, 0.0, -0.1\}$, we obtain 48 parameter configurations. In each parameter configuration, we generate 25 problem instances using the approach in the previous paragraph.

Computational Results: We show our computational results in Table 3.2.

In this table, the first column shows the parameter configuration by using the tuple (C, σ, a), where C, σ and a are as discussed in our experimental setup. In the rest of the table, there are three blocks, each with three columns. Each block corresponds to a particular value for the parameter m. In each block, the first column shows the average percent gap between the upper bound on the optimal expected revenue and the expected revenue from the solution obtained by our greedy search heuristic, where the average is computed over the 25 problem instances in a particular parameter configuration. In particular, using Rev^k and UB^k to, respectively, denote the expected revenue from the solution obtained by the greedy search heuristic and the upper bound on the optimal expected revenue for problem instance k, the first column shows the average of the data $\{100 \times \frac{\mathsf{Rev}^k - \mathsf{UB}^k}{\mathsf{UB}^k} : k = 1, \ldots, 25\}$. The second and third columns, respectively, show the maximum and standard deviation of the same data. Our result indicate that our heuristic performs quite well. Over all of our test problems, the average gap between the upper bound on the optimal expected revenue and the expected revenue from the greedy search heuristic is 1.43%, whereas the largest gap is 4.09%. We observe that the gaps tend to increase as the number of stages gets larger. Without knowing the optimal expected revenue, it is difficult to tell whether the increase in the gaps are due to a degradation in the upper bounds or a degration in the expected revenue from the greedy search heuristic. However, we note that the upper bound on the optimal expected revenue is based on treating $(\sum_{i \in S_1} e^{\alpha_i}, \ldots, \sum_{i \in S_m} e^{\alpha_i})$ in (3.6) as continuous quantities whose sum does not exceed $\sum_{i \in \mathcal{N}} e^{\alpha_i}$. Intuitively speaking, this assumption becomes harder to justify when the number of stages gets larger. Overall, the performance of the heuristic is substantially better than its 50% theoretical performance guarantee.

The number of iterations until termination for the heuristic varies between

Param. Conf.	$m = 6$			$m = 8$			$m = 10$		
(C, σ, a)	Avg.	Max.	Std.	Avg.	Max.	Sd.	Avg.	Max.	Std.
$(3, 0.5, +\infty)$	1.38%	1.59%	0.09%	1.95%	2.17%	0.09%	2.52%	2.74%	0.09%
$(3, 0.5, 0.5)$	1.15%	1.17%	0.01%	1.63%	1.66%	0.01%	2.12%	2.22%	0.03%
$(3, 0.5, 0.0)$	0.78%	0.79%	0.00%	1.04%	1.10%	0.01%	1.29%	1.38%	0.02%
$(3, 0.5, -0.1)$	0.70%	0.70%	0.00%	0.89%	0.90%	0.01%	1.06%	1.07%	0.00%
$(3, 1.0, +\infty)$	1.53%	2.61%	0.30%	2.10%	3.18%	0.31%	2.67%	3.75%	0.31%
$(3, 1.0, 0.5)$	1.18%	1.57%	0.08%	1.66%	2.12%	0.12%	2.20%	2.86%	0.17%
$(3, 1.0, 0.0)$	0.78%	0.90%	0.02%	1.03%	1.04%	0.00%	1.29%	1.38%	0.02%
$(3, 1.0, -0.1)$	0.70%	0.70%	0.00%	0.89%	0.92%	0.01%	1.06%	1.14%	0.02%
$(5, 0.5, +\infty)$	1.49%	1.77%	0.16%	2.06%	2.34%	0.17%	2.63%	2.92%	0.17%
$(5, 0.5, 0.5)$	1.16%	1.18%	0.01%	1.64%	1.72%	0.03%	2.15%	2.27%	0.05%
$(5, 0.5, 0.0)$	0.78%	0.79%	0.00%	1.04%	1.09%	0.02%	1.29%	1.33%	0.01%
$(5, 0.5, -0.1)$	0.70%	0.71%	0.00%	0.89%	0.91%	0.01%	1.06%	1.11%	0.02%
$(5, 1.0, +\infty)$	1.84%	2.95%	0.46%	2.41%	3.52%	0.46%	2.98%	4.09%	0.46%
$(5, 1.0, 0.5)$	1.21%	2.28%	0.22%	1.78%	2.59%	0.24%	2.36%	3.25%	0.31%
$(5, 1.0, 0.0)$	0.78%	0.86%	0.02%	1.07%	1.21%	0.06%	1.31%	1.46%	0.05%
$(5, 1.0, -0.1)$	0.70%	0.75%	0.01%	0.88%	0.89%	0.00%	1.07%	1.40%	0.07%

Table 3.2: Performance of the greedy search heuristic for joint pricing and assortment optimization.

one and 40 in our test problems, whereas the runtime varies between 0.11 and 5.14 seconds. The larger numbers of iterations and runtimes correspond to test problems with larger number of stages.

3.6.3 Assortment Optimization under a Space Constraint

We can efficiently compute an upper bound on the optimal expected revenue under a space constraint by building on the dynamic programs that we use in Sections 3.5.1 and 3.5.2. We defer the discussion of the upper bound to Appendix B.10.

Experimental Setup: In all of our test problems, the number of products is $n = 20$. We use the following approach to come up with the revenues $\{r_i : i \in \mathcal{N}\}$ and preference weights $\{v_i : i \in \mathcal{N}\}$. We generate r_i from the uniform distribution over $[1, 10]$. We reindex $(r_1, \ldots, r_n)$ so that $r_1 \geq r_2 \geq \ldots \geq r_n$. To come up with the preference weights, we generate γ_i from the uniform distribution $[1, 10]$. We set $v_i = \gamma_i/\Delta$, where $\Delta = P_0 \sum_{i \in \mathcal{N}} \gamma_i/(1 - P_0)$ and P_0 is a parameter that we vary. In this case, if we offer all products in the first stage, then a customer leaves without a purchase with probability P_0. After generating the preference weights, we post-process them to come up with two problem classes for the preference weights. In the first problem class, we leave the preference weights untouched. In the second problem class, we reindex $(v_1, \ldots, v_n)$ so that $v_1 \leq v_2 \leq \ldots \leq v_n$. Therefore, in the second problem class, the products with larger revenues tend to have smaller preference weights, so the more expensive products tend to be less attractive. We refer to the first and second problem classes, respectively, as "U" and "O," where "U" stands for unordered and "O" stands for ordered. We use the same approach that we use for joint pricing and assortment optimization to come up with the distribution for the patience levels. Recall that the parameter a controls the skewness of the distribution of the patience levels. In all of our test problems, to come up with the space consumptions $\{c_i : i \in \mathcal{N}\}$ and the space availability b, we generate c_i from the uniform distribution over $[0, 1]$ and set $b = 5$.

Using T denote the problem class for the preference weights, varying $m \in \{6, 8, 10\}$, $P_0 \in \{0.1, 0, 3\}$, $T \in \{\text{U}, \text{O}\}$ and $a \in \{+\infty, 0.5, 0, -0.1\}$, we obtain 48 parameter configurations. We generate 25 problem instances in each parameter configuration.

Computational Results: We executed our FPTAS with $\epsilon = 1/2$ to obtain a $\frac{1}{2}$-approximate solution to the CAPACITATED problem. Even with this setting, our FPTAS obtains solutions that provide expected revenues within 5% of the upper bound on the optimal expected revenue. The large number of test problems that we use in our experimental setup prevented us from reporting results for theoretical performance guarantees that are better than 50%, but a limited number of runs indicated that if we decrease ϵ to 1/4, we improve the percent gap between the upper bound and the performance of our FPTAS by about 1%. We show our computational results in Table 3.3. The layout of this table is identical to that of Table 3.2. The only difference is that we use the tuple (P_0, T, a) in the first column to show the parameter configuration and the percent gaps correspond to those between the upper bound on the optimal expected revenue and the expected revenue from the solution obtained by our FPTAS. Over all of our test problems, the average gap between the upper bound and the expected revenue from our FPTAS is 2.25%, whereas the maximum gap is 4.47%. The gaps increase only slightly as the number of stages gets larger. Overall, the performance of our FPTAS is substantially stronger than its theoretical performance guarantee of 50%.

The runtime for our FPTAS ranges between 26.23 minutes to 36.12 minutes. Note that our test problems are simply outside the range of solution through full enumeration since the number of possible sequences of assortments is $O(m^n)$. Considering the candidate construction and candidate stitching algorithms in Sections 3.5.1 and 3.5.2, a major portion of the runtime is spent on candidate construction. The runtime for the candidate stitching algorithm varies between 2.08 to 18.09 seconds, where the larger runtimes correspond to the test problems with the larger number of stages. The remaining portion of the runtime is for the candidate construction algorithm.

Param. Conf.	$m = 6$			$m = 8$			$m = 10$		
(P_0, T, a)	Avg.	Max.	Std.	Avg.	Max.	Sd.	Avg.	Max.	Std.
$(0.1, \text{U}, +\infty)$	2.92%	3.88%	0.56%	3.18%	4.20%	0.56%	3.27%	4.31%	0.58%
$(0.1, \text{U}, 0.5)$	2.40%	3.22%	0.42%	2.53%	3.44%	0.56%	2.79%	3.71%	0.47%
$(0.1, \text{U}, 0.0)$	2.17%	3.09%	0.49%	2.28%	3.71%	0.51%	2.19%	3.73%	0.65%
$(0.1, \text{U}, -0.1)$	2.12%	3.09%	0.55%	2.09%	3.07%	0.50%	2.11%	3.21%	0.56%
$(0.1, \text{O}, +\infty)$	2.77%	4.00%	0.60%	3.11%	4.27%	0.60%	3.30%	4.47%	0.59%
$(0.1, \text{O}, 0.5)$	2.14%	2.77%	0.40%	2.38%	3.72%	0.60%	2.72%	4.00%	0.57%
$(0.1, \text{O}, 0.0)$	1.96%	3.32%	0.62%	2.15%	3.34%	0.61%	1.99%	3.81%	0.53%
$(0.1, \text{O}, -0.1)$	2.25%	3.54%	0.53%	2.01%	2.93%	0.54%	2.03%	3.04%	0.54%
$(0.3, \text{U}, +\infty)$	2.52%	3.58%	0.45%	2.80%	3.84%	0.43%	2.93%	4.08%	0.46%
$(0.3, \text{U}, 0.5)$	2.15%	3.22%	0.52%	2.25%	3.01%	0.45%	2.16%	3.03%	0.47%
$(0.3, \text{U}, 0.0)$	1.87%	2.71%	0.40%	1.78%	2.59%	0.45%	1.87%	2.86%	0.49%
$(0.3, \text{U}, -0.1)$	1.80%	2.65%	0.43%	1.79%	2.71%	0.42%	2.02%	2.94%	0.45%
$(0.3, \text{O}, +\infty)$	2.37%	3.20%	0.47%	2.69%	3.58%	0.46%	2.85%	3.85%	0.46%
$(0.3, \text{O}, 0.5)$	1.74%	2.67%	0.50%	1.96%	3.19%	0.52%	2.07%	2.99%	0.43%
$(0.3, \text{O}, 0.0)$	1.59%	2.73%	0.55%	1.52%	2.48%	0.46%	1.73%	2.63%	0.53%
$(0.3, \text{O}, -0.1)$	1.50%	2.52%	0.53%	1.54%	2.47%	0.48%	1.55%	2.75%	0.52%

Table 3.3: Performance of the FPTAS for assortment optimization under a space constraint.

3.7 Conclusions

We developed a variant of the multinomial logit model with impatient customers. We gave an algorithm to compute the revenue-maximizing sequence of assortments offer. For fixed sequence of assortments, we showed that we can formulate the pricing problem as a convex program. When the prices, as well as the sequence of offered assortments, are decision variables, we developed a heuristic with 50% performance guarantee. We gave an FPTAS when there is a constraint on the total space consumption of the offered products. Our computational work on a dataset from Expedia indicated that our variant of the multinomial logit model can provide noticeable benefits in predicting the purchase behavior of the customers. As mentioned in the introduction, there are several open problems left by our work. First, it would be useful to find solution methods for the assortment problem when there is a constraint on the space consumption or the number of products offered in each stage. Our efforts to address this problem have so far been fruitless. Second, if there is a constraint on the total number of offered products, then we have an algorithm to find the optimal sequence of assortments to offer when the number of stages is fixed. The running time of this algorithm increases exponentially with the number of stages. We do not know the complexity of the assortment problem when the number of stages is also an input.

CHAPTER 4

CONCLUSION

In this thesis, we proposed two choice models under the multi-stage framework. These models can better characterize customers' choice behavior in the online setting, where the products are revealed sequentially as the customers browse through the webpages. We studied the assortment optimization and pricing problems under these models. In Chapter 2, we considered a multi-stage choice model, in which the choice process *within* each stage is driven by the multinomial logit model. In Chapter 3, we discussed a multi-stage choice model, in which the utility model under the *entire* choice process is similar as in the multinomial logit model. This thesis is only a starting point of possible choice models under the multi-stage framework. There are several potential directions to further extend our results.

First, both models in this thesis were generalized from the multinomial logit model. A useful line of research is to consider the generalizations of other single-stage choice models as we discussed in Chapter 1 such as the nested logit model, the mixed logit model, etc . Therefore, more complicated customer choice behavior could be captured under the multi-stage scheme.

Second, this thesis mainly focuses on the optimization part of the problems, where the preference weights of products are assumed given. However, in practice, we need to estimate preference weights from the transaction data as we did in the numerical experiments in Chapter 3. Then a natural question to investigate is the sensitivity of the optimal assortments under the current parameters: How would the optimal assortments change if we know some parameters are estimated incorrectly? It would also be interesting to consider a robust version of the assortment problem, where the preference weights of products are assumed taken from some uncertainty set.

Third, in both models of the thesis, the customers are assumed to be myopic in the sense they always pick the product with the largest utility at the current stage but never consider what could be offered in the future. It would be interesting to further model the customers as strategic and forward-looking. For example, if a customer believes there would be a product with higher utility offered in the following stages, she could still choose to continue to the next stage although the no-purchase option does not have the largest utility at the current stage. One potential way to do so is to assume that the utility of the ”continue” option at each stage is indeed a function of possible future assortments. Then the derived choice model could be very different from the ones presented here.

APPENDIX A

APPENDIX FOR CHAPTER 2

A.1 Upper Bound on State Variable

In this section, we give a proof for Lemma 2.6.1. First, we show that if $\widehat{f^k} > \lceil nR_{\max} \rceil$, then we have $f^k_{n+1} > 0$. Since $f^k_{i+1} = \lceil f^k_i - r_i\, v^k_i\, x^k_i \rceil$, we have $f^k_{i+1} \geq f^k_i - r_i\, v^k_i\, x^k_i$. Adding this inequality over all $i \in N$ and noting that $f^k_1 = \widehat{f^k}$, along with the definition of $R_{\max}$, we obtain $f^k_{n+1} \geq \widehat{f^k} - \sum_{i \in N} r_i\, v^k_i\, x^k_i \geq \widehat{f^k} - nR_{\max} \geq \widehat{f^k} - \lceil nR_{\max} \rceil$. In this case, the last inequality implies that if $\widehat{f^k} > \lceil nR_{\max} \rceil$, then we have $f^k_{n+1} > 0$. Second, we show that if $\widehat{h^k} \geq \lceil \Delta(\rho, n)\, V_{\max} \rceil$, then we have $h^k_{n+1} \geq 0$. We claim that if $h^k_i \geq \Delta(\rho, n+1-i)\, V_{\max}$, then $h^k_{i+1} \geq \Delta(\rho, n-i)\, V_{\max}$. To see the claim, if $h^k_i \geq \Delta(\rho, n+1-i)\, V_{\max}$, then we have

$$\begin{aligned} h^k_{i+1} = \lfloor h^k_i - v^k_i\, x^k_i \rfloor &\geq \frac{1}{1+\rho}\,(h^k_i - v^k_i\, x^k_i) \geq \frac{1}{1+\rho}\,(\Delta(\rho, n+1-i) - 1)\, V_{\max} \\ &= \frac{\frac{(1+\rho)^{n+1-i}-1}{\rho} - 1}{1+\rho}\, V_{\max} = \frac{(1+\rho)^{n-i} - 1}{\rho}\, V_{\max} = \Delta(\rho, n-i)\, V_{\max}, \end{aligned}$$

where the first inequality holds since $h_i - v^k_i\, x^k_i \geq \Delta(\rho, n+1-i)\, V_{\max} - V_{\max} \geq 0$ and $\lfloor x \rfloor \geq x/(1+\rho)$ for any $x \in \Re_+$ and the second equality is by the definition of $\Delta(\rho, n)$. The chain of inequalities above establishes the claim. Using the claim, if $h^k_1 \geq \Delta(\rho, n)\, V_{\max}$, then $h^k_2 \geq \Delta(\rho, n-1)\, V_{\max}$, but using the claim once more, if $h^k_2 \geq \Delta(\rho, n-1)\, V_{\max}$, then $h^k_3 \geq \Delta(\rho, n-2)\, V_{\max}$. Using the claim successively, if $h^k_1 \geq \Delta(\rho, n)\, V_{\max}$, then $h^k_{n+1} \geq \Delta(\rho, 0)\, V_{\max}$. Since $h^k_1 = \widehat{h^k}$, if $\widehat{h^k} \geq \lceil \Delta(\rho, n)\, V_{\max} \rceil$, then $h^k_1 = \widehat{h^k} \geq \Delta(\rho, n)\, V_{\max}$, but if $h^k_1 \geq \Delta(\rho, n)\, V_{\max}$, then $h^k_{n+1} \geq \Delta(\rho, 0)\, V_{\max}$. By the definition of $\Delta(\rho, n)$, we have $\Delta(\rho, 0) = 0$, so it follows that if $\widehat{h^k} \geq \lceil \Delta(\rho, n)\, V_{\max} \rceil$, then $h^k_{n+1} \geq 0$.

A.2 Computation of Thresholds

In the next lemma, we show the relationship between the value functions $\{V_i(\cdot,\cdot) : i \in N\}$ and $\{J_i(\cdot,\cdot) : i \in N\}$ that are computed through the dynamic programs in (2.13) and (2.15).

Lemma A.2.1 *For any $(\boldsymbol{f}_i, \boldsymbol{h}_i) \in \mathrm{DOM}_+^m \times \mathrm{DOM}_+^m$ and $i \in N$, if $f_i^1 \leq \lfloor J_i(\boldsymbol{f}_i^{-1}, \boldsymbol{h}_i)\rfloor$, then we have $V_i(\boldsymbol{f}_i, \boldsymbol{h}_i) = 0$. Similarly, if $f_i^1 > \lfloor J_i(\boldsymbol{f}_i^{-1}, \boldsymbol{h}_i)\rfloor$, then we have $V_i(\boldsymbol{f}_i, \boldsymbol{h}_i) = -\infty$.*

Proof. We use induction over the products to show that if $f_i^1 \leq \lfloor J_i(\boldsymbol{f}_i^{-1}, \boldsymbol{h}_i)\rfloor$, then we have $V_i(\boldsymbol{f}_i, \boldsymbol{h}_i) = 0$ for any $(\boldsymbol{f}_i, \boldsymbol{h}_i) \in \mathrm{DOM}_+^m \times \mathrm{DOM}_+^m$ and $i \in N \cup \{n+1\}$. For any $(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1}) \in \mathrm{DOM}_+^m \times \mathrm{DOM}_+^m$, since $f_{n+1}^1 \geq 0$ and $J_{n+1}(\cdot,\cdot)$ takes only the value zero or $-\infty$, if $f_{n+1}^1 \leq \lfloor J_{n+1}(\boldsymbol{f}_{n+1}^{-1}, \boldsymbol{h}_{n+1})\rfloor$, then we must have $J_{n+1}(\boldsymbol{f}_{n+1}^{-1}, \boldsymbol{h}_{n+1}) = 0$ and $f_{n+1}^1 \leq 0$. By the boundary condition of the dynamic program in (2.15), if $J_{n+1}(\boldsymbol{f}_{n+1}^{-1}, \boldsymbol{h}_{n+1}) = 0$, then we must have $f_{n+1}^k \leq 0$ for all $k \in M^{-1}$ and $h_{n+1}^k \geq 0$ for all $k \in M$. Thus, if $f_{n+1}^1 \leq \lfloor J_{n+1}(\boldsymbol{f}_{n+1}^{-1}, \boldsymbol{h}_{n+1})\rfloor$, then we must have $f_{n+1}^1 \leq 0$, $f_{n+1}^k \leq 0$ for all $k \in M^{-1}$ and $h_{n+1}^k \geq 0$ for all $k \in M$, in which case, by the boundary condition of the dynamic program in (2.13), we have $V_{n+1}(\boldsymbol{f}_{n+1}, \boldsymbol{h}_{n+1}) = 0$. Therefore, the result holds for product $n+1$. Next, we assume that the result holds for product $i+1$ and we show that the result holds for product i. Consider $(\boldsymbol{f}_i, \boldsymbol{h}_i) \in \mathrm{DOM}_+^m \times \mathrm{DOM}_+^m$ such that $f_i^1 \leq \lfloor J_i(\boldsymbol{f}_i^{-1}, \boldsymbol{h}_i)\rfloor$. We use $\widehat{\boldsymbol{x}}_i$ to denote an optimal solution to the problem on the right side of (2.15). Since $f_i^1 \leq \lfloor J_i(\boldsymbol{f}_i^{-1}, \boldsymbol{h}_i)\rfloor$, noting the dynamic program in (2.15), we have

$$f_i^1 \leq \left\lfloor r_i\, v_i^1\, \widehat{x}_i^1 + \left\lfloor J_{i+1}\left(\left\lceil \boldsymbol{f}_i^{-1} - \sum_{k \in M^{-1}} \boldsymbol{e}^k\, r_i\, v_i^k\, \widehat{x}_i^k \right\rceil, \left\lfloor \boldsymbol{h}_i - \sum_{k \in M} \boldsymbol{e}^k\, v_i^k\, \widehat{x}_i^k \right\rfloor \right) \right\rfloor \right\rfloor.$$

By a simple lemma, given as Lemma A.2.2 below, for any $a, b \in \mathrm{DOM}$ and $\alpha \in \Re$, we have $a \leq \lfloor \alpha + b \rfloor$ if and only if $\lceil a - \alpha \rceil \leq b$. Thus, the inequality above implies that we have

$$\lceil f_i^1 - r_i\, v_i^1\, \widehat{x}_i^1 \rceil \le \left\lfloor J_{i+1}\left(\left\lceil \boldsymbol{f}_i^{-1} - \sum_{k \in M^{-1}} \boldsymbol{e}^k\, r_i\, v_i^k\, \widehat{x}_i^k \right\rceil, \left\lfloor \boldsymbol{h}_i - \sum_{k \in M} \boldsymbol{e}^k\, v_i^k\, \widehat{x}_i^k \right\rfloor\right)\right\rfloor.$$

For notational brevity, we let $f_{i+1}^k = \lceil f_i^k - r_i\, v_i^k\, \widehat{x}_i^k \rceil$ and $h_{i+1}^k = \lfloor h_i^k - v_i^k\, \widehat{x}_i^k \rfloor$ for all $k \in M$, in which case, the inequality above is equivalent to $f_{i+1}^1 \le \lfloor J_{i+1}(\boldsymbol{f}_{i+1}^{-1}, \boldsymbol{h}_{i+1}) \rfloor$, but by the induction argument, if $f_{i+1}^1 \le \lfloor J_{i+1}(\boldsymbol{f}_{i+1}^{-1}, \boldsymbol{h}_{i+1}) \rfloor$, then $V_{i+1}(\boldsymbol{f}_{i+1}, \boldsymbol{h}_{i+1}) = 0$. Therefore, noting the definitions of f_{i+1}^k and h_{i+1}^k, we have $V_{i+1}(\boldsymbol{f}_{i+1}, \boldsymbol{h}_{i+1}) = V_{i+1}(\lceil \boldsymbol{f}^k - \sum_{k \in M} \boldsymbol{e}^k\, r_i\, v_i^k\, \widehat{x}_i^k \rceil, \lfloor \boldsymbol{h}^k - \sum_{k \in M} \boldsymbol{e}^k\, v_i^k\, \widehat{x}_i^k \rfloor) = 0$. By last equality, the solution $\widehat{\boldsymbol{x}}_i$ provides an objective value of zero for the problem on the right side of (2.13). Since $V_i(\cdot, \cdot)$ takes only the value zero or $-\infty$ and there exists a solution to the problem on the right side of (2.13) that provides an objective value of zero, we must have $V_i(\boldsymbol{f}_i, \boldsymbol{h}_i) = 0$, completing the induction argument. The discussion so far shows that if $f_i^1 \le \lfloor J_i(\boldsymbol{f}_i^{-1}, \boldsymbol{h}_i) \rfloor$, then we have $V_i(\boldsymbol{f}_i, \boldsymbol{h}_i) = 0$ for any $(\boldsymbol{f}_i, \boldsymbol{h}_i) \in \text{DOM}_+^m \times \text{DOM}_+^m$ and $i \in N \cup \{n+1\}$, establishing the first statement in the lemma. The second statement uses a similar reasoning. ■

In the next lemma, we show a result that we use in the proof of Lemma A.2.1.

Lemma A.2.2 *For any $a, b \in \text{DOM}$ and $\alpha \in \Re$, we have $a \le \lfloor \alpha + b \rfloor$ if and only if $\lceil a - \alpha \rceil \le b$.*

Proof. First, we show that if $a \le \lfloor \alpha + b \rfloor$, then we have $\lceil a - \alpha \rceil \le b$. If $a \le \lfloor \alpha + b \rfloor$, then $a \le \alpha + b$, so that $a - \alpha \le b$. Since $b \in \text{DOM}$, having $a - \alpha \le b$ implies that $\lceil a - \alpha \rceil \le b$, as desired. Second, we show that if $\lceil a - \alpha \rceil \le b$, then we have $a \le \lfloor \alpha + b \rfloor$. If $\lceil a - \alpha \rceil \le b$, then $a - \alpha \le b$, so that $a \le \alpha + b$. Since $a \in \text{DOM}$, having $a \le \alpha + b$ implies that $a \le \lfloor \alpha + b \rfloor$, as desired. ■

A.3 Nested by Revenue Sets

We show that there exists an optimal solution $(\widehat{S}^1, \ldots, \widehat{S}^m)$ to problem (2.1) that satisfies $\cup_{k \in M} \widehat{S}^k = \{i \in N : r_i \geq \widehat{\zeta}\}$ for some constant $\widehat{\zeta}$. In other words, the union of the sets offered over all stages is a nested by revenue set. Therefore, if we index the products such that $r_1 \geq r_2 \geq \ldots \geq r_n$, then an optimal solution $(\widehat{S}^1, \ldots, \widehat{S}^m)$ to problem (2.1) is of the form $\cup_{k \in M} \widehat{S}^k = \{1, \ldots, i\}$ for some $i \in N$. Although this result gives some insight into the structure of the optimal solution, it does not allow us to obtain an optimal solution to problem (2.1) efficiently, since this result does not specify the stage in which each product should be offered. To show that there exists an optimal solution $(\widehat{S}^1, \ldots, \widehat{S}^m)$ to problem (2.1) that satisfies $\cup_{k \in M} \widehat{S}^k = \{i \in N : r_i \geq \widehat{\zeta}\}$ for some constant $\widehat{\zeta}$, we use a recursive version of the objective function of problem (2.1). We use $R^\nu(S^\nu, \ldots, S^m)$ to denote the expected revenue obtained from a customer starting her choice process in stage ν when we offer the sets $S^\nu, \ldots, S^m$ in stages $\nu, \ldots, m$. Thus, noting the expected revenue expression in (2.1) and focusing only on the stages $\nu, \ldots, m$, $R^\nu(S^\nu, \ldots, S^m)$ is given by

$$R^\nu(S^\nu, \ldots, S^m) = \sum_{k=\nu}^{m} \left\{ \prod_{\ell=\nu}^{k} \frac{1}{1 + \sum_{i \in S^\ell} v_i^\ell} \right\} \sum_{i \in S^k} r_i \, v_i^k. \tag{A.1}$$

Comparing the expression above with (2.1), note that $R^1(S^1, \ldots, S^m)$ corresponds to the objective function of problem (2.1). In the next proposition, we show that there exists an optimal solution $(\widehat{S}^1, \ldots, \widehat{S}^m)$ to problem (2.1) that satisfies $\cup_{k \in M} \widehat{S}^k = \{i \in N : r_i \geq \widehat{\zeta}\}$ for some constant $\widehat{\zeta}$ and the constant $\widehat{\zeta}$ is given by $\min\{R^k(\widehat{S}^k, \ldots, \widehat{S}^m) : k \in M\}$.

Proposition A.3.1 *There exists an optimal solution* $(\widehat{S}^1, \ldots, \widehat{S}^m)$ *to problem* (2.1) *that satisfies* $\cup_{k \in M} \widehat{S}^k = \{i \in N : r_i \geq \widehat{\zeta}\}$, *where* $\widehat{\zeta} = \min\{R^k(\widehat{S}^k, \ldots, \widehat{S}^m) : k \in M\}$.

Proof. Let $(\widehat{S}^1,\ldots,\widehat{S}^m)$ be an optimal solution to problem (2.1) with the largest cardinality, so that if $(\widetilde{S}^1,\ldots,\widetilde{S}^m)$ is another optimal solution, then $|\cup_{k\in M}\widehat{S}^k| \geq |\cup_{k\in M}\widetilde{S}^k|$. By (A.1), we have

$$R^\nu(S^\nu,\ldots,S^m) = \frac{\sum_{i\in S^\nu} r_i\,v_i^\nu}{1+\sum_{i\in S^\nu} v_i^\nu} + \frac{1}{1+\sum_{i\in S^\nu} v_i^\nu}\sum_{k=\nu+1}^{m}\left\{\prod_{\ell=\nu+1}^{k}\frac{1}{1+\sum_{i\in S^\ell} v_i^\ell}\right\}\sum_{i\in S^k} r_i\,v_i^k$$
$$= \frac{\sum_{i\in S^\nu} r_i\,v_i^\nu + R^{\nu+1}(S^{\nu+1},\ldots,S^m)}{1+\sum_{i\in S^\nu} v_i^\nu}. \tag{A.2}$$

First, we show that $\cup_{k\in M}\widehat{S}^k \supseteq \{i\in N : r_i \geq \widehat{\zeta}\}$. To get a contradiction assume that there exists a product j such that $j\in\{i\in N : r_i\geq\widehat{\zeta}\}$ and $j\notin\cup_{k\in M}\widehat{S}^k$. For notational brevity, we let $\widehat{R}^k = R^k(\widehat{S}^k,\ldots,\widehat{S}^m)$. Since $j\in\{i\in N : r_i\geq\widehat{\zeta}\}$ and $\widehat{\zeta} = \min\{\widehat{R}^k : k\in M\}$, we have $r_j\geq\widehat{R}^\ell$ for some $\ell\in M$. Furthermore, since $j\notin\cup_{k\in M}\widehat{S}^k$, we have $j\notin\widehat{S}^\ell$. We define a solution $(\widetilde{S}^1,\ldots,\widetilde{S}^m)$ to problem (2.1) as $\widetilde{S}^k=\widehat{S}^k$ for all $k\in M\setminus\{\ell\}$ and $\widetilde{S}^\ell=\widehat{S}^\ell\cup\{j\}$. Since $j\notin\cup_{k\in M}\widehat{S}^k$, the product j is not offered in any stage in the solution $(\widehat{S}^1,\ldots,\widehat{S}^m)$. Therefore, the product j is only offered in stage ℓ in the solution $(\widetilde{S}^1,\ldots,\widetilde{S}^m)$, so $(\widetilde{S}^1,\ldots,\widetilde{S}^m)\in\mathcal{F}$. Letting $\widetilde{R}^k = R(\widetilde{S}^k,\ldots,\widetilde{S}^m)$ for notational brevity, we observe that $\widehat{R}^k=\widetilde{R}^k$ for all $k=\ell+1,\ldots,m$, since $R^k(S^k,\ldots,S^m)$ depends on $S^k,\ldots,S^m$ and $\widehat{S}^k=\widetilde{S}^k$ for all $k=\ell+1,\ldots,m$. In this case, we have

$$\widetilde{R}^\ell-\widehat{R}^\ell = \frac{\sum_{i\in\widetilde{S}^\ell} r_i\,v_i^\ell+\widetilde{R}^{\ell+1}}{1+\sum_{i\in\widetilde{S}^\ell} v_i^\ell}-\widehat{R}^\ell = \frac{\sum_{i\in\widehat{S}^\ell} r_i\,v_i^\ell + r_j\,v_j^\ell+\widehat{R}^{\ell+1}}{1+\sum_{i\in\widehat{S}^\ell} v_i^\ell+v_j^\ell}-\widehat{R}^\ell$$
$$= \frac{\widehat{R}^\ell\,(1+\sum_{i\in\widehat{S}^\ell} v_i^\ell)+r_j\,v_j^\ell}{1+\sum_{i\in\widehat{S}^\ell} v_i^\ell+v_j^\ell}-\widehat{R}^\ell = \frac{(r_j-\widehat{R}^\ell)\,v_j^\ell}{1+\sum_{i\in\widehat{S}^\ell} v_i^\ell+v_j^\ell}\geq 0,$$

where the second equality uses the fact that $\widetilde{S}^\ell=\widehat{S}^\ell\cup\{j\}$ and $\widetilde{R}^{\ell+1}=\widehat{R}^{\ell+1}$, the third equality uses the fact $\widehat{R}^\ell\,(1+\sum_{i\in\widehat{S}^\ell} v_i^\ell) = \sum_{i\in\widehat{S}^\ell} r_i\,v_i^\ell+\widehat{R}^{\ell+1}$ by (A.2) and the inequality follows from the fact that $r_j\geq\widehat{R}^\ell$. Therefore, we obtain $\widetilde{R}^\ell\geq\widehat{R}^\ell$. By (A.2), for all $k=1,\ldots,\ell-1$, we have $\widetilde{R}^k = (\sum_{i\in\widetilde{S}^k} r_i\,v_i^k+\widetilde{R}^{k+1})/(1+\sum_{i\in\widetilde{S}^k} v_i^k) = (\sum_{i\in\widehat{S}^k} r_i\,v_i^k+\widetilde{R}^{k+1})/(1+\sum_{i\in\widehat{S}^k} v_i^k)$, where we use the fact that $\widetilde{S}^k=\widehat{S}^k$. Similarly, we have $\widehat{R}^k = (\sum_{i\in\widehat{S}^k} r_i\,v_i^k+\widehat{R}^{k+1})/(1+\sum_{i\in\widehat{S}^k} v_i^k)$ for all $k=1,\ldots,\ell-1$. Subtracting the two equalities, we obtain $\widetilde{R}^k - \widehat{R}^k = (\widetilde{R}^{k+1} - \widehat{R}^{k+1})/(1 + \sum_{i \in \widehat{S}^k} v_j^k)$ for all $k = 1, \ldots, \ell - 1$. In this case, having $\widetilde{R}^\ell \geq \widehat{R}^\ell$ implies that $\widetilde{R}^1 \geq \widehat{R}^1$. Therefore, the objective value provided by the solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$ for problem (2.1) is at least as large as the one provided by the solution $(\widehat{S}^1, \ldots, \widehat{S}^m)$. Furthermore, $\cup_{k \in M} \widetilde{S}^k = \cup_{k \in M} \widehat{S}^k \cup \{j\}$, which contradicts the fact that $(\widehat{S}^1, \ldots, \widehat{S}^m)$ is an optimal solution to problem (2.1) with the largest cardinality.

Second, we show that $\cup_{k \in M} \widehat{S}^k \subseteq \{i \in N : r_i \geq \widehat{\zeta}\}$. To get a contradiction, assume that there exists a product j such that $j \in \cup_{k \in M} \widehat{S}^k$ and $j \notin \{i \in N : r_i \geq \widehat{\zeta}\}$. Since $j \in \cup_{k \in M} \widehat{S}^k$, we have $j \in \widehat{S}^\ell$ for some $\ell \in M$. Also, noting that $j \notin \{i \in N : r_i \geq \widehat{\zeta}\}$, we have $r_j < \widehat{\zeta} = \min\{\widehat{R}^k : k \in M\}$, which implies that $r_j < \widehat{R}^\ell$. We define a solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$ to problem (2.1) as $\widetilde{S}^k = \widehat{S}^k$ for all $k \in M \setminus \{\ell\}$ and $\widetilde{S}^\ell = \widehat{S}^\ell \setminus \{j\}$. Since $(\widehat{S}^1, \ldots, \widehat{S}^m) \in \mathcal{F}$ and $\widetilde{S}^k \subseteq \widehat{S}^k$ for all $k \in M$, each product is offered in at most one stage in the solution $(\widetilde{S}^1, \ldots, \widetilde{S}^m)$, so $(\widetilde{S}^1, \ldots, \widetilde{S}^m) \in \mathcal{F}$. Using the same argument in the previous paragraph and noting that $r_j < \widehat{R}^\ell$, we can show that $\widetilde{R}^1 > \widehat{R}^1$, contradicting the fact that $(\widehat{S}^1, \ldots, \widehat{S}^m)$ is an optimal solution to problem (2.1). ■

The proposition above indicates that there exists an optimal solution to problem (2.1) that has a structure similar to that of an optimal solution when there is a single stage in the choice process. This structure is adequate to obtain an optimal solution efficiently when there is a single stage, but it is not adequate even when there are as few as two stages.

APPENDIX B

APPENDIX FOR CHAPTER 3

B.1 Change in Expected Revenue with Product Exchanges

We give a proof for Lemma 3.3.2. We show only the first identity, as the proofs for the second and third identities are similar. In the expected revenue in (3.3), we have one term for each stage. Considering $\Pi(S_1, \ldots, S_{k-1} \cup \{i\}, S_k \setminus \{i\}, \ldots, S_m)$ and $\Pi(S_1, \ldots, S_m)$, except for the terms corresponding to stages $k-1$ and k, the terms for other stages are identical. So, for the sequence of assortments $(S_1, \ldots, S_m)$, letting $\widehat{W}_k = W(S_k)$, $\widehat{\theta}_k = \sum_{\ell=1}^{k} V(S_\ell)$ and $\widehat{R}_k = R_k(S_1, \ldots, S_m)$, we have

$$
\begin{aligned}
&\Pi(S_1, \ldots, S_{k-1} \cup \{i\}, S_k \setminus \{i\}, \ldots, S_m) - \Pi(S_1, \ldots, S_m) \\
&\quad = \frac{\lambda_{k-1}\,(\widehat{W}_{k-1} + v_i\, r_i)}{(1+\widehat{\theta}_{k-2})\,(1+\widehat{\theta}_{k-1}+v_i)} + \frac{\lambda_k\,(\widehat{W}_k - v_i\, r_i)}{(1+\widehat{\theta}_{k-1}+v_i)\,(1+\widehat{\theta}_k)} \\
&\qquad\qquad - \frac{\lambda_{k-1}\,\widehat{W}_{k-1}}{(1+\widehat{\theta}_{k-2})\,(1+\widehat{\theta}_{k-1})} - \frac{\lambda_k\,\widehat{W}_k}{(1+\widehat{\theta}_{k-1})\,(1+\widehat{\theta}_k)} \\
&\quad = \frac{v_i\, r_i}{1+\widehat{\theta}_{k-1}+v_i}\left(\frac{\lambda_{k-1}}{1+\widehat{\theta}_{k-2}} - \frac{\lambda_k}{1+\widehat{\theta}_k}\right) + \frac{\lambda_{k-1}\,\widehat{W}_{k-1}}{1+\widehat{\theta}_{k-2}}\left(\frac{1}{1+\widehat{\theta}_{k-1}+v_i} - \frac{1}{1+\widehat{\theta}_{k-1}}\right) \\
&\qquad\qquad + \frac{\lambda_k\,\widehat{W}_k}{1+\widehat{\theta}_k}\left(\frac{1}{1+\widehat{\theta}_{k-1}+v_i} - \frac{1}{1+\widehat{\theta}_{k-1}}\right) \\
&\quad = \frac{v_i\, r_i}{1+\widehat{\theta}_{k-1}+v_i}\left(\frac{\lambda_{k-1}}{1+\widehat{\theta}_{k-2}} - \frac{\lambda_k}{1+\widehat{\theta}_k}\right) - \left(\frac{\lambda_{k-1}\,\widehat{W}_{k-1}}{1+\widehat{\theta}_{k-2}} + \frac{\lambda_k\,\widehat{W}_k}{1+\widehat{\theta}_k}\right)\left(\frac{v_i}{(1+\widehat{\theta}_{k-1}+v_i)\,(1+\widehat{\theta}_{k-1})}\right) \\
&\quad = \frac{v_i\, r_i}{1+\widehat{\theta}_{k-1}+v_i}\left(\frac{\lambda_{k-1}}{1+\widehat{\theta}_{k-2}} - \frac{\lambda_k}{1+\widehat{\theta}_k}\right) - \frac{v_i}{1+\widehat{\theta}_{k-1}+v_i}\,(\widehat{R}_{k-1} + \widehat{R}_k) \\
&\quad = \frac{\frac{\lambda_{k-1}}{1+\widehat{\theta}_{k-2}} - \frac{\lambda_k}{1+\widehat{\theta}_k}}{1+\widehat{\theta}_{k-1}+v_i}\, v_i \left(r_i - \frac{\widehat{R}_{k-1} + \widehat{R}_k}{\frac{\lambda_{k-1}}{1+\widehat{\theta}_{k-2}} - \frac{\lambda_k}{1+\widehat{\theta}_k}}\right) = \frac{\frac{\lambda_{k-1}}{1+\widehat{\theta}_{k-2}} - \frac{\lambda_k}{1+\widehat{\theta}_k}}{1+\widehat{\theta}_{k-1}+v_i}\, v_i\,(r_i - t_k(S_1, \ldots, S_m)).
\end{aligned}
$$

B.2 Structure of the Optimal Solution under Space Constraints

We use the following result when constructing our FPTAS.

Lemma B.2.1 *In a non-dominated optimal solution* $(S_1^*, \ldots, S_m^*)$ *to the* CAPACITATED *problem, for all* $k \in \mathcal{M}$*, we have* $S_k^* \subseteq \{j_k^* + 1, \ldots, j_{k+1}^*\}$ *for* $j_1^*, \ldots, j_{m+1}^*$ *with* $0 = j_1^* \leq j_2^* \leq \ldots \leq j_m^* \leq j_{m+1}^* = n$.

Proof. For $i \in S_k^*$ and $j \in S_{k+1}^*$, we must have $r_i \geq t_{k+1}(S_1^*, \ldots, S_m^*) > r_j$. In particular, by the first part of Lemma 3.3.2, if $r_j \geq t_{k+1}(S_1^*, \ldots, S_m^*)$, then we can move product j from stage $k+1$ to stage k without degrading the expected revenue provided by the solution $(S_1^*, \ldots, S_m^*)$, which contradicts the fact that $(S_1^*, \ldots, S_m^*)$ is non-dominated. By the second part of Lemma 3.3.2, if $r_i < t_{k+1}(S_1^*, \ldots, S_m^*)$, then we can move product i from stage k to stage $k+1$ to obtain a solution strictly better than the solution $(S_1^*, \ldots, S_m^*)$, which contradicts the fact that $(S_1^*, \ldots, S_m^*)$ is an optimal solution. Thus, if $r_i \in S_k^*$ and $j \in S_{k+1}^*$, then we must have $r_i > r_j$. ■

B.3 Constructing Candidate Assortments

In this section, we give a proof for Lemma 3.5.2. In the next lemma, we show the monotonicity for the value functions in (3.10). In the proof, we use the fact that $\lfloor a \rfloor$ and $\lceil a \rceil$ are increasing in a. Note that this property holds even with the convention that $\lfloor a \rfloor = 0$ and $\lceil a \rceil = -\infty$ if $a < 0$.

Lemma B.3.1 *If the value functions* $\{\Theta_i^\ell(x,y) : (x,y) \in \text{Dom}^2,\ i = 1,\ldots,\ell+1\}$ *are computed through the dynamic program in* (3.10)*, then* $\Theta_i^\ell(x,y)$ *is increasing in* x *and decreasing in* y*.*

Proof: We show the result by using induction over the decision epochs. By the boundary condition, $\Theta_{\ell+1}^\ell(x,y)$ is increasing in x and decreasing in y. Assuming that $\Theta_{i+1}^\ell(x,y)$ is increasing in x and decreasing in y, we proceed to showing that $\Theta_i^\ell(x,y)$ is increasing in x and decreasing in y. Since $\lfloor a \rfloor$ and $\lceil a \rceil$ are increasing in a, $\lfloor x - v_i\, r_i\, u_i \rfloor$ and $\lceil y - v_i\, u_i \rceil$ are increasing in x and y, in which case, by the induction hypothesis, $\Theta_{i+1}^\ell(\lfloor x - v_i\, r_i\, u_i \rfloor, \lceil y - v_i\, u_i \rceil)$ is increasing in x and decreasing in y. Thus, for a fixed value of u_i, the objective function of the minimization problem in (3.10) is increasing in x and decreasing in y. So, the optimal objective value of this minimization problem, which is equal to $\Theta_i^\ell(x,y)$, must be increasing in x and decreasing in y as well. ∎

Next, we give a proof for Lemma 3.5.2.

Proof of Lemma 3.5.2: Throughout the proof, let $S \subseteq \{j+1,\ldots,\ell\}$ be such that $W(S) \ge x$ and $V(S) \le y$. Our proof proceeds in three parts.

Part 1: First, assuming that such an assortment S exists, we show that $\Theta_{j+1}^\ell(x,y) < +\infty$. For notational brevity, we let $S^i = S \cap \{i,\ldots,\ell\}$. Also, we define $\widetilde{u}_i \in \{0,1\}$ as $\widetilde{u}_i = 1$ if and only if $i \in S$. Since $\widetilde{u}_i$ is a feasible but not necessarily an optimal solution to the minimization problem on the right side of (3.10) with $x = W(S^i)$ and $y = V(S^i)$, we have

$$\begin{aligned}\Theta_i^\ell(W(S^i), V(S^i)) &\le c_i\,\widetilde{u}_i + \Theta_{i+1}^\ell(\lfloor W(S^i) - v_i\, r_i\, \widetilde{u}_i \rfloor, \lceil V(S^i) - v_i\, \widetilde{u}_i \rceil)\\ &= c_i\,\widetilde{u}_i + \Theta_{i+1}^\ell(\lfloor W(S^{i+1}) \rfloor, \lceil V(S^{i+1}) \rceil) \le c_i\,\widetilde{u}_i + \Theta_{i+1}^\ell(W(S^{i+1}), V(S^{i+1})),\end{aligned}$$

where the last inequality uses Lemma B.3.1. Noting that $S \subseteq \{j+1,\ldots,\ell\}$, $S^{j+1} =$

S and $S^{\ell+1} = \emptyset$. Thus, adding the chain of inequalities above over all $i = j+1,\ldots,\ell$, we obtain $\Theta^{\ell}_{j+1}(W(S),V(S)) \leq \sum_{i=j+1}^{\ell} c_i\,\widetilde{u}_i + \Theta^{\ell}_{\ell+1}(W(\emptyset),V(\emptyset)) = C(S)$, where the equality uses the fact that $\Theta^{\ell}_{\ell+1}(0,0) = 0$. Since $W(S) \geq x$, $V(S) \leq y$, by Lemma B.3.1, we get $\Theta^{\ell}_{j+1}(x,y) \leq \Theta^{\ell}_{j+1}(W(S),V(S)) \leq C(S) < +\infty$.

Part 2: Second, we show that $C(\widehat{S}_{x,y}) \leq C(S)$. Noting the last chain of inequalities at the end of the previous paragraph, it is enough to show that $\Theta^{\ell}_{j+1}(x,y) = C(\widehat{S}_{x,y})$. Consider executing the candidate construction algorithm with $(x,y) \in \text{Dom}^2$. Noting Steps 2 and 3 in the candidate construction algorithm, along with the dynamic program in (3.10), we have $\Theta^{\ell}_{i}(\widehat{x}_i,\widehat{y}_i) = c_i\,\widehat{u}_i + \Theta^{\ell}_{i+1}(\widehat{x}_{i+1},\widehat{y}_{i+1})$. Adding this equality over all $i = j+1,\ldots,\ell$, we get $\Theta^{\ell}_{j+1}(\widehat{x}_{j+1},\widehat{y}_{j+1}) = C(\widehat{S}_{x,y}) + \Theta^{\ell}_{\ell+1}(\widehat{x}_{\ell+1},\widehat{y}_{\ell+1})$. Since we start the candidate construction algorithm with $\widehat{x}_{j+1} = x$ and $\widehat{y}_{j+1} = y$, the last equality yields $\Theta^{\ell}_{j+1}(x,y) = C(\widehat{S}_{x,y}) + \Theta^{\ell}_{\ell+1}(\widehat{x}_{\ell+1},\widehat{y}_{\ell+1})$. From the discussion in the previous paragraph, $\Theta^{\ell}_{j+1}(x,y) < +\infty$. Also, $\Theta^{\ell}_{\ell+1}(\widehat{x}_{\ell+1},\widehat{y}_{\ell+1})$ can only take the value $+\infty$ or zero. If $\Theta^{\ell}_{\ell+1}(\widehat{x}_{\ell+1},\widehat{y}_{\ell+1}) = +\infty$, then we get a contradiction to the fact that $\Theta^{\ell}_{j+1}(x,y) < +\infty$ and $\Theta^{\ell}_{j+1}(x,y) = C(\widehat{S}_{x,y}) + \Theta^{\ell}_{\ell+1}(\widehat{x}_{\ell+1},\widehat{y}_{\ell+1})$. Thus, we must have $\Theta^{\ell}_{\ell+1}(\widehat{x}_{\ell+1},\widehat{y}_{\ell+1}) = 0$. In this case, having $\Theta^{\ell}_{j+1}(x,y) = C(\widehat{S}_{x,y}) + \Theta^{\ell}_{\ell+1}(\widehat{x}_{\ell+1},\widehat{y}_{\ell+1})$ implies that $\Theta^{\ell}_{j+1}(x,y) = C(\widehat{S}_{x,y})$.

Part 3: Third, we show that $W(\widehat{S}_{x,y}) \geq x/(1+\rho)^n$. Letting $\widehat{S}^i_{x,y} = \widehat{S}_{x,y} \cap \{i,\ldots,\ell\}$, we use induction over the decision epochs to show that $W(\widehat{S}^i_{x,y}) \geq \widehat{x}_i/(1+\rho)^{\ell+1-i}$, where $\widehat{x}_i$ is as in the candidate construction algorithm. By the discussion in the previous paragraph, we have $\Theta^{\ell}_{\ell+1}(\widehat{x}_{\ell+1},\widehat{y}_{\ell+1}) = 0$, in which case, by the boundary condition of the dynamic program in (3.10), we must have $\widehat{x}_{\ell+1} \leq 0$. Also, $\widehat{S}^{\ell+1}_{x,y} = \emptyset$. Therefore, we get $W(\widehat{S}^{\ell+1}_{x,y}) = 0 \geq \widehat{x}_{\ell+1}$, so the result holds for decision epoch $\ell+1$. Assuming that $W(\widehat{S}^{i+1}_{x,y}) \geq \widehat{x}_{i+1}/(1+\rho)^{\ell-i}$, we proceed to showing that $W(\widehat{S}^i_{x,y}) \geq \widehat{x}_i/(1+\rho)^{\ell+1-i}$. Since $\widehat{x}_{i+1} = \lfloor \widehat{x}_i - v_i\,r_i\,\widehat{u}_i \rfloor$, we have $\widehat{x}_{i+1} \geq \frac{1}{1+\rho}\,(\widehat{x}_i - v_i\,r_i\,\widehat{u}_i)$. Noting that

$\lfloor a \rfloor = 0$ for $a < 0$, the last inequality holds when $\widehat{x}_i - v_i\, r_i\, \widehat{u}_i < 0$ as well. The last inequality yields $(1+\rho)\,\widehat{x}_{i+1} + v_i\, r_i\, \widehat{u}_i \geq \widehat{x}_i$. Since $\widehat{S}^i_{x,y} \setminus \{i\} = \widehat{S}^{i+1}_{x,y}$ and $\widehat{u}_i = 1$ if and only if $i \in \widehat{S}^i_{x,y}$, we get

$$
\begin{aligned}
W(\widehat{S}^i_{x,y}) \;=\; W(\widehat{S}^{i+1}_{x,y}) + v_i\, r_i\, \widehat{u}_i \;&\geq\; \frac{\widehat{x}_{i+1}}{(1+\rho)^{\ell-i}} + v_i\, r_i\, \widehat{u}_i \\
&\geq\; \frac{1}{(1+\rho)^{\ell+1-i}} \Big\{ (1+\rho)\,\widehat{x}_{i+1} + v_i\, r_i\, \widehat{u}_i \Big\} \;\geq\; \frac{\widehat{x}_i}{(1+\rho)^{\ell+1-i}},
\end{aligned}
$$

where the first inequality uses the induction hypothesis. Thus, the induction argument is complete. Since $\widehat{S}^{j+1}_{x,y} = \widehat{S}_{x,y}$ and $\widehat{x}_{j+1} = x$, we get $W(\widehat{S}_{x,y}) = W(\widehat{S}^{j+1}_{x,y}) \geq \widehat{x}_{j+1}/(1+\rho)^{\ell-j} \geq x/(1+\rho)^n$.

Lastly, we can follow an argument that is similar to the one in the previous paragraph to also show that $V(\widehat{S}_{x,y}) \leq (1+\rho)^n\, y$. ■

B.4 Combining Candidate Assorments

In this section, we give a proof for Lemma 3.5.3. In the next lemma, we give two monotonicity properties of the value functions $\{\Psi_k(\ell, u, z) : \ell = 0, \ldots, n,\ (u,z) \in \textsc{Dom}^2,\ k \in \mathcal{M}\}$ computed through the dynamic program in (3.12). Intuitively speaking, the second one of these properties states that we can compensate for an increase by a factor of $(1+\rho)^2$ in the state variable z by an increase by a factor of $1+\rho$ in the state variable u. This result becomes critical in ultimately proving the performance guarantee of our FPTAS.

Lemma B.4.1 *If the value functions* $\{\Psi_k(j, u, z) : j = 0, \ldots, n,\ (u,z) \in \textsc{Dom}^2,\ k \in \mathcal{M}\}$ *are computed through the dynamic program in* (3.12)*, then* $\Psi_k(j,u,z)$ *is increasing in* j*,* u *and* z*. Furthermore, we have* $\Psi_k(j, (1+\rho)\,u, z) \leq \Psi_k(j, u, (1+\rho)^2\, z)$.

Proof: The fact that $\Psi_k(j,u,z)$ is increasing in j, u and z follows from an induction argument that is similar to the one in the proof of Lemma B.3.1. To show that $\Psi_k(j,(1+\rho)\,u,z) \le \Psi_k(j,u,(1+\rho)^2\,z)$, we use induction over the decision epochs. Since $\Psi_{m+1}(j,(1+\rho)\,u,z)$ depends only on the sign of z and the signs of z and $(1+\rho)^2\,z$ are the same, we have $\Psi_{m+1}(j,(1+\rho)\,u,z) = \Psi_{m+1}(j,u,(1+\rho)^2\,z)$. Assuming that $\Psi_{k+1}(j,(1+\rho)\,u,z) \le \Psi_{k+1}(j,u,(1+\rho)^2\,z)$, we proceed to showing that $\Psi_k(j,(1+\rho)\,u,z) \le \Psi_k(j,u,(1+\rho)^2\,z)$. We have $(1+\rho)\,u + V(S) \le (1+\rho)\lceil u + V(S)\rceil$. Since $(1+\rho)\lceil u+V(S)\rceil \in$ DOM, the last inequality implies that $\lceil (1+\rho)\,u + V(S)\rceil \le (1+\rho)\lceil u + V(S)\rceil$. In this case, we have

$$
\begin{aligned}
&C(S) + \Psi_{k+1}\Big(\ell\,,\, \lceil (1+\rho)\,u + V(S)\rceil\,,\, \Big\lceil z - \frac{\lambda_k\,W(S)}{(1+(1+\rho)\,u)\,(1+(1+\rho)\,u+V(S))}\Big\rceil\Big) \\
&\quad \overset{(a)}{\le} C(S) + \Psi_{k+1}\Big(\ell\,,\, (1+\rho)\,\lceil u + V(S)\rceil\,,\, \Big\lceil z - \frac{\lambda_k\,W(S)}{(1+(1+\rho)\,u)\,(1+(1+\rho)\,u+V(S))}\Big\rceil\Big) \\
&\quad \overset{(b)}{\le} C(S) + \Psi_{k+1}\Big(\ell\,,\, \lceil u + V(S)\rceil\,,\, (1+\rho)^2\Big\lceil z - \frac{\lambda_k\,W(S)}{(1+(1+\rho)\,u)\,(1+(1+\rho)\,u+V(S))}\Big\rceil\Big),
\end{aligned}
\tag{B.1}
$$

where (a) follows from the fact that $\Psi_k(\ell,u,z)$ is increasing in u and (b) follows from the induction argument.

Note that $(1+\rho)^2\lceil a\rceil \le \lceil (1+\rho)^2\,a\rceil$. If $a < 0$, then the inequality is trivial. For $a \ge 0$, $a \le \frac{1}{(1+\rho)^2}\lceil (1+\rho)^2 a\rceil$. Since $\frac{\lceil (1+\rho)^2 a\rceil}{(1+\rho)^2} \in$ DOM, the last inequality yields $\lceil a\rceil \le \frac{1}{(1+\rho)^2}\lceil (1+\rho)^2 a\rceil$. So,

$$
\begin{aligned}
(1+\rho)^2 &\Big\lceil z - \frac{\lambda_k\,W(S)}{(1+(1+\rho)u)\,(1+(1+\rho)u+V(S))}\Big\rceil \\
&\le \Big\lceil (1+\rho)^2 z - \frac{(1+\rho)^2\,\lambda_k\,W(S)}{(1+(1+\rho)u)\,(1+(1+\rho)u+V(S))}\Big\rceil \\
&\le \Big\lceil (1+\rho)^2 z - \frac{\lambda_k\,W(S)}{(1+u)\,(1+u+V(S))}\Big\rceil,
\end{aligned}
$$

where the second inequality uses the fact that $\lceil a\rceil$ is increasing in a, even with the convention that $\lceil a\rceil = -\infty$ for $a < 0$, as discussed right before the proof of Lemma B.3.1.

Using the chain of inequalities above and the fact that $\Psi_{k+1}(j, u, z)$ is increasing in z, we can bound the expression on the right side of (B.1) as

$$C(S) + \Psi_{k+1}\Big(\ell\,,\, \lceil u + V(S)\rceil\,, (1+\rho)^2\Big\lceil z - \frac{\lambda_k\, W(S)}{(1+(1+\rho)\,u)\,(1+(1+\rho)\,u + V(S))}\Big\rceil\Big)$$
$$\leq C(S) + \Psi_{k+1}\Big(\ell\,,\, \lceil u + V(S)\rceil\,, \Big\lceil (1+\rho)^2 z - \frac{\lambda_k\, W(S)}{(1+u)\,(1+u+V(S))}\Big\rceil\Big). \quad \text{(B.2)}$$

By (B.1) and (B.2), we have $C(S) + \Psi_{k+1}(\ell\,, \lceil(1+\rho)\,u + V(S)\rceil\,, \lceil z - \frac{\lambda_k\, W(S)}{(1+(1+\rho)\,u)\,(1+(1+\rho)\,u+V(S))}\rceil) \leq C(S) + \Psi_{k+1}(\ell\,, \lceil u + V(S)\rceil\,, \lceil(1+\rho)^2 z - \frac{\lambda_k\, W(S)}{(1+u)\,(1+u+V(S))}\rceil)$ for all S and ℓ. In this case, minimizing both sides of the inequality over (ℓ, S) with $\ell \geq j$ and $S \in \textsc{Cand}(j, \ell)$, the inequality is still preserved, but noting (3.12), the left side of the inequality gives $\Psi_k(j, (1+\rho)\,u, z)$, whereas the right side gives $\Psi_k(j, u, (1+\rho)^2\, z)$. Thus, we have $\Psi_k(\ell, (1+\rho)\,u, z) \leq \Psi_k(\ell, u, (1+\rho)^2\, z)$. ■

Next, we give a proof for Lemma 3.5.3.

Proof of Lemma 3.5.3: Let $(\widehat{S}_1, \ldots, \widehat{S}_m)$ be the output of the candidate stitching algorithm, $\widetilde{z}$ be the optimal objective value of problem (3.9) and $\widehat{z}_{\text{APP}} = \max\{z \in \textsc{Dom} : \Psi_1(0, 0, z) \leq b\}$. Our proof proceeds in three parts.

Part 1: First, we show that $\sum_{k\in\mathcal{M}} C(\widehat{S}_k) \leq b$. Noting Steps 2 and 3 in the candidate stitching algorithm, along with the dynamic program in (3.12), we have $\Psi_k(\widehat{j}_k, \widehat{u}_k, \widehat{z}_k) = C(\widehat{S}_k) + \Psi_{k+1}(\widehat{j}_{k+1}, \widehat{u}_{k+1}, \widehat{z}_{k+1})$. Adding this equality over all $k \in \mathcal{M}$ and noting that we start the candidate stitching algorithm with $\widehat{j}_1 = 0, \widehat{u}_1 = 0$ and $\widehat{z}_1 = \widehat{z}_{\text{APP}}$, we obtain $\Psi_1(0, 0, \widehat{z}_{\text{APP}}) = \sum_{k\in\mathcal{M}} C(\widehat{S}_k) + \Psi_{m+1}(\widehat{j}_{m+1}, \widehat{u}_{m+1}, \widehat{z}_{m+1})$. By Step 1 of the candidate stitching algorithm, we have $\Psi_1(0, 0, \widehat{z}_{\text{APP}}) \leq b$, in which case, the last equality implies that $\sum_{k\in\mathcal{M}} C(\widehat{S}_k) + \Psi_{m+1}(\widehat{j}_{m+1}, \widehat{u}_{m+1}, \widehat{z}_{m+1}) \leq b$. By the boundary condition of the dynamic program in (3.12), $\Psi_{m+1}(\widehat{j}_{m+1}, \widehat{u}_{m+1}, \widehat{z}_{m+1})$ takes the value $+\infty$ or zero. If we have $\Psi_{m+1}(\widehat{j}_{m+1}, \widehat{u}_{m+1}, \widehat{z}_{m+1}) = +\infty$, then we get a contradiction to the fact that $\sum_{k\in\mathcal{M}} C(\widehat{S}_k) + \Psi_{m+1}(\widehat{j}_{m+1}, \widehat{u}_{m+1}, \widehat{z}_{m+1}) \leq$

b. Thus, we must have $\Psi_{m+1}(\widehat{j}_{m+1}, \widehat{u}_{m+1}, \widehat{z}_{m+1}) = 0$, so having $\sum_{k \in \mathcal{M}} C(\widehat{S}_k) + \Psi_{m+1}(\widehat{j}_{m+1}, \widehat{u}_{m+1}, \widehat{z}_{m+1}) \le b$ implies that $\sum_{k \in \mathcal{M}} C(\widehat{S}_k) \le b$.

Part 2: Second, we show that $\textsc{Rev}(\widehat{S}_1, \ldots, \widehat{S}_m) \ge \widehat{z}_{\text{APP}}$. By Step 3 of the candidate stitching algorithm, we have $\widehat{u}_{k+1} \ge \widehat{u}_k + V(\widehat{S}_k)$. Adding this inequality over all $k = 1, \ldots, q-1$ and noting that $\widehat{u}_1 = 0$ in Step 1 of the algorithm, we get $\widehat{u}_q \ge \sum_{k=1}^{q-1} V(\widehat{S}_k)$. For notational brevity, we let $\widehat{R}_k = \sum_{q=k}^m \frac{\lambda_q \, W(\widehat{S}_q)}{(1+\sum_{r=1}^{q-1} V(\widehat{S}_r))\,(1+\sum_{r=1}^{q} V(\widehat{S}_r))}$ with the convention that $\widehat{R}_{m+1} = 0$. We use induction over the stages to show that $\widehat{R}_k \ge \widehat{z}_k$ for all $k = 1, \ldots, m+1$. By the discussion in the previous paragraph, $\Psi_{m+1}(\widehat{j}_{m+1}, \widehat{u}_{m+1}, \widehat{z}_{m+1}) = 0$, in which case, by the boundary condition in (3.12), we must have $\widehat{z}_{m+1} \le 0$. Thus, we have $\widehat{R}_{m+1} = 0 \ge \widehat{z}_{m+1}$. Assuming that $\widehat{R}_{k+1} \ge \widehat{z}_{k+1}$, we proceed to showing that $\widehat{R}_k \ge \widehat{z}_k$. Noting Step 3 of the candidate stitching algorithm and using the induction hypothesis, if $\widehat{z}_{k+1} \ge 0$, then $\widehat{z}_k - \frac{\lambda_k \, W(\widehat{S}_k)}{(1+\widehat{u}_k)\,(1+\widehat{u}_k+V(\widehat{S}_k))} \le \widehat{z}_{k+1} \le \widehat{R}_{k+1}$. Also, if $\widehat{z}_{k+1} < 0$, then $\widehat{z}_k - \frac{\lambda_k \, W(\widehat{S}_k)}{(1+\widehat{u}_k)\,(1+\widehat{u}_k+V(\widehat{S}_k))} < 0 \le \widehat{R}_{k+1}$. So, $\widehat{z}_k \le \widehat{R}_{k+1} + \frac{\lambda_k \, W(\widehat{S}_k)}{(1+\widehat{u}_k)\,(1+\widehat{u}_k+V(\widehat{S}_k))}$ in both cases. Thus, we get

$$\begin{aligned} \widehat{R}_k &= \widehat{R}_{k+1} + \frac{\lambda_k \, W(\widehat{S}_k)}{(1 + \sum_{q=1}^{k-1} V(\widehat{S}_q))\,(1 + \sum_{q=1}^{k} V(\widehat{S}_q))} \\ &\ge \widehat{R}_{k+1} + \frac{\lambda_k \, W(\widehat{S}_k)}{(1 + \widehat{u}_k)\,(1 + \widehat{u}_k + V(\widehat{S}_k))} \ \ge \ \widehat{z}_k, \end{aligned}$$

where we use the fact that $\widehat{u}_k \ge \sum_{q=1}^{k-1} V(\widehat{S}_q)$. The induction argument is complete, in which case, we have $\widehat{R}_1 \ge \widehat{z}_1$. Noting that $\widehat{R}_1 = \textsc{Rev}(\widehat{S}_1, \ldots, \widehat{S}_m)$ and $\widehat{z}_1 = \widehat{z}_{\text{APP}}$, the result follows.

Part 3: Third, we show that $\widehat{z}_{\text{APP}} \ge \widetilde{z}/(1+\rho)^{3m+1}$. Let $(\widetilde{S}_1, \ldots, \widetilde{S}_m, \widetilde{j}_1, \ldots, \widetilde{j}_m)$ be an optimal solution to problem (3.9). For notational brevity, we let $\widetilde{C}_k = \sum_{q=k}^m C(\widetilde{S}_q)$, $\widetilde{u}_k = \sum_{q=1}^{k-1} V(\widetilde{S}_q)$ and $\widetilde{z}_k = \sum_{q=k}^m \frac{\lambda_q \, W(\widetilde{S}_q)}{(1+\widetilde{u}_q)(1+\widetilde{u}_{q+1})}$ with the convention that $\widetilde{C}_{m+1} = 0$, $\widetilde{u}_1 = 0$ and $\widetilde{z}_{m+1} = 0$. We use induction over the stages to show that $\Psi_k(\widetilde{j}_k, \widetilde{u}_k, \widetilde{z}_k/(1+\rho)^{3(m+1-k)}) \le \widetilde{C}_k$. We have $\widetilde{z}_{m+1} = 0$ and $\widetilde{C}_{m+1} = 0$, in which case, noting the boundary condition in (3.12), we have $\Psi_{m+1}(\widetilde{j}_{m+1}, \widetilde{u}_{m+1}, \widetilde{z}_{m+1}) = \Psi_{m+1}(\widetilde{j}_{m+1}, \widetilde{u}_{m+1}, 0) = 0 = \widetilde{C}_{m+1}$. Assuming that $\Psi_{k+1}(\widetilde{j}_{k+1}, \widetilde{u}_{k+1}, \widetilde{z}_{k+1}/(1+\rho)^{3(m-k)}) \leq \widetilde{C}_{k+1}$, we proceed to showing that $\Psi_k(\widetilde{j}_k, \widetilde{u}_k, \widetilde{z}_k/(1+\rho)^{3(m+1-k)}) \leq \widetilde{C}_k$. We have

$$
\begin{aligned}
&(1+\rho)^2 \left\lceil \frac{\widetilde{z}_k}{(1+\rho)^{3(m+1-k)}} - \frac{\lambda_k\, W(\widetilde{S}_k)}{(1+\widetilde{u}_k)\,(1+\widetilde{u}_{k+1})} \right\rceil \\
&\leq\; (1+\rho)^3 \left(\frac{\widetilde{z}_k}{(1+\rho)^{3(m+1-k)}} - \frac{\lambda_k\, W(\widetilde{S}_k)}{(1+\widetilde{u}_k)\,(1+\widetilde{u}_{k+1})} \right) \\
&\overset{(a)}{=}\; \frac{\widetilde{z}_{k+1}}{(1+\rho)^{3(m-k)}} + \frac{\lambda_k\, W(\widetilde{S}_k)}{(1+\rho)^{3(m-k)}\,(1+\widetilde{u}_k)\,(1+\widetilde{u}_{k+1})} - \frac{(1+\rho)^3\,\lambda_k\, W(\widetilde{S}_k)}{(1+\widetilde{u}_k)\,(1+\widetilde{u}_{k+1})} \\
&\overset{(b)}{\leq}\; \frac{\widetilde{z}_{k+1}}{(1+\rho)^{3(m-k)}},
\end{aligned} \tag{B.3}
$$

where (a) follows from the fact that $\widetilde{z}_k = \widetilde{z}_{k+1} + \frac{\lambda_k W(\widetilde{S}_k)}{(1+\widetilde{u}_k)(1+\widetilde{u}_{k+1})}$ by the definition of $\widetilde{z}_k$ and (b) holds because we have $k \leq m$.

In (3.12), the action $(\widetilde{j}_{k+1}, \widetilde{S}_k)$ is feasible when the state of the system at decision epoch k is $(\widetilde{j}_k, \widetilde{u}_k, \widetilde{z}_k/(1+\rho)^{3(m+1-k)})$. In particular, since $(\widetilde{S}_1, \ldots, \widetilde{S}_m, \widetilde{j}_1, \ldots, \widetilde{j}_m)$ is a feasible solution to problem (3.9), we have $\widetilde{j}_{k+1} \geq \widetilde{j}_k$ and $\widetilde{S}_k \in \textsc{Cand}(\widetilde{j}_k, \widetilde{j}_{k+1})$. Since, the action $(\widetilde{j}_{k+1}, \widetilde{S}_k)$ is feasible to the minimization problem in (3.12) with $(j, u, z) = (\widetilde{j}_k, \widetilde{u}_k, \widetilde{z}_k/(1+\rho)^{3(m+1-k)})$, we get

$$
\begin{aligned}
&\Psi_k\left(\widetilde{j}_k, \widetilde{u}_k, \frac{\widetilde{z}_k}{(1+\rho)^{3(m+1-k)}}\right) \\
&\quad\leq\; C(\widetilde{S}_k) + \Psi_{k+1}\left(\widetilde{j}_{k+1}\,,\, \lceil \widetilde{u}_k + V(\widetilde{S}_k) \rceil\,,\, \left\lceil \frac{\widetilde{z}_k}{(1+\rho)^{3(m+1-k)}} - \frac{\lambda_k\, W(\widetilde{S}_k)}{(1+\widetilde{u}_k)\,(1+\widetilde{u}_k + V(\widetilde{S}_k))} \right\rceil\right) \\
&\quad=\; C(\widetilde{S}_k) + \Psi_{k+1}\left(\widetilde{j}_{k+1}\,,\, \lceil \widetilde{u}_{k+1} \rceil\,,\, \left\lceil \frac{\widetilde{z}_k}{(1+\rho)^{3(m+1-k)}} - \frac{\lambda_k\, W(\widetilde{S}_k)}{(1+\widetilde{u}_k)\,(1+\widetilde{u}_{k+1})} \right\rceil\right) \\
&\quad\overset{(a)}{\leq}\; C(\widetilde{S}_k) + \Psi_{k+1}\left(\widetilde{j}_{k+1}\,,\, (1+\rho)\,\widetilde{u}_{k+1}\,,\, \left\lceil \frac{\widetilde{z}_k}{(1+\rho)^{3(m+1-k)}} - \frac{\lambda_k\, W(\widetilde{S}_k)}{(1+\widetilde{u}_k)\,(1+\widetilde{u}_{k+1})} \right\rceil\right) \\
&\quad\overset{(b)}{\leq}\; C(\widetilde{S}_k) + \Psi_{k+1}\left(\widetilde{j}_{k+1}\,, \widetilde{u}_{k+1}\,,\, (1+\rho)^2 \left\lceil \frac{\widetilde{z}_k}{(1+\rho)^{3(m+1-k)}} - \frac{\lambda_k\, W(\widetilde{S}_k)}{(1+\widetilde{u}_k)\,(1+\widetilde{u}_{k+1})} \right\rceil\right) \\
&\quad\overset{(c)}{\leq}\; C(\widetilde{S}_k) + \Psi_{k+1}\left(\widetilde{j}_{k+1}\,, \widetilde{u}_{k+1}\,,\, \frac{\widetilde{z}_{k+1}}{(1+\rho)^{3(m-k)}}\right) \\
&\quad\overset{(d)}{\leq}\; C(\widetilde{S}_k) + \widetilde{C}_{k+1} \;=\; \widetilde{C}_k,
\end{aligned}
$$

where (a) follows from the fact that $\Psi_k(\ell, u, z)$ is increasing in u and $(1+\rho)\, u \geq \lceil u \rceil$, (b) follows by the second part of Lemma B.4.1, (c) follows by noting the fact that $\Psi_k(j, u, z)$ is increasing in z and using the inequality in (B.3) and (d) is by the induction hypothesis. Thus, the induction argument is complete, so it follows that $\Psi_k(\widetilde{j}_k, \widetilde{u}_k, \widetilde{z}_k/(1+\rho)^{3(m+1-k)}) \leq \widetilde{C}_k$.

By the definition of $\widetilde{z}_k$ and $\widetilde{u}_k$, $\widetilde{z}_1 = \sum_{k\in\mathcal{M}} \frac{\lambda_k\, W(\widetilde{S}_k)}{(1+\widetilde{u}_k)(1+\widetilde{u}_{k+1})} = \sum_{k\in\mathcal{M}} \frac{\lambda_k\, W(\widetilde{S}_k)}{(1+\sum_{q=1}^{k-1} V(\widetilde{S}_q))(1+\sum_{q=1}^{k} V(\widetilde{S}_q))}$ $= \textsc{Rev}(\widetilde{S}_1, \ldots, \widetilde{S}_m) = \widetilde{z}$, where the last equality uses the fact that $(\widetilde{S}_1, \ldots, \widetilde{S}_m, \widetilde{j}_1, \ldots, \widetilde{j}_m)$ is an optimal solution to problem (3.9). Thus, using the inequality $\Psi_k(\widetilde{j}_k, \widetilde{u}_k, \widetilde{z}_k/(1+\rho)^{3(m+1-k)}) \leq \widetilde{C}_k$ with $k = 1$, we get $\Psi_1(\widetilde{j}_1, 0, \widetilde{z}/(1+\rho)^{3m}) \leq \widetilde{C}_1 \leq b$, where the last inequality uses the fact that $(\widetilde{S}_1, \ldots, \widetilde{S}_m, \widetilde{j}_1, \ldots, \widetilde{j}_m)$ is an optimal solution to problem (3.9) so that $\widetilde{C}_1 = \sum_{k\in\mathcal{M}} C(\widetilde{S}_k) \leq b$. Since $\Psi_k(j, u, z)$ is increasing in j and z by Lemma B.4.1, we obtain $\Psi_1(0, 0, \lfloor\widetilde{z}\rfloor/(1+\rho)^{3m}) \leq \Psi_1(\widetilde{j}_1, 0, \widetilde{z}/(1+\rho)^{3m}) \leq b$, which implies that $\lfloor\widetilde{z}\rfloor/(1+\rho)^{3m} \in \textsc{Dom}$ is a feasible solution to the problem $\widehat{z}_{\textsc{App}} = \max\{z \in \textsc{Dom} : \Psi_1(0, 0, z) \leq b\}$. Therefore, $\widehat{z}_{\textsc{App}} \geq \lfloor\widetilde{z}\rfloor/(1+\rho)^{3m} \geq \widetilde{z}/(1+\rho)^{3m+1}$. ■

B.5 Bound on the State Variable

We use the next lemma when accounting for the number of operations that is necessary to execute Part 2 in the development of our FPTAS.

Lemma B.5.1 *For $\rho \leq \frac{1}{2m}$, if we compute $\{\widehat{u}_k : k = 1, \ldots, m+1\}$ as $\widehat{u}_{k+1} = \lceil \widehat{u}_k + V(S_k) \rceil$ with $\widehat{u}_1 = 0$ and $S_k \cap S_q = \varnothing$ for all $k \neq q$, then $\widehat{u}_{m+1} \leq 2n\, v_{\max}$.*

Proof: We use induction to show that $\widehat{u}_k \leq (1+\rho)^{k-1}\, (V(S_1) + \ldots + V(S_{k-1}))$. For $k = 1$, we have $\widehat{u}_1 = 0$. Therefore, the result holds for $k = 1$. Assuming that

$\widehat{u}_k \leq (1+\rho)^{k-1}(V(S_1) + \ldots + V(S_{k-1}))$, we proceed to showing that $\widehat{u}_{k+1} \leq (1 + \rho)^k (V(S_1) + \ldots + V(S_k))$. We have

$$\begin{aligned}\widehat{u}_{k+1} &= \lceil \widehat{u}_k + V(S_k) \rceil \leq (1+\rho)(\widehat{u}_k + V(S_k)) \\ &\leq (1+\rho)\Big((1+\rho)^{k-1}(V(S_1) + \ldots + V(S_{k-1})) + V(S_k)\Big) \\ &\leq (1+\rho)^k (V(S_1) + \ldots + V(S_{k-1}) + V(S_k)),\end{aligned}$$

which completes the induction argument. Thus, we have $\widehat{u}_{m+1} \leq (1+\rho)^m (V(S_1) + \ldots + V(S_m)) \leq (1+\rho)^m \, n v_{\max}$. In this case, the result follows because $(1+\rho)^m \leq \left(1 + \frac{1}{2m}\right)^m \leq \exp(1/2) \leq 2$. ■

B.6 Assortment Optimization under a Cardinality Constraint

In this section, we consider a version of the CAPACITATED problem, where each product occupies one unit of space. Therefore, we can express the constraint $\sum_{k \in \mathcal{M}} C(S_k) \leq b$ as $\sum_{k \in M} |S_k| \leq b$, in which case, we ensure that the total number of products offered over all stages does not exceed b. Note that b is an integer. Otherwise, we can round it down to the nearest integer. First, we show that we can obtain an optimal solution by checking the expected revenue from $O(b^m n^{3m-1})$ possible solutions. The running time of this approach is polynomial in the number of products for a fixed number of stages. In general, since each one of as many as b products in an optimal solution can be offered in one of the m stages, the number of possible solutions is $O(n^b b^{m+1})$, which is exponential in the number of products even for a fixed number of stages. Second, treating the preference weights as the problem input, if all of the preference weights take on integer values, then we give a pseudo polynomial-time algorithm that obtains an optimal solution in $O(v_{\max} m n^5 b^2)$ operations. This algorithm is based on a dynamic programming formulation of the problem. If the preference weights take on rational values, then we can ensure that the preference weights take on integer values by scaling all of the preference weights by a constant, since the choice probabilities do not change by doing so. Third, by discretizing the state variable in the dynamic program through a geometric grid, we obtain an FPTAS. Our FPTAS obtains a $(1-\epsilon)$-approximate solution in $O(m^2\, n^4\, b^2\, \log(n v_{\max}/v_{\min})/\epsilon)$ operations. All of these three results are based on constructing a collection of candidate assortments for each stage so that an optimal assortment to offer in a stage lies within this collection. Therefore, we focus on constructing the collections of candidate assortments for the different stages. Throughout this section, when we refer to the CAPACITATED problem, we refer to the version where each product occupies one unit of space.

Constructing Collections of Candidate Assortments:

Note that Lemma B.2.1 continues to hold when each product occupies one unit of space. Thus, there exists an optimal solution $(S_1^*, \ldots, S_m^*)$ such that $S_k^* \subseteq \{j_k^* + 1, \ldots, j_{k+1}^*\}$ and $S_q^* \cap \{j_k^* + 1, \ldots, j_{k+1}^*\} = \emptyset$ for all $q \neq k$, for some $j_1^*, \ldots, j_{m+1}^*$ that satisfy $0 = j_1^* \leq j_2^* \leq \ldots \leq j_m^* \leq j_{m+1}^* = n$. To construct the collection of candidate assortments for stage k, we proceed under the assumption that we know the values of j_k^*, j_{k+1}^*, $|\cup_{q\neq k} S_q^*|$ along with $V(S_q^*)$ and $W(S_q^*)$ for all $q \neq k$. In this case, since the assortment that we offer in stage k affects the expected revenue in stages $k, \ldots, m$, we can recover an optimal assortment to offer in stage k by solving

$$\max_{\substack{S \subseteq \{j_k^*+1, \ldots, j_{k+1}^*\},\\ |S| \leq b - |\cup_{q\neq k} S_q^*|}} \left\{ \frac{\lambda_k\, W(S)}{(1 + \sum_{q=1}^{k-1} V(S_q^*))\,(1 + \sum_{q=1}^{k-1} V(S_q^*) + V(S))} + \sum_{\ell=k+1}^{m} \frac{\lambda_\ell\, W(S_\ell^*)}{(1 + \sum_{q=1, q\neq k}^{\ell-1} V(S_q^*) + V(S))\,(1 + \sum_{q=1, q\neq k}^{\ell} V(S_q^*) + V(S))} \right\},$$

where we use the fact that if we know the value of $|\cup_{q\neq k} S_q^*|$, then we can offer at most $b - |\cup_{q\neq k} S_q^*|$ products in stage k.

For notational brevity, we let $b_k^* = b - |\cup_{q\neq k} S_q^*|$, $f_\ell^* = \lambda_\ell\, W(S_\ell^*)/V(S_\ell^*)$ and $u_\ell^* = \sum_{q=1,q\neq k}^{\ell} V(S_q^*)$. We write the objective function of the problem above as

$$\frac{\lambda_k\, W(S)}{(1+u_{k-1}^*)\,(1+u_{k-1}^*+V(S))} + \sum_{\ell=k+1}^{m} f_\ell^* \left\{ \frac{1}{1+u_{\ell-1}^*+V(S)} - \frac{1}{1+u_\ell^*+V(S)} \right\}$$
$$= \frac{\lambda_k\, W(S)}{(1+u_{k-1}^*)\,(1+u_{k-1}^*+V(S))} + \sum_{\ell=k+1}^{m} (f_\ell^* - f_{\ell+1}^*) \left\{ \frac{1}{1+u_{k-1}^*+V(S)} - \frac{1}{1+u_\ell^*+V(S)} \right\}$$

with the convention that $f_{m+1}^* = 0$. The equality above follows by noting that the sum on the left side of the equality is equivalent to $f_{k+1}^* \frac{1}{1+u_k^*+V(S)} + \sum_{\ell=k+1}^{m} (f_{\ell+1}^* - f_\ell^*) \frac{1}{1+u_\ell^*+V(S)} = \sum_{\ell=k+1}^{m} (f_\ell^* - f_{\ell+1}^*) \frac{1}{1+u_k^*+V(S)} - \sum_{\ell=k+1}^{m} (f_\ell^* - f_{\ell+1}^*) \frac{1}{1+u_\ell^*+V(S)}$, along with the fact that $u_k^* = u_{k-1}^*$. In this case, to recover an optimal assortment to offer in stage k, we can solve the problem

$$\max_{\substack{S \subseteq \{j_k^*+1,\ldots,j_{k+1}^*\},\\ |S| \le b_k^*}} \left\{ \frac{\lambda_k\, W(S)}{(1+u_{k-1}^*)\,(1+u_{k-1}^*+V(S))} + \sum_{\ell=k+1}^{m} (f_\ell^* - f_{\ell+1}^*) \left\{ \frac{1}{1+u_{k-1}^*+V(S)} - \frac{1}{1+u_\ell^*+V(S)} \right\} \right\}. \quad \text{(B.4)}$$

In the next lemma, we show that we can efficiently construct a collection of candidate assortments that includes an optimal solution to problem (B.4) for any values of $\{(f_\ell^*, u_\ell^*) : \ell \in \mathcal{M},\ \ell \neq k\}$.

Lemma B.6.1 *Given j_k^*, j_{k+1}^* and b_k^*, there exists a collection of candidate assortment* $\textsc{Cand}_k(j_k^*, j_{k+1}^*, b_k^*)$ *with* $|\textsc{Cand}_k(j_k^*, j_{k+1}^*, b_k^*)| = O(n^2)$ *that includes an optimal solution to problem* (B.4) *for any values of* $\{(f_\ell^*, u_\ell^*) : \ell \in \mathcal{M},\ \ell \neq k\}$.

Proof: Let $g_\ell^* = (f_\ell^* - f_{\ell+1}^*)\,(u_\ell^* - u_{k-1}^*) \ge 0$. Multiplying the objective function of problem (B.4) by the constant $1 + u_{k-1}^*$, we can obtain an optimal assortment to offer in stage k by solving

$$\max_{\substack{S \subseteq \{j_k^*+1,\ldots,j_{k+1}^*\},\\ |S| \le b_k^*}} \left\{ \frac{1}{1+u_{k-1}^*+V(S)} \left\{ \lambda_k\, W(S) + (1+u_{k-1}^*) \sum_{\ell=k+1}^{m} \frac{g_\ell^*}{1+u_\ell^*+V(S)} \right\} \right\}.$$

Letting t^* be the optimal objective value of the problem above, t^* is no smaller than the objective function of the problem above at each S such that $S \subseteq \{j_k^*+1,\ldots,j_{k+1}^*\}$ and $|S| \le b_k^*$.

Therefore, letting $\mathcal{G} = \{S \subseteq \{j_k^*+1,\ldots,j_{k+1}^*\} : |S| \le b_k^*\}$, we can obtain an optimal solution to the problem above by using the so-called dual formulation, which is given by

$$\begin{aligned}
&\min\left\{ t \,:\, t \ge \frac{1}{1+u_{k-1}^*+V(S)} \left\{ \lambda_k\, W(S) + (1+u_{k-1}^*) \sum_{\ell=k+1}^{m} \frac{g_\ell^*}{1+u_\ell^*+V(S)} \right\} \;\; \forall\, S \in \mathcal{G} \right\} \\
&\quad = \min\left\{ t \,:\, t \ge \frac{\lambda_k\, W(S)}{1+u_{k-1}^*} - \frac{t\, V(S)}{1+u_{k-1}^*} + \sum_{\ell=k+1}^{m} \frac{g_\ell^*}{1+u_\ell^*+V(S)} \;\; \forall\, S \in \mathcal{G} \right\} \\
&\quad = \min\left\{ t \,:\, t \ge \max_{S \in \mathcal{G}} \left\{ \frac{\lambda_k\, W(S)}{1+u_{k-1}^*} - \frac{t\, V(S)}{1+u_{k-1}^*} + \sum_{\ell=k+1}^{m} \frac{g_\ell^*}{1+u_\ell^*+V(S)} \right\} \right\}
\end{aligned}$$

where the first equality follows by multiplying both sides of the constraint in the first minimization problem above by $1+u_{k-1}^*+V(S)$ and arranging the terms.

By the discussion so far, if t^* is an optimal solution to the last minimization problem above, then we can recover an optimal assortment to offer in stage k by replacing t in the maximization problem on the right side of the constraint with t^* and solving this maximization problem. Thus, we can obtain an optimal assortment to offer in stage k by solving the problem

$$\max_{S \in \mathcal{G}} \left\{ \frac{\lambda_k\, W(S)}{1+u_{k-1}^*} - \frac{t\, V(S)}{1+u_{k-1}^*} + \sum_{\ell=k+1}^{m} \frac{g_\ell^*}{1+u_\ell^*+V(S)} \right\} \tag{B.5}$$

for some value of t. We will construct a collection of $O(n^2)$ candidate assortments that includes an optimal solution to the problem above for any values of $\{(g_\ell^*, u_\ell^*) : \ell \in \mathcal{M},\ \ell \ne k\}$ and t.

Note that $\lambda_k W(S) = \lambda_k \sum_{i \in S} r_i v_i$ and $t\, V(S) = t \sum_{i \in S} v_i$. In this case, using the decision variables $\boldsymbol{x} = (x_1, \ldots, x_n)$ and noting the definition of $\mathcal{G}$, we write problem (B.5) equivalently as

$$\max_{\boldsymbol{x} \in \{0,1\}^n} \left\{ \frac{\lambda_k}{1 + u^*_{k-1}} \sum_{i \in \mathcal{N}} r_i\, v_i\, x_i - \frac{t}{1 + u^*_{k-1}} \sum_{i \in \mathcal{N}} v_i\, x_i + \sum_{\ell = k+1}^{m} \frac{g^*_\ell}{1 + u^*_\ell + \sum_{i \in \mathcal{N}} v_i\, x_i} \right.$$
$$\left. : \sum_{i \in \mathcal{N}} x_i \le b^*,\ x_i = 0 \ \ \forall\, i \notin \{j^*_k + 1, \ldots, j^*_{k+1}\} \right\}. \quad \text{(B.6)}$$

If $g^*_\ell \ge 0$, then the objective function of the problem above is convex in $\boldsymbol{x}$, in which case, an optimal solution occurs at an extreme point, so we can relax $\boldsymbol{x} \in \{0,1\}^n$ to $\boldsymbol{x} \in [0,1]^n$.

Indeed, we have $g^*_\ell \ge 0$. Note that $W(S^*_\ell)/V(S^*_\ell)$ is the weighted average of the revenues of the products in S^*_ℓ. By Lemma B.2.1, the revenues of the products in S^*_ℓ are larger than those of the products in $S^*_{\ell+1}$, so we have $W(S^*_\ell)/V(S^*_\ell) \ge W(S^*_{\ell+1})/V(S^*_{\ell+1})$. Furthermore, we have $\lambda_\ell \ge \lambda_{\ell+1}$, in which case, we get $f^*_\ell = \lambda_\ell\, W(S^*_\ell)/V(S^*_\ell) \ge \lambda_{\ell+1}\, W(S^*_{\ell+1})/V(S^*_{\ell+1}) = f^*_{\ell+1}$. We have $u^*_\ell \ge u^*_{k-1}$ for all $\ell \ge k+1$ as well, so $g^*_\ell = (f^*_\ell - f^*_{\ell+1})\,(u^*_\ell - u^*_{k-1}) \ge 0$. We solve problem (B.6) with $\boldsymbol{x} \in [0,1]^n$ in two stages. First, intuitively speaking, we guess the value of $\sum_{i \in \mathcal{N}} v_i\, x_i$. Second, we find solution $\boldsymbol{x}$ that maximizes the objective function, while satisfying our guess.

Using w to denote our guess of $\sum_{i \in \mathcal{N}} v_i\, x_i$, we can write the last problem in two stages. In particular, problem (B.6) is equivalent to the problem

$$\begin{aligned}
&\max_{w\in\mathbb{R}_+}\max_{\boldsymbol{x}\in[0,1]^n}\left\{\frac{\lambda_k}{1+u^*_{k-1}}\sum_{i\in\mathcal{N}} r_i\,v_i\,x_i-\frac{t\,w}{1+u^*_{k-1}}+\sum_{\ell=k+1}^{m}\frac{g^*_\ell}{1+u^*_\ell+w}\right.\\
&\qquad\qquad\left.:\ \sum_{i\in\mathcal{N}} x_i\le b^*,\quad \sum_{i\in\mathcal{N}} v_i\,x_i\le w,\quad x_i=0\ \ \forall\, i\notin\{j^*_k+1,\ldots,j^*_{k+1}\}\right\}\\
&=\max_{w\in\mathbb{R}_+}\left\{-\frac{t\,w}{1+u^*_{k-1}}+\sum_{\ell=k+1}^{m}\frac{g^*_\ell}{1+u^*_\ell+w}+\frac{\lambda_k}{1+u^*_{k-1}}\max_{\boldsymbol{x}\in[0,1]^n}\left\{\sum_{i\in\mathcal{N}} r_i\,v_i\,x_i\right.\right.\\
&\qquad\qquad\left.\left.:\ \sum_{i\in\mathcal{N}} x_i\le b^*,\quad \sum_{i\in\mathcal{N}} v_i\,x_i\le w,\quad x_i=0\ \ \forall\, i\notin\{j^*_k+1,\ldots,j^*_{k+1}\}\right\}\right\}.
\end{aligned}\tag{B.7}$$

The first problem above is equivalent to problem (B.6) since $g^*_\ell\ge 0$, in which case, the objective function of the first problem above is decreasing in w. Therefore, w takes the value $\sum_{i\in\mathcal{N}} v_i\,x_i$ in an optimal solution to the first problem above. Considering the second problem above, the inner maximization problem is a linear program (LP) with two constraints. We let $Q(w)$ be the optimal objective value and $\boldsymbol{x}^*(w)$ be an optimal solution of this LP as a function of w. It is a standard result in LP theory that $Q(w)$ is a piecewise linear function of w with $O(n^2)$ points of nondifferentiability. Furthermore, these points of nondifferentiability for $Q(\cdot)$ do not depend on the values of $\{(f^*_\ell,u^*_\ell):\ell\in\mathcal{M},\ \ell\ne k\}$ and t.

Letting $T=\sum_{i\in\mathcal{N}} v_i$, $\sum_{i\in\mathcal{N}} v_i\,x_i\in[0,T]$. We use $\{\widehat{w}_s : s\in\mathcal{Q}\}$ to denote the points of nondifferentiability of $Q(\cdot)$ with the convention that $0,T\in\mathcal{Q}$. We write problem (B.7) as

$$\begin{aligned}
&\max_{w\in\mathbb{R}_+}\left\{-\frac{t\,w}{1+u^*_{k-1}}+\sum_{\ell=k+1}^{m}\frac{g^*_\ell}{1+u^*_\ell+w}+\frac{\lambda_k}{1+u^*_{k-1}}\,Q(w)\right\}\\
&\qquad\qquad=\max_{s\in\mathcal{Q}}\left\{-\frac{t\,\widehat{w}_s}{1+u^*_{k-1}}+\sum_{\ell=k+1}^{m}\frac{g^*_\ell}{1+u^*_\ell+\widehat{w}_s}+\frac{\lambda_k}{1+u^*_{k-1}}\,Q(\widehat{w}_s)\right\},
\end{aligned}$$

where the equality holds since the objective function of the first problem above is convex in w, in which case, an optimal solution must occur at a point of nondifferentiability.

Thus, the collection $\{\boldsymbol{x}^*(\widehat{w}_s) : s \in Q\}$ with $|Q| = O(n^2)$ includes an optimal solution to problem (B.5) for any value of $\{(g^*_\ell, u^*_\ell) : \ell \in \mathcal{M},\ \ell \neq k\}$ and t. ■

The main computational effort in constructing the collection of candidate assortments $\textsc{Cand}_k(j_k, j_{k+1}, b_k)$ is to solve a parametric LP with $O(n^2)$ points of nodifferentiability.

A Polynomial-Time Algorithm for Fixed Number of Stages:

We can solve the CAPACITATED problem as follows. We construct the collection of candidate assortments $\textsc{Cand}_k(j_k, j_{k+1}, b_k)$ for all $j_k, j_{k+1} \in \mathcal{N}$, $b_k \leq b$, $k \in \mathcal{M}$. There are $O(n^{m-1})$ choices of $(j_1, \ldots, j_m)$ such that $0 = j_1 \leq j_2 \leq \ldots \leq j_m \leq j_{m+1} = n$, as well as $O(b^m)$ choices of $(b_1, \ldots, b_m)$ such that $\sum_{k \in \mathcal{M}} b_k = b$. For each choice of $(j_1, \ldots, j_m)$ and $(b_1, \ldots, b_m)$, since $|\textsc{Cand}_k(j_k, j_{k+1}, b_k)| = O(n^2)$, there are $O(n^{2m})$ ways of picking an assortment from the collection for each stage to construct a possible solution to the CAPACITATED problem. Thus, we get the next result.

Theorem B.6.2 *We can construct a collection of $O(b^m\, n^{3m-1})$ possible solutions to the* CAPACITATED *problem that is guaranteed to include an optimal solution to this problem. Letting* LP *be the number of operations to solve a parametric LP with $O(n^2)$ points of nondifferentiability, constructing these solutions requires $O(b\, n^2 \mathsf{LP} + b^m\, n^{3m-1})$ operations.*

A Pseudo Polynomial-Time Algorithm:

Noting the objective function of the CAPACITATED problem, knowing the value of j_k^* such that $S_1^* \cup \ldots \cup S_{k-1}^* \subseteq \{1, \ldots, j_k^*\}$, the value of b_k^* such that $|S_1^* \cup \ldots \cup S_{k-1}^*| = b_k^*$ and the value of u_{k-1}^* such that $\sum_{q=1}^{k-1} V(S_q^*) = u_{k-1}^*$ is enough to compute the optimal expected revenue in stages $k+1, \ldots, m$. Thus, we can solve the CAPACITATED problem by using dynamic programming. The decision epochs are the stages. The state variable at decision epoch k is (j_k, b_k, u_{k-1}) such that the assortments $S_1, \ldots, S_{k-1}$ offered in the previous stages satisfy $S_1 \cup \ldots \cup S_{k-1} \subseteq \{1, \ldots, j_k\}$, $|S_1 \cup \ldots \cup S_{k-1}| = b_k$ and $\sum_{q=1}^{k-1} V(S_q) = u_{k-1}$. The action at decision epoch k is the value of j_{k+1} such that the assortment offered in stage k satisfies $S_k \subseteq \{j_k+1, \ldots, j_{k+1}\}$, along with the assortment $S_k \in \cup_{d=0}^{b} \textsc{Cand}_k(j_k, j_{k+1}, d)$ offered in stage k. So, we consider the dynamic program

$$J_k(j, c, u) = \max_{\substack{(\ell, S)\,:\, \ell \in \{j, \ldots, n\} \\ S \in \cup_{d=0}^{b} \textsc{Cand}_k(j, \ell, d)}} \left\{ \frac{\lambda_k W(S)}{(1+u)(1+u+V(S))} + J_{k+1}(\ell\,,\, c+|S|\,,\, u+V(S)) \right\}$$

with the boundary condition that $J_{m+1}(j, c, u) = -\infty$ if $c > b$. If $c \leq b$, then $J_{m+1}(j, c, u) = 0$. Solving the dynamic program above requires constructing the collections of candidate assortments a priori.

Since $|\textsc{Cand}_k(j, \ell, d)| = O(n^2)$, at each decision epoch, there are $O(v_{\max}\, b\, n^2)$ possible values of the state variable and $O(bn^3)$ possible values of the action. So, we have the next result.

Theorem B.6.3 *Letting* LP *be as in Theorem B.6.2, we can obtain an optimal solution to the* CAPACITATED *problem in* $O(b\, n^2 \mathsf{LP} + v_{\max}\, m\, n^5\, b^2)$ *operations.*

Fully Polynomial-Time Approximation Scheme:

To obtain an FPTAS, we discretize the state variable in the dynamic program that we use to construct a pseudo polynomial-time algorithm. We consider the dynamic program

$$\Psi_k(j,c,u) = \max_{\substack{(\ell,S)\,:\,\ell\in\{j,\dots,n\} \\ S\in\cup_{d=0}^{b}\textsc{Cand}_k(j,\ell,d)}} \left\{ \frac{\lambda_k\, W(S)}{(1+u)\,(1+u+V(S))} + \Psi_{k+1}(\ell\,,\, c+|S|\,,\, \lceil u+V(S)\rceil) \right\}$$

with the boundary condition that $\Psi_{m+1}(j,c,u) = -\infty$ if $c > b$. If $c \leq b$, then $\Psi_{m+1}(j,c,u) = 0$. In the dynamic program above, the roundup operator $\lceil\cdot\rceil$ is as in Section 3.5.1.

Building on the dynamic program above, we can give an FPTAS by using an argument similar to the one in Section 3.5. In particular, once we compute the value functions $\{\Psi_k(j,c,u) : j = 0,\dots,n+1,\ c = 0,\dots,b,\ u \in \textsc{Dom},\ k \in \mathcal{M}\}$ through the dynamic program above, starting from state $(0,0,0)$, we follow the sequence of optimal state-action pairs to obtain the assortments $(\widehat{S}_1,\dots,\widehat{S}_m)$ over m stages. We can show that expected revenue from the assortments $(\widehat{S}_1,\dots,\widehat{S}_m)$ deviate from the optimal expected revenue by at most a factor of $(1+\rho)^{2m}$, where ρ is the size of the geometric grid. For given $\epsilon \in (0,1)$, setting $\rho = \epsilon/(2m)$, we get the next result.

Theorem B.6.4 *Letting* LP *be as in Theorem B.6.2, for each* $\epsilon \in (0,1)$, *we can obtain a* $(1-\epsilon)$*-approximate solution to the* Capacitated *problem in* $O(b\,n^2\mathsf{LP} + \frac{m^2\,n^4\,b^2}{\epsilon}\log(\frac{n\,v_{\max}}{v_{\min}}))$ *operations.*

B.7 Preprocessing the Dataset from Expedia

We explain our approach for preprocessing the dataset from Expedia and give a full description of the columns. The raw dataset includes about ten million rows and 54 columns. In some of the search queries, the price is given as the total amount over the whole length of the stay, whereas in some others, the price is given as the amount per night. It is not possible to reliably tell which approach is used in each search query. To avoid ambiguity, we focused our attention on the search queries for a single night stay and dropped the remaining search queries. Furthermore, we dropped the columns for which the entries are missing for more than 25% of the rows. Considering the remaining columns, we dropped the search queries for which the entries were missing in one of the remaining columns. Lastly, some rows in the dataset included entries that are too large or too small. We dropped all search queries which had an entry in a column that falls outside the 0.5-th and 99.5-th percentile band of the entries in the corresponding column. After preprocessing the dataset, we end up with 595,965 rows representing 34,561 search queries and 15 columns. We describe the first three columns in the main text.

The remaining 12 columns give the star rating and the average review score for the hotel, an indicator for whether the hotel is part of a chain, two location desirability scores, the average price of the hotel over the last trading period, the displayed price, an indicator for whether the hotel is on promotion, the number of days until the day of stay, the number of adults and children in the search query, and an indicator for whether the stay is over the weekend.

B.8 Running Time for Fitting the Choice Models

We used the routine fmincon in Matlab to maximize the log-likelihood functions for both choice models under consideration. In Table B.1, we give the average

P_0	$b = 1$	$b = 3$	$b = 5$	$b = 10$	$b = 20$
0.5	251.65	126.49	108.12	86.39	73.53
0.7	281.75	143.99	119.20	90.04	74.66
0.9	278.77	143.25	117.72	85.93	70.85

Table B.1: CPU seconds to estimate the parameters of our choice model.

CPU seconds to estimate the parameters of our multinomial logit model with impatient customers for different values of P_0 and b, where the average is computed over the 50 datasets. We observe that the CPU seconds to estimate the parameters of our choice model increase as b gets smaller so that we have more stages in the choice model. For a fixed value of b, the CPU seconds showed less than 20% variation from one dataset to another. For comparison purposes, we note that the average CPU seconds to estimate the parameters of the standard multinomial logit model is 18.34 seconds.

B.9 Upper Bound for Joint Pricing and Assortment Optimization

We give a proof for Proposition 3.6.1. By the discussion in Section 3.6.2, it is enough to show that $J_1^\alpha(L)$ is an upper bound on the optimal objective value of the problem

$$\max_{\boldsymbol{q}\in\mathbb{R}_+^m}\left\{\sum_{k\in\mathcal{M}}\lambda_k\,(q_{k-1}-q_k)\,(\log(q_{k-1}\,q_k)+\log\tfrac{\lambda_k}{\alpha}-1)\;:\;q_{k-1}\geq q_k\;\forall\,k\in\mathcal{M}\right\}+\alpha T.$$

Let $\boldsymbol{q}^*$ be an optimal solution to the problem above and p_k^* be such that $q_k^*\in[\nu_{p_k^*},\nu_{p_k^*+1}]$. Since $q_{k-1}^*\geq q_k^*$, we have $p_{k-1}^*\geq p_k^*$. Also, since $q_0^*=1$, we have $p_0^*=L$.

Let $Z_k=\sum_{\ell=k}^m\lambda_\ell\,(q_{\ell-1}^*-q_\ell^*)\,(\log(q_{\ell-1}^*\,q_\ell^*)+\log\frac{\lambda_\ell}{\alpha}-1)+\alpha T$ with $Z_{m+1}=\alpha T$. We use induction over the stages to show that $J_k^\alpha(p_{k-1}^*)\geq Z_k$. Since $J_{m+1}^\alpha(p)=\alpha T$, the result holds for stage $m+1$. Assuming that $J_{k+1}^\alpha(p_k^*)\geq Z_{k+1}$, we proceed to showing that $J_k^\alpha(p_{k-1}^*)\geq Z_k$. Since $p_k^*\leq p_{k-1}^*$, when computing $J_k^\alpha(p_{k-1}^*)$ though the dynamic program in Section 3.6.2, p_k^* is a feasible but not necessarily an optimal decision. Therefore, we get

$$\begin{aligned}J_k^\alpha(p_{k-1}^*)\;&\geq\;G_k^\alpha(p_{k-1}^*,p_k^*)+J_{k+1}^\alpha(p_k^*)\\&\geq\;\lambda_k\,(q_{k-1}^*-q_k^*)\,(\log(q_{k-1}^*\,q_k^*)+\log\tfrac{\lambda_k}{\alpha}-1)+Z_{k+1}\;=\;Z_k,\end{aligned}$$

where the second inequality uses the fact that $J_{k+1}^\alpha(p_k^*)\geq Z_{k+1}$ by the induction hypothesis, along with the fact that $G_k^\alpha(p,r)\geq\lambda_k\,(q_{k-1}-q_k)\,(\log(q_{k-1}\,q_k)+\log\frac{\lambda_k}{\alpha}-1)$ for all $q_{k-1}\in[\nu_p,\nu_{p+1}]$ and $q_k\in[\nu_r,\nu_{r+1}]$ by the definition of $G_k^\alpha(p,r)$, as well as noting that $q_{k-1}^*\in[\nu_{p_{k-1}^*},\nu_{p_{k-1}^*+1}]$ and $q_k^*\in[\nu_{p_k^*},\nu_{p_k^*+1}]$. Thus, the induction argument is complete. So, we have $J_1^\alpha(L)=J_1^\alpha(p_0^*)\geq Z_1$ and Z_1 is the optimal objective value of the problem at the beginning of the proof.

B.10 Upper Bound under a Space Constraint

To obtain an upper bound on the optimal expected revenue in the assortment problem under a space constraint, we consider the linear program

$$\textsc{Cap}(j,\ell,x,y) \;=\; \min_{\boldsymbol{w}\in[0,1]^{\ell-j}} \left\{ \sum_{i=j+1}^{\ell} c_i\, w_i \;:\; \sum_{i=j+1}^{\ell} v_i\, r_i\, w_i \geq x, \quad \sum_{i=j+1}^{\ell} v_i\, w_i \leq y \right\}. \tag{B.8}$$

If we impose the constraints $\boldsymbol{w} \in \{0,1\}^{\ell-j}$ in the problem above and drop the round down and up operators in (3.10), then the problem above and (3.10) solve the same knapsack problem.

If the problem above is infeasible, then we set $\textsc{Cap}(j,\ell,x,y) = +\infty$. Note that $W(S) \leq n\, w_{\max}$ and $V(S) \leq n\, v_{\max}$ for all $S \subseteq \mathcal{N}$. Also, letting $r_{\max} = \max\{r_i : i \in \mathcal{N}\}$, we have $\Pi(S_1,\ldots,S_m) \leq r_{\max}$ for all $(S_1,\ldots,S_m) \in \mathcal{F}$. Letting $B = \max\{n\, w_{\max}, n\, v_{\max}, r_{\max}\}$, we divide the interval $[0,B]$ into $L+1$ subintervals using $\nu_0,\ldots,\nu_{L+1}$ that satisfy $0 = \nu_0 < \nu_1 < \ldots < \nu_L < \nu_{L+1} = B$. Throughout this section, we define the round down operator "$\lfloor\cdot\rfloor$" that rounds its argument down to the closest point in $\{\nu_p : p = 0,\ldots,L+1\}$ when the argument is positive. That is, if $a \geq 0$, then $\lfloor a \rfloor = \max\{\nu_r : \nu_r \leq a,\ r = 0,\ldots,L+1\}$. If $a < 0$, then $\lfloor a \rfloor = -\infty$. We consider the dynamic program

$$\overline{\Psi}_k(j,u,z) = \min_{\substack{(\ell,p,r)\,:\,\ell\in\{j,\ldots,n\},\\ p\in\{0,\ldots,L\},\\ r\in\{1,\ldots,L+1\}}} \Bigg\{ \textsc{Cap}(j,\ell,\nu_p,\nu_r) + \overline{\Psi}_{k+1}\bigg(\ell\,,\,\lfloor u+\nu_{r-1}\rfloor\,,\,\bigg\lfloor z - \frac{\lambda_k\,\nu_{p+1}}{(1+u)\,(1+u+\nu_{r-1})}\bigg\rfloor\bigg)\Bigg\} \tag{B.9}$$

with the boundary condition that $\overline{\Psi}_{m+1}(j,u,z) = 0$ if $z \leq 0$. Otherwise, we have $\overline{\Psi}_{m+1}(j,u,z) = +\infty$. Note that the dynamic program above is analogous to the one in (3.12).

In the next proposition, we show that we obtain an upper bound on the optimal expected revenue in the CAPACITATED problem by solving the dynamic program above.

Proposition B.10.1 *Letting* $\bar{z}_{\text{APP}} = \max\{z \in \mathbb{R}_+ : \overline{\Psi}_1(0, 0, z) \le b\}$, $\bar{z}$ *is an upper bound on the optimal expected revenue in the* CAPACITATED *problem.*

Proof: Using an induction argument that is similar to the one in the proof of Lemma B.3.1, it follows that $\overline{\Psi}_k(j, u, z)$ is increasing in j, u and z. Let $(S_1^*, \ldots, S_m^*)$ be an optimal solution to the CAPACITATED problem. By Lemma B.2.1, there exist $j_1^*, \ldots, j_{m+1}^*$ satisfying $0 = j_1^* \le j_2^* \le \ldots \le j_m^* \le j_{m+1}^* = n$ such that $S_k^* \subseteq \{j_k^* + 1, \ldots, j_{k+1}^*\}$. Also, let $p_k^* = 0, \ldots, L$ and $r_k^* = 1, \ldots, L + 1$ be such that $W(S_k^*) \in [\nu_{p_k^*}, \nu_{p_k^*+1}]$ and $V(S_k^*) \in [\nu_{r_k^*-1}, \nu_{r_k^*}]$. Consider solving problem (B.8) with $j = j_k^*$, $\ell = j_{k+1}^*$, $x = \nu_{p_k^*}$ and $y = \nu_{r_k^*}$. Setting $w_i = 1$ if $i \in S_k$ and $w_i = 0$ if $i \notin S_k$ provides a feasible solution to this problem with the objective value $C(S_k^*)$. Thus, $\text{CAP}(j_k^*, j_{k+1}^*, \nu_{p_k^*}, \nu_{r_k^*}) \le C(S_k^*)$.

For notational brevity, we let $C_k^* = \sum_{q=k}^m C(S_q^*)$, $u_k^* = \sum_{q=1}^{k-1} V(S_q^*)$ and $z_k^* = \sum_{q=k}^m \frac{\lambda_q W(S_q^*)}{(1+u_q^*)(1+u_{q+1}^*)}$ with the convention that $C_{m+1}^* = 0$, $u_1^* = 0$ and $z_{m+1}^* = 0$. Observe that z_1^* corresponds to the optimal objective value of the CAPACITATED problem. We use induction over the stages to show that $\overline{\Psi}_k(j_k^*, u_k^*, z_k^*) \le C_k^*$. Since $z_{m+1}^* = 0$, we have $\overline{\Psi}_{m+1}(j_{m+1}^*, u_{m+1}^*, z_{m+1}^*) = 0 = C_{m+1}^*$. Therefore, the result holds for the base case. Assuming that $\overline{\Psi}_{k+1}(j_{k+1}^*, u_{k+1}^*, z_{k+1}^*) \le C_{k+1}^*$, we proceed to showing that $\overline{\Psi}_k(j_k^*, u_k^*, z_k^*) \le C_k^*$. Using the fact that $\overline{\Psi}(j, u, z)$ is increasing in u and z along with

$\lfloor a \rfloor \leq a$ and noting that $W(S_k^*) \leq \nu_{p_k^*+1}$ and $V(S_k^*) \geq \nu_{r_k^*-1}$, we have

$$\overline{\Psi}_{k+1}\Big(j_{k+1}^*\,,\, \lfloor u_k^* + \nu_{r_k^*-1} \rfloor\,,\, \Big\lfloor z_k^* - \frac{\lambda_k\, \nu_{p_k^*+1}}{(1+u_k^*)\,(1+u_k^*+\nu_{r_k^*-1})} \Big\rfloor\Big)$$

$$\leq\ \overline{\Psi}_{k+1}\Big(j_{k+1}^*\,,\, u_k^* + V(S_k^*)\,, z_k^* - \frac{\lambda_k\, W(S_k^*)}{(1+u_k^*)\,(1+u_k^*+V(S_k^*))}\Big)\ =\ \overline{\Psi}_{k+1}(j_{k+1}^*\,,\, u_{k+1}^*\,,\, z_{k+1}^*),$$

where the equality above uses the definition of u_k^* and z_k^*. Consider computing $\overline{\Psi}_k(j_k^*, u_k^*, z_k^*)$ through the dynamic program in (B.9). Since $j_{k+1}^* \geq j_k^*$, the solution $(j_{k+1}^*, p_k^*, r_k^*)$ is feasible but not necessarily optimal to the minimization problem on the right side of (B.9) when we solve this problem with $(j, u, z) = (j_k^*, u_k^*, z_k^*)$. Therefore, we have the chain of inequalities

$$\begin{aligned}\overline{\Psi}_k(j_k^*, u_k^*, z_k^*)\ &\leq\ \text{CAP}(j_k^*, j_{k+1}^*, \nu_{p_k^*}, \nu_{r_k^*}) \\ &\qquad + \overline{\Psi}_{k+1}\Big(j_{k+1}^*\,,\, \lfloor u_k^* + \nu_{r_k^*-1} \rfloor\,,\, \Big\lfloor z_k^* - \frac{\lambda_k\, \nu_{p_k^*+1}}{(1+u_k^*)\,(1+u_k^*+\nu_{r_k^*-1})} \Big\rfloor\Big) \\ &\overset{(a)}{\leq}\ C(S_k^*) + \overline{\Psi}_{k+1}(j_{k+1}^*\,,\, u_{k+1}^*\,,\, z_{k+1}^*) \\ &\overset{(b)}{\leq}\ C(S_k^*) + C_{k+1}^*\ \overset{(c)}{=}\ C_k^*,\end{aligned}$$

where (a) follows from the inequality that we give earlier in this paragraph and the fact that $\text{CAP}(j_k^*, j_{k+1}^*, \nu_{p_k^*}, \nu_{r_k^*}) \leq C(S_k^*)$, (b) uses the induction hypothesis and (c) is by the definition of C_k^*. Thus, the induction argument is complete, in which case, noting that $j_1^* = 0$ and $u_1^* = 0$, we obtain $\overline{\Psi}_1(0, 0, z_1^*) \leq C_1^* = \sum_{k \in \mathcal{M}} C_k(S_k^*) \leq b$, where the last inequality uses the fact that $(S_1^*, \ldots, S_m^*)$ is a feasible solution to the CAPACITATED problem. Therefore, z_1^* is a feasible solution to the problem $\bar{z}_{\text{APP}} = \max\{z \in \mathbb{R}_+ : \overline{\Psi}_1(0, 0, z) \leq b\}$, which implies that the optimal objective value of this problem is at least as large as z_1^*. In other words, we have $\bar{z}_{\text{APP}} \geq z_1^*$. In this case, the result follows by noting that z_1^* is the optimal objective value of the CAPACITATED problem. ■

Note that the upper bound in the proposition above holds for any choice of $\nu_0, \ldots, \nu_{L+1}$ that satisfy $0 = \nu_0 < \nu_1 < \ldots < \nu_L < \nu_{L+1} = B$.

BIBLIOGRAPHY

[1] A. Aouad, V. Farias, and R. Levi. Assortment optimization under consider-then-choose choice models. Technical report, MIT, Massachusetts, MA, 2016.

[2] A. Aouad, J. Feldman, D. Segev, and D. J. Zhang. Click-based MNL: Algorithmic frameworks for modeling click data in assortment optimization. Technical report, Washington University, St. Louis, MO, 2019.

[3] A. Aouad and D. Segev. Display optimization for vertically differentiated locations under multinomial logit preferences. Technical report, MIT, Massachusetts, MA, 2018.

[4] Ali Aouad, Vivek Farias, Retsef Levi, and Danny Segev. The approximability of assortment optimization under ranking preferences. *Operations Research*, 66(6):1661–1669, 2018.

[5] J. Blanchet, G. Gallego, and V. Goyal. A Markov chain approximation to choice modeling. *Operations Research*, 64(4):886–905, 2016.

[6] J. J. M. Bront, I. Mendez Diaz, and G. Vulcano. A column generation algorithm for choice-based network revenue management. *Operations Research*, 57(3):769–784, 2009.

[7] J.M. Davis, G. Gallego, and H. Topaloglu. Assortment optimization under variants of the nested logit model. *Operations Research*, 62(2):250–273, 2014.

[8] M. Derakhshan, N. Golrezaei, V. Manhshadi, and V. Mirrokni. Product ranking on online platforms. Technical report, MIT, Massachusetts, MA, 2018.

[9] A. Desir and V. Goyal. Near-optimal algorithms for capacity constrained assortment optimization. Technical report, Columbia University, New York, NY, 2014.

[10] A. Desir, V. Goyal, S. Jagabathula, and D. Segev. Assortment optimization under the mallows model. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4700–4708. Curran Associates, Inc., 2016.

[11] A. Desir, V. Goyal, D. Segev, and C. Ye. Capacity constrained assortment optimization under the markov chain based choice model. Technical report, Columbia University, New York, NY, 2015.

[12] James Dong, A. Serdar Simsek, and Huseyin Topaloglu. Pricing problems under the markov chain choice model. *Production and Operations Management*, 28(1):157–175, 2019.

[13] Vivek F. Farias, Srikanth Jagabathula, and Devavrat Shah. A non-parametric approach to modeling choice with limited data. *Management Science*, 59(2):305–322, 2013.

[14] J. Feldman, N. Liu, H. Topaloglu, and S. Ziya. Appointment scheduling under patient preference and no-show behavior. *Operations Research*, 62(4):794–811, 2014.

[15] J. Feldman, A. Paul, and H. Topaloglu. Assortment optimization with small consideration sets. *Operations Research*, (to appear), 2019.

[16] J. Feldman and H. Topaloglu. Technical note: Capacitated assortment optimization under the multinomial logit model with nested consideration sets. *Operations Research*, 66(2):380–391, 2018.

[17] J. B. Feldman and H. Topaloglu. Technical note: Capacity constraints across nests in assortment optimization under the nested logit model. *Operations Research*, 63(4):812–822, 2015.

[18] A. Flores, G. Berbeglia, and P. van Hentenryck. Assortment optimization under the sequential multinomial logit model. *European Journal of Operational Research*, 273(3):1052–1064, 2019.

[19] G. Gallego, G. Iyengar, R. Phillips, and A. Dubey. Managing flexible products on a network. Computational Optimization Research Center Technical Report TR-2004-01, Columbia University, 2004.

[20] G. Gallego, A. Li, V. A. Truong, and X. Wang. Approximation algorithms for product framing and pricing. Technical report, Columbia University, New York, NY, 2016.

[21] G. Gallego, R. Ratliff, and S. Shebalov. A general attraction model and sales-based linear programming formulation for network revenue

management under customer choice. *Operations Research*, 63(1):212–232, 2015.

[22] G. Gallego and H. Topaloglu. Constrained assortment optimization for the nested logit model. *Management Science*, 60(10):2583–2601, 2014.

[23] G. Gallego and R. Wang. Multi-product price optimization and competition under the nested attraction model. *Operations Research*, 62(2):450–461, 2014.

[24] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, NY, 1979.

[25] W. J. Hopp and X. Xu. Product line selection and pricing with modularity in design. *Manufacturing & Service Operations Management*, 7(3):172–187, 2005.

[26] S. Jagabathula. Assortment optimization under general choice. Technical report, NYU, New York, NY, 2016.

[27] Kaggle. Personalize Expedia hotel searches. Last checked: August 5, 2019, 2013.

[28] H. Li and W. T. Huh. Pricing multiple products with the multinomial logit and nested models: Concavity and implications. *Manufacturing & Service Operations Management*, 13(4):549–563, 2011.

[29] H. Li and S. Webster. Optimal pricing of correlated product options under the paired combinatorial logit model. *Operations Research*, 65(5):1215–1230, 2017.

[30] N. Liu, Y. Ma, and H. Topaloglu. Assortment optimization under the multinomial logit model with sequential offerings. *INFORMS Journal on Computing*, (to appear), 2019.

[31] N. Liu, P. M. van de Ven, and B. Zhang. Managing appointment booking under customer choices. Technical report, Boston College, Boston, MA, 2017.

[32] Joern Meissner, Arne Strauss, and Kalyan Talluri. An enhanced concave

program relaxation for choice network revenue management. *Production and Operations Management*, 22(1):71–87, 2012.

[33] I. Mendez-Diaz, J. J. M. Bront, G. Vulcano, and P. Zabala. A branch-and-cut algorithm for the latent-class logit assortment problem. *Discrete Applied Mathematics*, 36:383–390, 2010.

[34] S. Mittal and A. S. Schulz. A general framework for designing approximation schemes for combinatorial optimization problems with many objectives combined into one. *Operations Research*, 61(2):389–397, 2013.

[35] S. Najafi, I. Duenyas, S. Jasin, and J. Uichanco. Multi-product dynamic pricing with limited inventories under cascade click model. Technical report, University of Michigan, Ann Arbor, MI, 2019.

[36] P. Rusmevichientong, Z.-J. M. Shen, and D. B. Shmoys. Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Operations Research*, 58(6):1666–1680, 2010.

[37] Paat Rusmevichientong, David B. Shmoys, Chaoxu Tong, and Huseyin Topaloglu. Assortment optimization under the multinomial logit model with random choice parameters. *Production and Operations Management*, 23(11):2023–2039, 2014.

[38] Paat Rusmevichientong and Huseyin Topaloglu. Robust assortment optimization in revenue management under the multinomial logit choice model. *Operations Research*, 60(4):865–882, 2012.

[39] J.-S. Song and Z. Xue. Demand management and inventory control for substitutable products. Technical report, Duke University, Durham, NC, 2007.

[40] M. Sumida, G. Gallego, P. Rusmevichientong, H. Topaloglu, and J. M. Davis. Revenue-utility tradeoff in assortment optimization under the multinomial logit model with totally unimodular constraints. Technical report, Cornell University, Ithaca, NY, 2019.

[41] K. Talluri and G. van Ryzin. Revenue management under a general discrete choice model of consumer behavior. *Management Science*, 50(1):15–33, 2004.

[42] G. Vulcano, G. J. van Ryzin, and R. Ratliff. Estimating primary demand

for substitutable products from sales transaction data. *Operations Research*, 60(2):313–334, 2012.

[43] R. Wang. Capacitated assortment and price optimization under the multinomial logit model. *Operations Research Letters*, 40(6):492–497, 2012.

[44] R. Wang. Assortment management under the generalized attraction model with a capacity constraint. *Journal of Revenue and Pricing Management*, 12(3):254–270, 2013.

[45] R. Wang and O. Sahin. The impact of consumer search cost on assortment planning and pricing. *Management Science*, 64(8):3649–3666, 2018.

[46] H. Zhang, P. Rusmevichientong, and H. Topaloglu. Technical note: Multiproduct pricing under the generalized extreme value models with homogeneous price sensitivity parameters. *Operations Research*, 66(6):1559–1570, 2018.