

# CUSTOMER CHOICE MODELS AND ASSORTMENT OPTIMIZATION

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

James Mario Davis

August 2015

© 2015 James Mario Davis  
ALL RIGHTS RESERVED

# CUSTOMER CHOICE MODELS AND ASSORTMENT OPTIMIZATION

James Mario Davis, Ph.D.

Cornell University 2015

This thesis handles a fundamental problem in retail: given an enormous variety of products which does the retailer display to its customers? This is the assortment planning problem. We solve this problem by developing algorithms that, given input parameters for products, can efficiently return the set of products that should be displayed. To develop these algorithms we use a mathematical model of how customers react to displayed items, a customer choice model. Below we consider two classic customer choice models, the Multinomial Logit model and Nested Logit model. Under each of these customer choice models we develop algorithms that solve the assortment planning problem. Additionally, we consider the constrained assortment planning problem where the retailer must display products to customers but must also satisfy operational constraints.

## **BIOGRAPHICAL SKETCH**

I was raised in the farmlands of Vineland New Jersey. Fond memories of picking vegetables are well mixed with less fond memories of my early schooling. My schooling pleasantly ended after high school, where I passed enough classes to be forcefully ejected from the system with a degree. I wandered into a job as a cook at a local grocery store where I learned that, in contrast to myself, the average grocery store employee does not read text books on their lunch break. I decided to explore my desire to learn, enrolled in Cumberland Community College, transferred to Rutgers University Camden, and then moved to Cornell.

I dedicate this thesis to the amazing people willing to tolerate my special brand of insanity and laziness. This includes my mother, my wife, and my two advisers, Huseying Topaloglu and David P. Williamson. However, my track through higher education would not be possible without the influence and wisdom of my undergraduate adviser, Rajiv Gandhi.

## ACKNOWLEDGEMENTS

Half of my success in life comes from embracing the opportunities other people make for me. The other half comes from being lucky enough that those people are in my life.

The NSF has graciously supported this work both directly, through the NSF Graduate Research Fellowship Program grant DGE-0707428 and through grants CCF-111525 and CMMI-0969113.

## TABLE OF CONTENTS

Biographical Sketch . . . . .	iii
Dedication . . . . .	iv
Acknowledgements . . . . .	v
Table of Contents . . . . .	vi
List of Tables . . . . .	viii
List of Figures . . . . .	ix
<b>1 Introduction</b>	<b>1</b>
1.1 Overview . . . . .	1
1.2 Constrained Assortment Planning Under the Multinomial Logit Model	3
1.2.1 Motivation and Literature Review . . . . .	4
1.2.2 Main Results . . . . .	5
1.3 Assortment Planning Under the Nested Logit Model . . . . .	10
1.3.1 Motivation and Literature Review . . . . .	12
1.3.2 Main Results . . . . .	14
1.4 Quality Consistent Pricing Under the Nested Logit Model . . . . .	17
1.4.1 Motivation and Literature Review . . . . .	20
1.4.2 Main Results . . . . .	22
1.5 Assortment Planning Over Time . . . . .	24
1.5.1 Motivation and Literature Review . . . . .	26
1.5.2 Main Results . . . . .	27
<b>2 Constrained Assortment Optimization Under the Multinomial Logit Model</b>	<b>28</b>
2.1 Introduction . . . . .	28
2.2 Literature Review . . . . .	31
2.3 Problem Formulation . . . . .	33
2.4 Main Result . . . . .	34
2.5 Applications . . . . .	37
2.5.1 Cardinality Constraints . . . . .	37
2.5.2 Display Location Effects . . . . .	39
2.5.3 Pricing with a Finite Price Menu . . . . .	40
2.5.4 Quality Consistent Pricing . . . . .	42
2.5.5 Product Precedence Constraints . . . . .	46
<b>3 Assortment Optimization Under the Nested Logit Model</b>	<b>48</b>
3.1 Introduction . . . . .	48
3.2 Literature Review . . . . .	51
3.3 Problem Formulation . . . . .	54
3.4 Linear Programming Representation . . . . .	61
3.5 Competitive Products and Fully-Captured Nests . . . . .	66
3.6 Possibly Synergistic Products and Fully-Captured Nests . . . . .	71

3.6.1	Computational Complexity . . . . .	72
3.6.2	Performance of Nested-by-Revenue Assortments . . . . .	76
3.7	Competitive Products and Partially-Captured Nests . . . . .	80
3.8	Possibly Synergistic Products and Partially-Captured Nests . . . . .	86
3.8.1	Performance Guarantee . . . . .	87
3.8.2	Near-Optimal Assortments . . . . .	89
3.9	Upper Bounds on Optimal Expected Revenue . . . . .	93
3.10	Computational Experiments . . . . .	96
3.10.1	Experimental Setup . . . . .	96
3.10.2	Computational Results . . . . .	100
3.11	Conclusions . . . . .	106
3.12	Appendix: Omitted Proofs . . . . .	110
3.12.1	Proof of Theorem 3.6.3 . . . . .	110
3.12.2	Proof of Theorem 3.7.1 . . . . .	115
3.12.3	Proof of Theorem 3.7.3 . . . . .	117
3.12.4	Proof of Theorem 3.8.1 . . . . .	119
3.12.5	Knapsack Problems with Equality Constraints . . . . .	123
3.12.6	Proof of Theorem 3.8.2 . . . . .	130
<b>4</b>	<b>Quality Consistent Pricing Under the Nested Logit Model</b>	<b>138</b>
4.1	Introduction . . . . .	138
4.2	Literature Review . . . . .	142
4.3	Price Ladders Inside Nests . . . . .	145
4.3.1	Problem Formulation . . . . .	145
4.3.2	Connection to a Fixed Point Representation . . . . .	148
4.3.3	Characterizing Candidate Price Vectors . . . . .	151
4.3.4	Counting Candidate Price Vectors . . . . .	155
4.3.5	Constructing Candidate Price Vectors . . . . .	158
4.4	Price Ladders Between Nests . . . . .	161
4.4.1	Problem Formulation . . . . .	161
4.4.2	Connection to a Fixed Point Representation . . . . .	163
4.4.3	Characterizing Candidate Price Vectors . . . . .	168
4.4.4	Counting Candidate Price Vectors . . . . .	170
4.4.5	Constructing Candidate Price Vectors . . . . .	172
4.5	Numerical Experiments . . . . .	174
4.5.1	Price Ladders Inside Nests . . . . .	174
4.5.2	Price Ladders Between Nests . . . . .	178
<b>5</b>	<b>Assortment Optimization Over Time</b>	<b>185</b>
5.1	Introduction . . . . .	185
5.2	Literature Review . . . . .	185
5.3	Preliminaries . . . . .	186
5.4	Algorithm . . . . .	187
5.5	Hardness . . . . .	190



## LIST OF TABLES

1.1	Summary of results. $\rho$ is the ratio of largest to smallest revenue. $\kappa$ is the ratio of largest to smallest preference weight. . . . .	17
3.1	Four cases considered in the chapter. The ratio between the largest and the smallest product revenues in a nest is bounded by $\rho$ , the ratio between the largest and the smallest attractiveness parameters in a nest is bounded by $\kappa$ and the dissimilarity parameters are bounded by $\bar{\gamma}$ . . . . .	51
3.2	Performance of nested-by-revenue assortments. . . . .	103
3.3	Performance of nested-by-preference-and-revenue assortments. . . . .	104
3.4	Performance of powers-of- $\delta$ assortments. . . . .	104
3.5	Performance of the three different collections of assortments as a function of $\gamma^U$ . . . . .	107
4.1	Computational results for test problems with price ladders inside nests. . . . .	177
4.2	Computational results for test problems with price ladders between nests. . . . .	179

## LIST OF FIGURES

2.1	Shortest path problem in the quality consistent pricing setting. . .	45
3.1	Optimality gaps for nested-by-preference-and-revenue assortments.	106
4.1	Nested logit model, price ladders inside nests and price ladders between nests. . . . .	140

# CHAPTER 1

## INTRODUCTION

### 1.1 Overview

While retailers make decisions about managing inventory or fulfilling orders they must carefully consider what to display to customers. The assortment of products a retailer chooses shapes demand and, as a result, affects the other business decisions. At a physical retailer the assortment decision is strategic. What products do we put on the shelves this week? In e-commerce the decision can be strategic, but can also be real time and, potentially, personalized. What products do I show to *this* customer *right now* in order to maximize revenue and grow market share?

This is the assortment planning problem and is the primary driver of my thesis research. Which products does a retailer display to its customers? I answer it within the framework of Operations Research, using mathematical models and data driven techniques. Of special importance is the model of how customers purchase products, the customer choice model. Formally, the retailer has a stock of products,  $N$ , and for every  $S \subseteq N$  the customer choice model gives the probability product  $j \in S$  is purchased,  $P_j(S)$ . The critical feature of a choice model is that the probability a product is purchased depends on which assortment the retailer offers to customers.

To understand our approach, and the place of a customer choice model, we can put the assortment planning decision in a larger context. We can divide the decision making into two separate stages: estimation and optimization. The link between these two stages is the customer choice model. In the estimation stage we are presented with raw transaction data and use statistical methods to fit the customer choice model to the data. In the second stage we assume the parameters

of the model are given and use them as input to an optimization routine. We focus on the second, optimization stage.

We would like to optimize the assortment of products a retailer displays to its customers. In what follows we will be most interested in optimizing expected revenue. We incorporate revenues by assigning each product  $j$  a fixed revenue  $r_j$ ; if selling a unit of product  $j$  involves a constant cost for each unit sold, then we assume that  $r_j$  is the profit from the sale, given by the difference between revenue and cost. Given the revenues  $r_j$  and the function  $P_j(S)$  the assortment planning problem is to find the set  $S^*$  such that:

$$S^* = \operatorname{argmax}_S \sum_{j \in S} r_j P_j(S).$$

In addition to the pure assortment planning problem we will be interested in variations. These variations can include additional constraints on the assortments that can be offered. For example, an online retailer must display products on a web page with limited space; there may only be enough room to display  $k$  products. This introduces an operational constraint: the retailer can not display an assortment with more than  $k$  products. This constraint induces a collection of feasible assortments  $\mathcal{F} = \{S : |S| \leq k\}$  and the retailer must offer some  $S \in \mathcal{F}$ . The retailer is then interested in finding a set  $S^*$  such that

$$S^* = \operatorname{argmax}_{S \in \mathcal{F}} \sum_{j \in S} r_j P_j(S).$$

This is the constrained assortment planning problem. Different constraints will result in different  $\mathcal{F}$ .

We will also be interested in variations of the assortment planning problem that include additional problem dynamics. Notably we will be interested in pricing problems, where in addition to selecting an assortment to display the retailer must also post prices for products. The probability a product is purchased depends

on the assortment offered and the posted prices. We will also be interested in location based effects, where the probability a product is purchased depends on the offered assortment and the locations of the offered products. Including these, and other problem dynamics will consist of reducing the dynamics to a constrained assortment planning problem of a special variety.

## 1.2 Constrained Assortment Planning Under the Multinomial Logit Model

The starting point for our work considers constrained assortment planning problems assuming customers choose according to the multinomial logit model (MNL). The set of available products is  $N$ . The per-unit revenue associated with product  $j$  is  $r_j$ . The key feature of the MNL is the introduction of a preference  $v_j$  for each product  $j$ . The preference weight captures the relative attractiveness of the product. To capture the product offer decisions we introduce the decision variable  $x_j \in \{0, 1\}$  such that  $x_j = 1$  if we offer product  $j$ ; otherwise  $x_j = 0$ . Under the MNL, if the product offer decisions are given by the vector  $x = \{x_j : j \in N\} = \{0, 1\}^{|N|}$ , then a customer purchases product  $j$  with probability  $P_j(x) = v_j x_j / (v_0 + \sum_{k \in N} v_k x_k)$ . Therefore, if the products that we offer correspond to the vector  $x$ , then the expected revenue obtained from a customer can be written as

$$R(x) = \sum_{j \in N} r_j P_j(x) = \frac{\sum_{j \in N} r_j v_j x_j}{1 + \sum_{j \in N} v_j x_j}.$$

For a constraint matrix  $A = [a_{ij}]_{i \in M, j \in N}$  with dimensions  $|M| \times |N|$ , the feasible set of product offer decisions are given by  $\mathcal{F} = \{x \in \{0, 1\}^{|N|} : \sum_{j \in N} a_{ij} x_j \leq b_i \forall i \in M\}$ . Our goal is to find a set of feasible products to offer so as to maximize

the expected revenue obtained from each customer, yielding the problem

$$z^* = \max_{x \in \mathcal{F}} R(x). \tag{1.1}$$

### 1.2.1 Motivation and Literature Review

The MNL has a strong presence in the literature. There are known methods to estimate the MNL parameters very quickly; see [59]. These methods run very quickly, making the MNL a very attractive option in practice. Additionally, the MNL has been included in many treatments of network revenue management problems, where an airline must make decisions about offering itineraries based on the capacity of individual flight legs in the underlying network; see [18], [31], [36], and [42]. And, perhaps most importantly, the MNL is used considerably in practice. These considerations make the MNL a natural choice model to study; extensions of the MNL can easily be incorporated in existing revenue management systems.

The unconstrained assortment planning problem under the MNL has been studied by [57]. They show that the optimal assortment can be obtained by greedily adding products into the offered assortment in the order of decreasing revenues. [22] give a linear programming formulation for the unconstrained assortment planning problem under the MNL. Our main result will introduce a linear program that can be viewed as a considerable extension of the linear program in [22].

The advances in the unconstrained assortment planning problem offer limited options to practitioners. In practical retail settings constraints on the offered assortment are essential; retailers are not able to offer an unlimited number of products, for example. [50] consider assortment problems with a limit on the total number of offered products. The algorithms they develop are special purpose combinatorial algorithms. By contrast we will give a direct linear programming formulation that is considerably easier to implement. Our approach also extends

to more general cardinality constraints.

In addition to constrained assortment planning problems we also use our framework to incorporate additional problem dynamics. We will consider location based effects, where the preference weight of a product depends on *where* it is offered. Additionally, we introduce joint assortment and pricing problems, where the preference weight of a product depends on its posted price. In Section 4 we consider a more general framework for joint assortment and pricing problems. In this section we only give a sketch of how our techniques apply to pricing problems.

### 1.2.2 Main Results

Problem 1.1 has a nonlinear objective function and integrality requirements on its decision variables. Hence it is intractable. To overcome this intractability we introduce a linear program

$$\begin{aligned}
 \max \quad & \sum_{j \in N} r_j w_j & (1.2) \\
 \text{st} \quad & \sum_{j \in N} w_j + w_0 = 1 \\
 & \sum_{j \in N} a_{ij} \frac{w_j}{v_j} \leq b_i \frac{w_0}{v_0} & \forall i \in M \\
 & 0 \leq \frac{w_j}{v_j} \leq \frac{w_0}{v_0} & \forall j \in N,
 \end{aligned}$$

where the decision variables are  $\{w_j : j \in N \cup \{0\}\}$ . In this linear program we interpret the decision variable  $w_j$  as the probability that a customer purchases product  $j$  and  $w_0$  as the probability that a customer leaves without making a purchase. The first constraint ensures that a customer either purchases a product or leaves without purchasing. Interestingly, the second set of constraints are enough to ensure that the product offer decisions are chosen within the feasible set  $\mathcal{F}$ .

The third set of constraints ensure the connection between the probability that a customer purchases a product and leaves without purchasing anything. Our main result is that when  $A$  is a totally unimodular (TU) matrix then the LP captures the constrained assortment planning problem exactly.

**Theorem 1.2.1.** *When the constraint matrix  $A$  is TU problems (1.1) and (2.3) have the same optimal objective value and we can construct an optimal solution to one of these problems by using an optimal solution to the other.*

We give a number of specific cases where we can use Theorem 2.4.1 to obtain the optimal solutions to certain constrained assortment planning problems under the MNL. For a majority of these cases our results provide the first tractable algorithms to obtain optimal solutions. Additionally, we reduce problems that incorporate additional problem dynamics to constrained assortment planning problems for which Theorem 2.4.1 applies.

**Cardinality Constraints:** Consider the case where the total number of products that can be offered is limited to  $b$ . So, the feasible set of product offer decisions can be written as  $\mathcal{F} = \{x \in \{0, 1\}^{|N|} : \sum_{j \in N} x_j \leq b\}$ . These constraints are TU. [50] give an efficient algorithm for finding the optimal set of products to offer under MNL with a cardinality constraint but our use of Theorem 2.4.1 allows us to solve this problem directly by using a linear program. Furthermore, by building on this theorem, we can find the optimal solution under more general cardinality constraints. We give two examples of more general cardinality constraints below.

Consider the case where there are  $K$  nested subsets of products such that  $S_1 \subset S_2 \subset \dots \subset S_K \subset N$  and there are integers  $b_1 \leq b_2 \leq \dots \leq b_K$  associated with each one of these subsets. The total number of products that we can offer in subset  $S_k$  is limited to  $b_k$ . Thus, the feasible set of product offer decisions can be written as  $\mathcal{F} = \{x \in \{0, 1\}^{|N|} : \sum_{j \in S_k} x_j \leq b_k \forall k = 1, \dots, K\}$ . This structure



can arise when a more specialized set of products,  $S_1$ , is allocated less space than a more general set  $S_2$ .

As another example, consider the case where the products are partitioned into  $K$  disjoint subsets  $S_1, S_2, \dots, S_K$ . Without loss of generality, we assume that the products are indexed by the integers  $\{1, \dots, n\}$  and we have  $S_k = \{i_k, \dots, i_{k+1} - 1\}$  with  $i_1 = 1$  and  $i_{K+1} = n + 1$ . The total number of products we can offer in subset  $S_k$  is limited to  $b_k$ . In this case, the feasible set of product offer decisions is  $\mathcal{F} = \{x \in \{0, 1\}^{|N|} : \sum_{j=i_k}^{i_{k+1}-1} x_j \leq b_k \forall k = 1, \dots, K\}$  and this constraint matrix still corresponds to an interval matrix. This structure can arise when  $S_1$  and  $S_2$  correspond to different product categories that we would like to limit independently.

**Product Precedence Constraints:** We consider assortment planning problems where a particular product cannot be offered to customers unless a certain set of related products are also offered. For example, it may not be possible to offer the brand name version of a drug unless the generic version is offered. To model such product precedence constraints, we use  $S_j \subset N$  to denote the set of products that we need to offer to be able to offer product  $j$ . So, the feasible set of product offer decisions is given by  $\mathcal{F} = \{x \in \{0, 1\}^{|N|} : x_j - x_i \leq 0 \forall j \in N, i \in S_j\}$ , indicating that we can have  $x_j = 1$  only when  $x_i = 1$  for all  $i \in S_j$ . In this constraint matrix, each row includes only a +1 and a -1. Such matrices are known to be TU. We observe that the subsets  $\{S_j : j \in N\}$  in product precedence constraints can be completely arbitrary. In particular, they can be overlapping and products can have circular dependencies on each other.

**Pricing with a Finite Price Menu:** Consider the case where the price of a product is a decision variable, rather than being fixed, and the preference weight of a product depends on its price. Increasing the price of a product is expected to

make it less desirable to customers, effectively decreasing its preference weight, but we make no assumptions on the correlation between prices and preference weights. The goal is to choose the prices of the products so as to maximize the expected revenue. We outline how to use Theorem 2.4.1 to solve this pricing problem.

To simplify exposition and notation we consider maximizing expected profit rather than expected revenue. We let  $K$  be the set of possible price levels for a product. The price corresponding to price level  $k$  for a product is given by  $r_k$ . Therefore,  $\{r_k : k \in K\}$  becomes the possible prices for a product. If we use the price level  $k$  for product  $j$ , then its preference weight is  $v_{jk}$ . Our notation indicates that the set of possible prices for each product is the same; it is straightforward to relax this. To capture our pricing decisions, we use  $x = \{x_{jk} : j \in N, k \in K\} \in \{0, 1\}^{|N| \times |K|}$ , where  $x_{jk} = 1$  if we set the price of product  $j$  at price level  $k$ , otherwise  $x_{jk} = 0$ . Thus, we want to solve the problem

$$\begin{aligned} \max \quad & \frac{\sum_{j \in N} \sum_{k \in K} r_k v_{jk} x_{jk}}{1 + \sum_{j \in N} \sum_{k \in K} v_{jk} x_{jk}} \\ \text{st} \quad & \sum_{k \in K} x_{jk} = 1 \quad \forall j \in N \\ & x_{jk} \in \{0, 1\} \quad \forall j \in N, k \in K, \end{aligned}$$

where the constraints ensure that each product is offered at one price level.

**Display Location Effects:** Consider the case where the preference weight of a product depends on where it is displayed. For example, in online retail customers may be more likely to choose products that are displayed at the top of search results, which can be captured by using preference weights that depend on the display order of the product. It turns out we can build on Theorem 2.4.1 to find the optimal set of products to offer when the preference weights depend on the display location.

We use  $v_{jl}$  to denote the preference weight of product  $j$  when this product is displayed at location  $l$ . Without loss of generality, we assume that there are as many possible locations as the number of products so that we can offer all products at once. In this case, we can index both the products and the locations by  $N$ ; this restriction is easily relaxed. To capture the product offer decisions, we use  $x = \{x_{jl} : j, l \in N\} \in \{0, 1\}^{|N| \times |N|}$ , where  $x_{jl} = 1$  if we offer product  $j$  in location  $l$ , otherwise  $x_{jl} = 0$ . If the product offer decisions are given by  $x$ , then we obtain an expected revenue of  $\sum_{j,l \in N} r_j v_{jl} x_{jl} / (1 + \sum_{j,l \in N} v_{jl} x_{jl})$ . Therefore, we are interested in solving the problem

$$\begin{aligned}
\max \quad & \frac{\sum_{j,l \in N} r_j v_{jl} x_{jl}}{1 + \sum_{j,l \in N} v_{jl} x_{jl}} \\
\text{st} \quad & \sum_{l \in N} x_{jl} \leq 1 \quad \forall j \in N \\
& \sum_{j \in N} x_{jl} \leq 1 \quad \forall l \in N \\
& x_{jl} \in \{0, 1\} \quad \forall j, l \in N,
\end{aligned}$$

where the first set of constraints ensure that each product is offered at most in one location and the second set of constraints ensure that each location is used by at most one product.

**General Constraints:** When the constraint matrix  $A$  is not TU then Theorem 2.4.1 no longer applies and problems (1.1) and (2.3) are not equivalent. However, we can still use (2.3) to generate useful solutions.

**Theorem 1.2.2.** *When the constraint matrix  $A$  is arbitrary then the objective value of (2.3) is an upper bound on the objective value of (1.1). Further, an optimal solution  $\bar{x}^*$  to (2.3) can be expressed as a convex combination  $\bar{x}^* = \alpha_1 \bar{x}_1^* + \dots + \alpha_l \bar{x}_l^*$  where each  $\bar{x}_i^*$  corresponds to an assortment.*

Theorem 1.2.2 is useful in two ways. First, because (2.3) provides an upper bound we can use it to test heuristics for (1.1); by comparing the solution value of a heuristic with the computed upper bound we can judge the quality of the heuristic algorithm. Second, the convex combination of assortments can be used in settings where the offered assortment changes quickly, in online retail for example. We can interpret  $\alpha_i$  as the proportion of time to offer assortment  $\bar{x}_i^*$  and, provided assortments change rapidly, can offer each assortment the appropriate amount of time.

### 1.3 Assortment Planning Under the Nested Logit Model

The MNL is simple, easy to estimate, and, with the above results, flexible in modeling practical business problems. However, due in part to its simplicity it has some undesirable features. Notable among these is the independence of irrelevant alternatives (IIA) property, which refers to the fact that if a product is added to the offered assortment, then the market share of all other offered products decreases by the same relative amount. Clearly, this property should not hold when products cannibalize each other to different extents; see [2]. The Nested Logit model (NL), introduced by [64], was designed to overcome the IIA property. Justifications and extensions for the nested logit model are provided in [39] and [4].

In the NL the set of products is decomposed into nests, or groups of correlated products. Given these nests customers make a two stage decision process: first a customer chooses a nest and then chooses a product within the selected nest. There are  $m$  nests indexed by  $M = \{1, \dots, m\}$ . Each nest contains  $n$  products indexed by  $N = \{1, \dots, n\}$ ; we can relax the assumption that each nest includes  $n$  products. Product  $ij$  refers to product  $j$  in nest  $i$ . Every product  $ij$  has a preference weight  $v_{ij}$ . We introduce two types of no purchase products: an “outer” no purchase

product, indexed by 0, that customers select when leaving in the nest selection stage and an “inner” no purchase product, indexed by  $i0$ , that customers select when leaving nest  $i$ .

We denote an assortment by a vector  $S = (S_1, \dots, S_m)$  where  $S_i \subseteq N$ . We let  $V_i(S) = v_{i0} + \sum_{j \in S_i} v_{ij}$ . If a customer selects nest  $i$  in the nest selection stage then the probability a customer purchases product  $j \in S_i$  is given by

$$P_{ij}(S_i) = \frac{v_{ij}}{v_{i0} + \sum_{k \in S_i} v_{ik}} = \frac{v_{ij}}{V_i(S_i)}$$

In addition to preference weights there is also a nest dissimilarity parameter  $\gamma_i$  associated with every nest  $i$ . In our nested logit model, the dissimilarity parameters  $(\gamma_1, \dots, \gamma_m)$  are assumed to be constants, as described in [40]. The probability a customer chooses nest  $i$  when  $(S_1, \dots, S_m)$  is offered is given by

$$Q_i(S_1, \dots, S_m) = \frac{V_i(S_i)^{\gamma_i}}{v_0 + \sum_{l \in M} V_l(S_l)^{\gamma_l}}$$

Note that higher values of  $\gamma_i$  magnify the total preference weight of products available in  $S_i$ .

The standard NL model studied in [39] restricts  $\gamma_i \leq 1$  and  $v_{i0} = 0$  for all  $i \in M$ . In what follows we focus on the standard NL model but also consider non-standard variants that relax these restrictions.

If we offer the assortment  $S_i$  in nest  $i$ , then we can write the expected revenue we obtain from this nest as

$$R_i(S_i) = \sum_{j \in S_i} r_{ij} P_{ij}(S_i) = \frac{\sum_{j \in S_i} r_{ij} v_{ij}}{V_i(S_i)},$$

with the interpretation that  $R_i(\emptyset) = 0$ . Therefore, if we offer the assortment  $(S_1, \dots, S_m)$  over all nests with  $S_i \subset N$  for all  $i \in M$ , then we obtain an expected revenue of

$$\Pi(S_1, \dots, S_m) = \sum_{i \in M} Q_i(S_1, \dots, S_m) R_i(S_i).$$

Our goal is to choose an assortment  $(S_1, \dots, S_m)$  that maximizes the expected revenue over all nests, yielding the assortment planning problem

$$Z^* = \max_{(S_1, \dots, S_m): S_i \subset N, i \in M} \Pi(S_1, \dots, S_m). \quad (1.3)$$

### 1.3.1 Motivation and Literature Review

Like the MNL, the NL is well studied in the literature. There are known methods to quickly estimate NL parameters; see [59]. Furthermore, it is possible to show that the nested logit model is compatible with a random utility-based choice model. This feature gives some behavioral justification to the NL; see [4]. Finally, the nested logit model allows correlations between the utilities of the products, capturing the fact that the way a customer evaluates a certain product may help us predict how this customer would evaluate other similar products.

This makes the NL an attractive model, especially for behavioral descriptions where only estimating model parameters is a concern. However, there is limited work on optimization routines involving the NL; this prevents it from being part of operational decision making systems. [30] consider joint assortment planning and pricing problems under the NL. [11] study assortment problems under the NL without any constraints, and [49] consider constraints under the NL. Most of this work, however, considers NL models with a small number of nests.

In what follows we fill this gap. We give tractable algorithms that, given the parameters of the NL, provide assortments that give high revenue. These algorithms scale polynomially in the number of nests. Chief among these results is that under the standard NL model we can find an assortment that achieves the optimal revenue.

The standard NL model can be limiting in many ways, though. Using dissimilarity parameters that take on values larger than one allows us to model synergistic

effects among products within the same nest. In particular if we offer the assortment  $(S_1, \dots, S_m)$ , then the probability that a customer purchases product  $j \in S_i$  in nest  $i$  is given by

$$Q_i(S_1, \dots, S_m) P_{ij}(S_i) = \frac{V_i(S_i)^{\gamma_i-1}}{v_0 + \sum_{l \in M} V_l(S_l)^{\gamma_l}} v_{ij}.$$

From the expression above, we observe that when  $\gamma_i \leq 1$ , adding a product  $k \notin S_i$  to nest  $i$  always decreases the purchase probability of product  $j \in S_i$ . In practice, this is not always the case. We observe that if  $\gamma_i > 1$  and we add a new product to nest  $i$ , then both  $V_i(S_i)^{\gamma_i-1}$  and  $\sum_{l \in M} V_l(S_l)^{\gamma_l}$  increase in the expression above. As a result, the probability that a customer purchases product  $j$  may increase or decrease. This feature allows us to model synergies between different products in a nest. These synergies may provide justification for including loss leaders in a nest in order to attract traffic.

There are a number of empirical studies that fit the NL to customer choice data and estimate the nest dissimilarity parameters as exceeding one; see [61], [32], [58] and [66]. [4] revisits the compatibility of the NL with utility maximization principle and shows that even if we do have  $\gamma_i > 1$  for some  $i \in M$  the NL can still be compatible with a random utility-based choice model. [61] interpret a nest dissimilarity parameter exceeding one as a signal that substitution across nests happens more readily than substitution within a nest.

Additionally, in many settings relaxing the constraint that  $v_{i0} = 0$  for all  $i \in M$  is practically important. When  $v_{i0} = 0$  this corresponds to a situation where if a customer decides to make a purchase within a particular nest, then the customer always makes a purchase within the selected nest; the nests are *fully-captured nests*. On the other hand, if the preference weight of the no purchase option within a nest is strictly positive, then a customer may leave without purchasing anything in the selected nest; the nests are *partially-captured nests*. There are a number of reasons

to consider the case with partially-captured nests. Allowing  $v_{i0} > 0$  gives the retailer freedom to model competition:  $v_{i0}$  can be interpreted as the preference weight of products offered by the retailers competition. Our use of the partially-captured nests is closely related to more general version of the NL, the cross nested logit model; see [13].

### 1.3.2 Main Results

We classify the instances of the assortment planning problem in (3.2) along two dimensions. The first dimension is based on the values of the dissimilarity parameters  $(\gamma_1, \dots, \gamma_m)$  of the nests. Along this dimension, we separately consider the two cases where (i) we have  $\gamma_i \leq 1$  for all  $i \in M$ , and (ii) there are no restrictions on the dissimilarity parameters. The second dimension of classification is based on the values of the preference weights  $(v_{10}, \dots, v_{m0})$  of the no purchase options within the nests. Along this dimension, we separately consider the two cases where (i) we have  $v_{i0} = 0$  for all  $i \in M$ , and (ii) there are no restrictions on the preference weights of the no purchase options. Since there are two cases to consider along each one of the two dimensions, we study the assortment planning problem in (3.2) under four cases. The standard version of the NL model, studied in [38] and [40], has  $\gamma_i \leq 1$  and  $v_{i0} = 0$  for all  $i \in M$ . In these papers, the author notes that having  $\gamma_i \leq 1$  for all  $i \in M$  implies that the nested logit model is always compatible with a random utility-based choice model, irrespective of the values of the preference weights of the products and the preference weights of the no purchase options

All of the results we present depend on a linear program that can recover optimal solutions to (3.2).

**Theorem 1.3.1.** *Suppose that there is some collection  $\mathcal{T}_1, \dots, \mathcal{T}_m$  such that for an optimal solution  $S^* = (S_1^*, \dots, S_m^*)$  to (3.2) we have  $S_i^* \in \mathcal{T}_i$  for all  $i$ . Then  $Z^*$  is*



the optimal objective value to

$$\begin{aligned}
\min \quad & x & (1.4) \\
\text{s.t.} \quad & v_0 x \geq \sum_{i \in M} y_i \\
& y_i \geq V_i(S_i)^{\gamma_i} (R_i(S_i) - x) \quad \forall i \in M, S_i \in \mathcal{T}_i.
\end{aligned}$$

Further, the optimal solution  $S^*$  can be recovered from a solution to (3.4).

The decision variable  $x$  in (3.4) is the total expected revenue and  $y_i$  is the expected revenue we receive given a customer selects nest  $i$ . The program has a total of  $1 + m$  variables but  $1 + \sum_i |\mathcal{T}_i|$  constraints. In order to recover an optimal solution in a tractable fashion we need to find  $\mathcal{T}_i$  that are both small and guaranteed to contain  $S_i^*$ .

**Standard NL:** One of our main results is that the assortment problem under the standard version of the NL admits a tractable algorithm. To achieve this result we use (3.4) and a characterization of optimal solutions:

**Theorem 1.3.2.** *Suppose we are under the standard NL. If  $S^* = (S_1^*, \dots, S_m^*)$  is an optimal solution to (3.2) then each  $S_i^*$  is revenue ordered. That is, if  $ij \in S_i^*$  and  $r_{ij'} > r_{ij}$  then  $ij' \in S_i^*$ .*

This theorem lets us construct the  $\mathcal{T}_i$  necessary in (3.4): we let  $\mathcal{T}_i$  consist of all revenue ordered sets of products in nest  $i$ . By Theorem 1.3.2 the resulting  $\mathcal{T}_i$  will contain  $S_i^*$  and, because of the structure,  $|\mathcal{T}_i| \leq n + 1$  allowing (3.4) to be solved in polynomial time.

**Non-Standard Variants of NL:** While the assortment planning problem in (3.2) is polynomially solvable when  $\gamma_i \leq 1$  and  $v_{i0} = 0$  for all  $i \in M$ , we show lifting any one of these restrictions renders the problem NP-hard and we resort to approximation methods. For these cases we take a two pronged approach.

For each of the NP-hard cases we develop tractable algorithms that run very quickly and return a solution with a provable worst-case guarantee. If an algorithm has a worst case guarantee of  $\alpha$  then, on all problem instances, it is guaranteed to return a solution with objective value at least  $1/\alpha$  of the optimal solution. These algorithms make use of a more general version of Theorem 1.3.1. This more general version shows that if the  $\mathcal{T}_i$  of Theorem 1.3.1 are guaranteed to contain a solution with a worst-case guarantee of  $\alpha$  to problem (3.2) then (3.4) will recover a solution with worst case guarantee at least  $\alpha$ . We summarize these worst-case guarantees in Figure 3.1.

In addition to these worst case guarantee algorithms we also develop three alternative algorithmic approaches that apply to the most general forms of the NL that we consider. First, we develop a pseudopolynomial time algorithm, an algorithm in which the running time depends polynomially on a unary encoding of the values of the input parameters. Second, we develop an approximation scheme that can find solutions of arbitrary accuracy at the expense of expending more computation time. Finally, we develop an easily computed upper bound on the optimal revenue.

The upper bound is of special practical importance. It can be used to benchmark the quality of heuristic solution approaches; by computing the quality of the solution returned by the heuristic and comparing it with our upper bound a practitioner can determine the quality of their algorithms. This is especially important as the size of the problem grows and heuristics become more appealing for practical revenue management systems.

The upper bound also allows us to test the quality of our worst-case guarantee algorithms experimentally. There are two values that drive problem difficulty: the ratio between smallest and largest preference weight,  $\kappa$ , and the magnitude of the

		NEST BEHAVIOR	
		Fully Captured	Partially Captured
Dissimilarity	$0 \leq \gamma \leq 1$	Polynomially-solvable	Worst-case guarantee of 2
	Free	Worst-case guarantee of $\min\{\rho, 2\kappa\}$	Worst-case guarantee of $2\kappa$ , pseudo-poly-time algorithm, approximation scheme, upper bound

Table 1.1: Summary of results.  $\rho$  is the ratio of largest to smallest revenue.  $\kappa$  is the ratio of largest to smallest preference weight.

largest  $\gamma_i$ ,  $\hat{\gamma}$ . In the hardest instances, where  $\kappa$  and  $\hat{\gamma}$  are very high, the average gap between our upper bound and solution is only 0.29% and the maximum gap is 3.26%. In easier instances, where  $\kappa$  is high but  $\bar{\gamma}$  is low, the average gap is 0.19% and the maximum gap is 2.42%. When both  $\kappa$  and  $\bar{\gamma}$  are low the performance is very good; our algorithms achieve an average gap of 0.01% and a maximum gap of 0.20%. These experiments give strong evidence that our algorithms will perform very well in practice.

## 1.4 Quality Consistent Pricing Under the Nested Logit Model

In this section, we consider pricing problems when there is a quality consistency constraint and customers choose according to the NL. In the quality consistency constraint there is an intrinsic ordering of the products according to quality and the retailer is constrained to offer higher prices for higher quality products. The goal is to find the prices to charge that maximize the expected revenue, while making sure that the prices satisfy the quality consistency constraint.

Recalling the variations of the NL model introduced previously, in this section

we work exclusively under the standard version of the NL model where  $\gamma_i \leq 1$  and  $v_{i0} = 0$  for all  $i \in M$ . We index the nests by  $M = \{1, \dots, m\}$  and in each nest there are  $n$  products indexed by  $N = \{1, \dots, n\}$ . For each product, there are  $q$  possible prices given by  $\Theta = \{\theta^1, \dots, \theta^q\}$ . Without loss of generality, we index the possible prices so that  $0 < \theta^1 < \theta^2 < \dots < \theta^q$ . We use  $p_{ij} \in \Theta$  to denote the price that we charge for product  $j$  in nest  $i$ . If we charge price  $p_{ij}$  then the preference weight of this product is given by  $v_{ij}(p_{ij})$ . We impose the mild assumption that if we charge a larger price for a product, then its preference weight becomes smaller, implying that  $v_{ij}(\theta^1) > v_{ij}(\theta^2) > \dots > v_{ij}(\theta^q) > 0$ . Although our notation implies that the number of products in each nest is the same and the set of possible prices that we can charge for each product is the same this assumption is easily relaxed.

We use  $p_i = (p_{i1}, \dots, p_{in}) \in \Theta^n$  to capture the price vector charged in nest  $i$ . As a function of  $p_i$  we use  $V_i(p_i)$  to denote the total preference weight of the products in nest  $i$ , so that  $V_i(p_i) = \sum_{j \in N} v_{ij}(p_{ij})$ . Under the NL, if we charge  $p_i$  in nest  $i$ , then a customer that has already decided to make a purchase in nest  $i$  chooses product  $j$  in this nest with probability  $v_{ij}(p_{ij})/V_i(p_i)$ . In this case, if a customer has already decided to make a purchase in this nest, then the expected revenue obtained from this customer is given by

$$R_i(p_i) = \sum_{j \in N} p_{ij} \frac{v_{ij}(p_{ij})}{V_i(p_i)} = \frac{\sum_{j \in N} p_{ij} v_{ij}(p_{ij})}{V_i(p_i)}.$$

If we charge the price vectors  $(p_1, \dots, p_m) \in \Theta^{m \times n}$  over all nests, then a customer decides to make a purchase in nest  $i$  with probability  $Q_i(p_1, \dots, p_m) = V_i(p_i)^{\gamma_i} / (v_0 + \sum_{l \in M} V_l(p_l)^{\gamma_l})$ . If the customer decides to make a purchase in nest  $i$ , then the expected revenue obtained from this customer is  $R_i(p_i)$ . Thus, if we charge the price vectors  $(p_1, \dots, p_m)$  over all nests, then the expected revenue from

a customer is given by

$$\Pi(p_1, \dots, p_m) = \sum_{i \in M} Q_i(p_1, \dots, p_m) R_i(p_i) = \frac{\sum_{i \in M} V_i(p_i)^{\gamma_i} R_i(p_i)}{v_0 + \sum_{i \in M} V_i(p_i)^{\gamma_i}}.$$

Our goal is to find the price vectors  $(p_1, \dots, p_m)$  to charge over all nests to maximize the expected revenue above subject to the constraint that the price vector charged in each nest satisfies a quality consistency constraint.

We consider two types of quality consistency constraints. In *price ladders inside nests* there is an intrinsic ordering between the qualities of the products in each nest. There is no dictated ordering between the qualities or prices of the products in different nest. This type of quality consistency constraint becomes relevant when, for example, nests correspond to products offered by different brands. There is a natural quality ordering among products a particular brand offers but it is difficult to compare the products across brands.

In *price ladders between nests* there is an intrinsic ordering between the qualities of the nests, but there is no clear ordering between the qualities of the products in the same nest. This type of quality consistency constraint becomes relevant when the nests correspond to different quality levels and the products within a particular nest correspond to products that differ in cosmetic or personal features.

Each type of constraint induces a set of feasible prices  $\mathcal{F}$  that can be offered. Thus, we are interested in solving

$$z^* = \max_{(p_1, \dots, p_m) \in \mathcal{F}} \left\{ \Pi(p_1, \dots, p_m) \right\}. \quad (1.5)$$

In the problem above, the price of each product takes values in the discrete set  $\Theta$ . Furthermore, the objective function depends on the prices of the products in a nonlinear fashion. Thus, this problem is a nonlinear combinatorial optimization problem.

### 1.4.1 Motivation and Literature Review

There is a significant amount of work on solving pricing problems under variants of the MNL and NL. Most of this work assumes a parametric relationship between preference weights and prices. Under the MNL there is considerable literature of this flavor. [25], [55], and [12] consider pricing problem under the MNL where preference weights are a parametric function of price. [8] and [63] give tractable solution methods for joint assortment planning and pricing problems under the MNL, again where preference weights depend parametrically on price. The literature on solving pricing problems under the NL has recently started growing. [34] study pricing problems under the assumption that the products in the same nest have the same price sensitivity; [21] relax this assumption and [47] develop further algorithms. [35] and [33] consider pricing problems under the NL where the choice process proceeds in more than two stages.

Our work on pricing diverges from the majority of these results. We do not assume a parametric relationship between preference weight and price. Rather, we work within a discrete pricing framework where there is a finite set of discrete price levels a retailer can post for each product. There is some precedent for this in the literature. Under the MNL [29] study pricing problems, where the attractiveness of a product depends on its price in a general fashion and there are constraints on the expected number of sales for the products. Under the NL model [20] consider the joint discrete pricing and assortment problem.

The discrete pricing approach provides two advantages. First, preference weights can depend on price in an arbitrary way. This is especially important in settings where pricing competition is a dominating consideration: the preference weight of a product should, intuitively, increase sharply when a retailer offers a price lower than one of its competitors. Such sharp increases are not captured in the para-

metric models that are studied. Additionally, by introducing discrete price levels the retailer is implicitly constrained to offer natural, attractive prices, like those ending in .99. In parametric models the retailer has a continuum of prices available to them.

The pure pricing problem is not immediately useful in practice. There are frequently pricing controls in place that limit the range of prices that can be posted for a product. In the literature these pricing controls are usually modeled with price bounds, upper and lower bounds on the prices for a product; see [21] and [47]. While price bounds are very useful they are often not enough. We augment price bounds with a quality consistency constraint.

In many settings the quality consistency constraint is essential. When prices fail to be quality consistent customers lose confidence in the retailer. Such prices also convey a sense of fairness to customers; see [19]. Quality consistency constraints have been introduced by [48]; their work is motivated by a pricing problem in the automobile industry, where the prices of the automobiles with richer features should be larger.

We work with quality consistent pricing problems under both the MNL and NL models. Under the MNL model our approach leverages Theorem 2.4.1. Applying this theorem to the quality consistent pricing problem follows the same ideas as the joint assortment and pricing problem outlined in Section 2. Since the NL model is a strict generalization of the MNL model we focus on that in the remainder of the section. It is important to note, however, that while we place mild restrictions on the relationship between price and preference weight in the NL model, this relationship can be completely arbitrary in the MNL model.

## 1.4.2 Main Results

As mentioned above we can use Theorem 2.4.1 to solve the quality consistent pricing problem under the MNL. This gives a linear programming approach that is easily implemented and allows us to capture arbitrary relationships between price and preference weight. Below we outline the results related to the NL.

**Quality Consistent Pricing Within Nests:** Our strategy for solving quality consistent pricing problems under the NL closely follows the strategy implied by Theorem 1.3.1. In this context, however, we use a variation of Theorem 1.3.1:

**Theorem 1.4.1.** *Suppose that there is some collection  $\mathcal{P}_1, \dots, \mathcal{P}_m$  such that for some optimal price vector  $p^* = (p_1^*, \dots, p_m^*)$  for (1.5) we have  $p_i^* \in \mathcal{P}_i$  for all  $i$ . Then the optimal objective value to (1.5) is achieved by*

$$\min \left\{ x : v_0 x \geq \sum_{i \in M} y_i, y_i \geq V_i(p_i)^{\gamma_i} (R_i(p_i) - x) \quad \forall p_i \in \mathcal{P}_i, i \in M \right\}, \quad (1.6)$$

In the quality consistent pricing within nests the constraints naturally decompose across nests and so finding the  $\mathcal{P}_i$  of Theorem 1.4.1 is a natural approach. Our strategy is to find  $\mathcal{P}_i$  that are both small and guaranteed to contain  $p_i^*$ , so that (1.6) can be solved efficiently.

To generate  $\mathcal{P}_i$  for a nest  $i \in M$  we expand on a theorem of [20]

**Theorem 1.4.2.**  *$p_i^*$  is the optimal solution to*

$$\max_{p_i \in \mathcal{P}_i} \{V_i(p_i)(R_i(p_i) - u_i^*)\}$$

where  $u^*$  is a function of the optimal solution  $z^*$ .

Because  $u^*$  is a function of  $z^*$  we do not know  $u^*$  a priori and can not immediately use Theorem 1.4.2. However, we make two observations that allow us to leverage it. First, we observe that for fixed  $u^*$  we can find the optimal quality



consistent prices by solving a longest path problem in a specific network. Second, we note that over all values of  $u^*$  there are a small number of unique longest paths in this network. We identify the set of longest paths over all values of  $u^*$  as the  $\mathcal{P}_i$  of Theorem 1.4.1; the set of longest paths over *all* values of  $u^*$  is small and guaranteed to contain the optimal price vector  $p_i^*$ . We can solve the longest path problem with a linear program and, because over all values of  $u^*$  there are a small number of unique longest paths, the parametric simplex method will find all such paths in a tractable fashion.

Ultimately these techniques lead to tractable algorithms that involve using the parametric simplex method to generate candidate price vectors  $\mathcal{P}_i$  and then solving the linear program of Theorem 1.4.1. This algorithm is easily implemented and runs very quickly in practice: with 6 nests, 30 products, and 30 price levels the algorithm takes only a few seconds on laptop hardware.

**Quality Consistent Pricing Across Nests:** Quality consistent pricing across nests is significantly harder than quality consistent pricing within nests. This is because the quality consistency constraint does not decompose by nests and, as a result, Theorem 1.4.1 is not applicable.

Instead, we use a different solution approach based on a dynamic program. In this dynamic program the decision epochs are the nests. When making the decision for nest  $i$  there are two state variables: the largest price charged in nest  $i - 1$  and the largest price charged in nest  $i$ . These state variables determine the range of prices that can be charged in nest  $i$ ; the largest price charged in nest  $i - 1$  and largest price charged in nest  $i$  are, respectively, a lower and upper bound on the prices that can be charged in nest  $i$ . The dynamic program glues together the prices across nests in such a way that the total price vector will be quality consistent.

This dynamic program decomposes the quality consistent pricing problem across nests into a new type of subproblem: given an upper and lower bound on the prices that can be offered within a nest, which prices are optimal? We must solve the problem for each nest, and for each upper and lower bound; this is a polynomial number of subproblems.

To find the optimal prices in nest  $i$  for an upper and lower bound we take much the same approach as we did for quality consistency within nests. We use Theorem 1.4.2 to generate a set of candidate price vectors and, among these prices, choose the optimal one. Now, however, the candidate price vectors satisfy an upper and lower bound instead of a quality consistency constraint.

This approach yields tractable algorithms for quality consistent pricing across nests. Though it is somewhat more complicated than our algorithms for quality consistent pricing within nests this algorithm is still easily implemented and runs very quickly in practice: with 6 nests, 30 products, and 30 price levels the algorithm takes only a few minutes on laptop hardware.

## 1.5 Assortment Planning Over Time

In this chapter we introduce the problem of assortment planning over time. In contrast to the results above, in this chapter we do not focus on a specific choice model. Instead we are interested in the impact of introducing a new dynamic into an arbitrary choice model. Specifically, we are interested in introducing a notion of time. We have a sequence of time periods and can only introduce one new product per time step. We are not allowed to remove products from our assortment that have already been introduced. In each time period the customers choose among the currently offered products according to a specified choice model. The goal is to determine which products to introduce, and in what order, so as to maximize

the total revenue realized over all the time steps under some choice model.

Recalling the notation from the assortment planning problem we let  $N$  be the set of products that can be offered for sale, and let  $n = |N|$ . We let  $r_j$  be the revenue of product  $j$  and let  $P_j(S)$  be the probability that product  $j$  is purchased if  $S \subseteq N$  is offered for sale; then  $P_j(S) = 0$  if  $j \notin S$ . We do not consider any specific choice model. Instead we consider choice models where the following two properties hold. First,

$$P_j(S) \geq P_j(T) \quad \forall T \forall j \in S \subset T;$$

that is, the probability of purchasing product  $j$  cannot increase if we offer a larger set of products. This holds for the MNL and NL model, for example, as well as any choice model based on utility maximization. Second,  $\sum_{j \in S} P_j(S) \leq 1$  for any non-empty set of products  $S$ . If an assortment  $S$  is offered for sale, then with probability  $1 - \sum_{j \in S} P_j(S)$ , no product is purchased. The expected revenue for a set  $S$  of products is  $R(S) = \sum_{j \in S} r_j P_j(S)$ . For lack of a better term, let us call such choice models *monotone* choice models.

Additionally, we consider the case in which we only know that the revenue function  $R(S)$  is *monotone* (that is,  $R(S) \leq R(T)$  for any  $S \subseteq T \subseteq N$ ) and *submodular* (that is, for any  $j \notin S \subseteq T \subseteq N$ , then  $R(S \cup \{j\}) - R(S) \geq R(T \cup \{j\}) - R(T)$ ), without knowing anything about the underlying choice model.

We wish to study the assortment planning problem over time. Intuitively, we would like to find a sequence of products in which we can offer at most one new product for sale at each of  $T$  time steps that maximizes the overall expected revenue achieved over the given time horizon. Once a product is offered for sale, it remains available to purchase for the remainder of the time horizon. More precisely, we would like to find sets  $S_1, S_2, \dots, S_T$  such that  $|S_t| \leq t$  for all  $1 \leq t \leq T$  and  $S_1 \subseteq S_2 \subseteq \dots \subseteq S_T$  that maximizes  $\sum_{t=1}^T R(S_t)$ .

### 1.5.1 Motivation and Literature Review

The assortment planning over time problem arises naturally in a setting where products must be incrementally introduced. For example, when a firm introduces a product line they often introduce individual products months apart from one another.

There is substantial literature related to building and adjusting assortments of products over time. In dynamic assortment problems, the offered assortment is adjusted over time, possibly due to depleted product inventories, better understanding of customer choice processes or changes in customer tastes. [28] and [37] study the problem of finding an assortment to offer and the corresponding stocking quantities with the understanding that customers choose only among the products that are still in stock. [3] and [24] consider the problem of dynamically customizing the assortment offerings based on the preferences of each customer and remaining product inventories. [6] and [9] study assortment problems where the attractiveness of the products diminishes over time and they seek optimal policies to replace such products. [7] and [62] develop models where the assortment offering needs to be adjusted over time in response to a better understanding of the customer choice process.

There is also a less direct relationship between assortment planning over time and online retail. Often in online retail settings products are displayed to customers as a list of products on a web page. A customer will view the list of products but, because of limited patience, the customer may not view all of the products in the list. Instead the customer will view the the list up to some point and then not view the rest. In this way we can consider a customer building the assortment they view incrementally: they continue to view products in the list until they run out of patience. After they exhaust their patience they choose a product

from the assortment they have seen according to a choice model. The problem introduced above corresponds to precisely this scenario when customers have a uniform patience level from 1 to  $T$ .

### 1.5.2 Main Results

We provide two main results for the incremental problem. First, for monotone choice models we relate the incremental assortment planning problem to the capacitated assortment planning problem. In the capacitated assortment planning problem there is a capacity  $c$  on the number of products that can be offered for sale and we wish to find the set  $S^* = \operatorname{argmax}_{S \in \mathcal{F}} r_j P_j(S)$ , where  $\mathcal{F} = \{S : |S| \leq c\}$ . Given an algorithm for the capacitated assortment problem we can provide an algorithm for the assortment over time problem that achieves at least  $1/2$  of the optimal revenue. This result extends to the case when we only have an algorithm that achieves an approximate solution for the capacitated problem.

Second, for choice models where the revenue function is monotone we show that the assortment problem over time is NP-hard. This NP-hardness result also applies to the case when we have a monotone choice model. In addition to the NP-hardness result we also provide a  $(1 - 1/e)$ -approximation algorithm for the assortment problem over time when we know the revenue function is monotone and submodular. This algorithm uses a well-known greedy  $(1 - 1/e)$ -approximation algorithm for finding a maximum valued set  $S$  with  $|S| \leq c$  for any monotone, submodular set function due to [45].

**CONSTRAINED ASSORTMENT OPTIMIZATION UNDER THE  
MULTINOMIAL LOGIT MODEL**

## 2.1 Introduction

In this chapter we consider constrained assortment optimization problems assuming customers choose according to the multinomial logit model (MNL). The objective is to maximize the expected revenue obtained from each customer and the constraints on the offered assortment can be captured by a totally unimodular constraint matrix. We formulate this constrained assortment problem as a fractional program with binary decision variables to model the inclusion of products in the assortment. Our main result shows that we can transform this fractional program into a linear program where the integrality constraints can be relaxed because of the totally unimodular nature of the constraints. This result allows us to formulate and solve a variety of practical assortment and pricing problems, the majority of which were not known to be tractable in the literature.

Let  $N$  be the product consideration set from which we want to select an assortment  $S \subset N$  to offer to customers. To facilitate the discussion we identify an assortment with an incidence vector  $x = \{x_j : j \in N\} \in \{0, 1\}^{|N|}$ , where  $x_j = 1$  if  $j \in S$  and  $x_j = 0$  if  $j \notin S$ . The set of feasible assortments is given by  $\mathcal{F} = \{x \in \{0, 1\}^{|N|} : \sum_{j \in N} a_{ij} x_j \leq b_i \forall i \in M\}$  for a totally unimodular matrix  $[a_{ij}]_{i \in M, j \in N}$ . Given feasible product offer decisions  $x \in \mathcal{F}$ , each customer chooses among the offered products according to MNL. The objective is to choose  $x \in \mathcal{F}$ , a feasible set of products to offer, to maximize the expected revenue obtained from each customer. Although this assortment problem is a fractional program with binary decision variables, our main result shows that it can directly be solved as a

linear program.

Building on our main result, we show how to solve five classes of practical assortment and pricing problems. First, we work with assortment problems under MNL where there are cardinality constraints on the assortment. [50] consider assortment problems with a limit on the total number of offered products. Their approach generates candidate assortments and checks the performance of the candidates, whereas we give a direct linear programming formulation. Our approach also extends to more general cardinality constraints. In particular, our results apply when the set of products are partitioned into a number of subsets and we limit the number of products offered in each subset. Such constraints occur when, for example, television sets are partitioned as small, medium and large and we limit the number of offered television sets of each size. We can also handle overlaps between successive subsets. That is, if some television sets can be considered as both small and medium and some can be considered as both medium and large, then we can limit the numbers of offered small, medium and large television sets. We can deal with nested cardinality constraints as well, which occur when products are categorized into subsets such that one subset includes another one and we limit the number of products offered in each subset. For example, some products may be specialty products and we may want to offer at most  $k$  specialty products, while limiting the number of all offered products to  $\ell$ . Such overlapping or nested cardinality constraints are not studied in the earlier work.

Second, we consider assortment problems with display location effects, where the attractiveness of a product depends not only on its own attributes but also on the location at which it is displayed. Such problems arise when products are displayed in a window or a shelf and the products with prime locations may have a better chance of attracting attention, which can be modeled by a higher attrac-

tiveness parameter in MNL. Another relevant setting is online retail, where the propensity of a customer to choose a product depends not only on the attributes of the product, but also on where the product is shown within search results or web site. We show that the assortment optimization problem with display specific attractiveness parameters can be formulated as a linear program. To our knowledge, tractability of the assortment problem with display location effects was not previously known.

Third, we consider pricing problems under MNL, when there are a finite number of possible price levels for the products and the attractiveness of a product depends on its price. The objective is to choose the product prices to maximize the expected revenue from each customer. We show how to obtain an optimal solution to this problem by using a linear program. Earlier pricing formulations assume that the attractiveness parameter of product  $j$  is a parametric function  $e^{\alpha_j - \beta_j p}$  of the price  $p$  of this product, where  $\alpha_j$  and  $\beta_j$  are fixed coefficients, in which case, the pricing problem can be formulated as a continuous optimization problem involving the product prices. In our approach, the attractiveness of a product depends on its price arbitrarily, not limited to the parametric form  $e^{\alpha_j - \beta_j p}$ . Also, since we work with a finite number of possible price levels, we can put explicit restrictions on the product prices. For example, we can limit ourselves to prices in increments of a dollar or make sure that the prices of the products are chosen within specific intervals.

Fourth, we consider quality consistent pricing problems. In such pricing problems, there is an inherent ordering between the products, where some products are considered lower in quality than others. The prices should be chosen to be quality consistent, so that lower quality products are priced lower than higher quality products. We show how to obtain the optimal solution to such quality consistent



pricing problems. The quality consistent pricing terminology is introduced by [19] in the context of MNL, but to our knowledge, there is no work on finding optimal prices under quality consistency constraints when customers choose according to MNL. Quality consistency constraints are especially important when there is excessive demand for lower quality products, in which case, one might price lower quality products higher than higher quality products when quality consistency constraints are ignored.

Finally, we consider assortment problems with product precedence constraints where a product cannot be offered unless a certain set of related products are offered. We give a linear programming formulation of the assortment problem under such precedence constraints.

## 2.2 Literature Review

As mentioned in the introduction the MNL model possesses the independence of irrelevant alternatives property, which refers to the fact that if a product is added to the offered assortment, then the market share of all other offered products decreases by the same relative amount. Clearly, this property should not hold when products cannibalize each other to different extents; see [2]. Our linear programming formulation under MNL extends to a more general attraction model that mitigates some of the shortcomings of MNL. Unlike MNL, the purchase probabilities in the general attraction model depend on all of the products, both those offered and those not offered. This model was proposed by [22] and involves a shadow attraction value for each product that impinges on the choice probabilities when the products are not offered. It turns out that a simple transformation of our formulation for MNL allows for the assortment problem to be solved under this more general attraction model.

Assortment problems under variants of MNL is closely related to our work. [57] study assortment problems under MNL without any constraints and show that the optimal assortment can be obtained by greedily adding products into the offered assortment in the order of decreasing revenues. [50] consider cardinality constraints on the offered assortment, whereas [49] assume that each product has a space requirement and they limit the total space requirement of the offered assortment. [22] give a linear programming formulation for the assortment problem under MNL without considering constraints on feasible assortments and extend their formulation to the general attraction model described in the previous paragraph. [5], [43] and [52] consider the case where there are multiple customer types and customers of each type choose according to a different MNL. They provide heuristics, integer programming formulations and approximation methods. [11] study assortment problems under the nested logit model without any constraints, whereas [49] and [20] consider constraints under the nested logit model. [18], [36], [31], [67], [56], [41] and [42] use assortment problems to make extensions to network revenue management, where itinerary products consume the capacities on flight legs in bundles.

For pricing problems, [25] observe that the expected revenue is not a concave function of prices when customers choose according to MNL, but [55] and [12] are able to recover a concave objective function by using the market share of each product as the decision variable. [34] solve pricing problems under the nested logit model. They also use market shares as decision variables, but work under the assumption that the products in the same nest share the same price sensitivity. [21] study the same problem, but they relax the assumption that the price sensitivities in a nest are the same. [8] and [63] study joint assortment and pricing problems, where one chooses the products to offer and their corresponding prices.

## 2.3 Problem Formulation

The set of products is  $N$ . The revenue and the preference weight associated with product  $j$  are respectively  $r_j$  and  $v_j$ . To capture the product offer decisions, we define the decision variable  $x_j \in \{0, 1\}$  such that  $x_j = 1$  if we offer product  $j$ , otherwise  $x_j = 0$ . Under MNL, if the product offer decisions are given by the vector  $x = \{x_j : j \in N\} = \{0, 1\}^{|N|}$ , then a customer purchases product  $j$  with probability  $P_j(x) = v_j x_j / (1 + \sum_{k \in N} v_k x_k)$ , where we normalize the preference weight of the no purchase option to one. Therefore, if the products that we offer correspond to the vector  $x$ , then the expected revenue obtained from a customer can be written as

$$R(x) = \sum_{j \in N} r_j P_j(x) = \frac{\sum_{j \in N} r_j v_j x_j}{1 + \sum_{j \in N} v_j x_j}. \quad (2.1)$$

Without loss of generality, if selling a unit of product  $j$  involves a constant cost for each unit sold, then we assume that  $r_j$  is the profit from the sale, given by the difference between revenue and cost. For a totally unimodular matrix  $A = [a_{ij}]_{i \in M, j \in N}$  with dimensions  $|M| \times |N|$ , the feasible set of product offer decisions are given by  $\mathcal{F} = \{x \in \{0, 1\}^{|N|} : \sum_{j \in N} a_{ij} x_j \leq b_i \forall i \in M\}$ . For the moment, we do not go into the specific structure of the matrix  $A$ , but we give specific examples for  $A$  in the next section. Our goal is to find a set of feasible products to offer so as to maximize the expected revenue obtained from each customer, yielding the problem

$$z^* = \max_{x \in \mathcal{F}} R(x). \quad (2.2)$$

Our use of “less than” constraints to capture the feasible set of product offer decisions is without loss of generality. A “greater than” constraint can be multiplied by  $-1$  to get a “less than” constraint and an equality constraint can be replaced a pair of “less than” and “greater than” constraints. Recalling that multiplying

a row of a matrix by  $-1$  or duplicating a row of a matrix does not change its total unimodularity properties, we end up with “less than” constraints and a totally unimodular constraint matrix after transforming equality and “greater than” constraints to “less than” constraints; see Proposition 2.1 in Chapter III.1 of [44].

## 2.4 Main Result

Problem (2.2) has a nonlinear objective function and integrality requirements on its decision variables. Our main result shows that problem (2.2) is equivalent to the problem

$$\begin{aligned}
 \max \quad & \sum_{j \in N} r_j w_j & (2.3) \\
 \text{st} \quad & \sum_{j \in N} w_j + w_0 = 1 \\
 & \sum_{j \in N} a_{ij} \frac{w_j}{v_j} \leq b_i w_0 & \forall i \in M \\
 & 0 \leq \frac{w_j}{v_j} \leq w_0 & \forall j \in N,
 \end{aligned}$$

where the decision variables are  $\{w_j : j \in N \cup \{0\}\}$ . The problem above is a linear program. In this problem, we interpret the decision variable  $w_j$  as the probability that a customer purchases product  $j$  and  $w_0$  as the probability that a customer leaves without making a purchase. The first constraint ensures that a customer either purchases a product or leaves without purchasing. Interestingly, the second set of constraints are enough to ensure that the product offer decisions are chosen within the feasible set  $\mathcal{F}$ . The third set of constraints ensure the connection between the probability that a customer purchases a product and leaves without purchasing anything. In the next theorem, we show that problems (2.2) and (2.3) are equivalent to each other.

**Theorem 2.4.1.** *Problems (2.2) and (2.3) have the same optimal objective value and we can construct an optimal solution to one of these problems by using an optimal solution to the other.*

*Proof.* Using the decision variables  $y = \{y_j : j \in N\} \in \{0, 1\}^{|N|}$ , we claim that problem (2.2) is equivalent to the problem

$$\begin{aligned} \max \quad & \sum_{j \in N} (r_j - z^*) v_j y_j & (2.4) \\ \text{st} \quad & \sum_{j \in N} a_{ij} y_j \leq b_i & \forall i \in M \\ & 0 \leq y_j \leq 1 & \forall j \in N, \end{aligned}$$

which is a linear program. To see this equivalence, let  $x^*$  and  $y^*$  respectively be optimal solutions to problems (2.2) and (2.4). Since  $z^* = R(x^*) = \sum_{j \in N} r_j v_j x_j^* / (1 + \sum_{j \in N} v_j x_j^*)$ , we get  $\sum_{j \in N} r_j v_j x_j^* = z^* (1 + \sum_{j \in N} v_j x_j^*)$ . In this case, evaluating the objective value of problem (2.4) at the feasible solution  $x^*$ , we obtain  $\sum_{j \in N} (r_j - z^*) v_j x_j^* = z^* (1 + \sum_{j \in N} v_j x_j^*) - z^* \sum_{j \in N} v_j x_j^* = z^*$ , which implies that the optimal objective value of problem (2.4) is at least as large as the optimal objective value of problem (2.2). On the other hand, since  $A$  is totally unimodular and the objective function of problem (2.4) is linear, we can assume that  $y^* \in \{0, 1\}^{|N|}$  without loss of generality. Thus,  $y^* \in \mathcal{F}$ . In this case, evaluating the objective value of problem (2.2) at the feasible solution  $y^*$ , we have  $z^* \geq R(y^*) = \sum_{j \in N} r_j v_j y_j^* / (1 + \sum_{j \in N} v_j y_j^*)$ . Focusing on the first and last expressions in the last chain of inequalities and arranging the terms, we get  $z^* \geq \sum_{j \in N} (r_j - z^*) v_j y_j^*$ , which implies that the optimal objective value of problem (2.4) is at most as large as the optimal objective value of problem (2.2). Thus, problems (2.2) and (2.4) are equivalent to each other, sharing the same optimal objective value, establishing the claim. So, it is enough to show that problems

(2.3) and (2.4) are equivalent to each other, which is what we do in the rest of the proof.

We let  $w^* = \{w_j^* : j \in N \cup \{0\}\}$  be an optimal solution to problem (2.3) with objective value  $\zeta^*$  and  $y^*$  be an optimal solution to problem (2.4). The discussion above shows that  $y^*$  provides the objective value  $z^*$  for problem (2.4). We construct the solution  $\hat{w} = \{\hat{w}_j : j \in N \cup \{0\}\}$  to problem (2.3) as  $\hat{w}_j = P_j(y^*) = v_j y_j^* / (1 + \sum_{i \in N} v_i y_i^*)$  for all  $j \in N$  and  $\hat{w}_0 = 1 - \sum_{j \in N} P_j(y^*) = 1 / (1 + \sum_{i \in N} v_i y_i^*)$ . Since  $y^*$  is feasible to problem (2.4), it is simple to check that  $\hat{w}$  is feasible to problem (2.3). In particular,  $\hat{w}$  clearly satisfies the first constraint in problem (2.3). Furthermore, noting that  $\hat{w}_j / (v_j \hat{w}_0) = y_j^*$ , we have  $b_i \geq \sum_{j \in N} a_{ij} y_j^* = \sum_{j \in N} a_{ij} \hat{w}_j / (v_j \hat{w}_0)$  showing that  $\hat{w}$  satisfies the second set of constraints in problem (2.3). Finally, noting that  $1 \geq y_j^* = \hat{w}_j / (v_j \hat{w}_0)$ ,  $\hat{w}$  satisfies the third set of constraints in problem (2.3). In this case, the objective value provided by the feasible solution  $\hat{w}$  to problem (2.3) satisfies  $\zeta^* \geq \sum_{j \in N} r_j \hat{w}_j = \sum_{j \in N} r_j P_j(y_j^*) = R(y^*) = z^*$ . So, we have  $\zeta^* \geq z^*$ . On the other hand, we construct the solution  $\hat{y} = \{\hat{y}_j : j \in N\}$  to problem (2.3) as  $\hat{y}_j = w_j^* / (v_j w_0^*)$  for all  $j \in N$ . By using an argument similar to the one we just used, it is possible to show that  $\hat{y}$  is a feasible solution to problem (2.4). In this case, the objective value provided by the feasible solution  $\hat{y}$  to problem (2.4) satisfies  $z^* \geq \sum_{j \in N} r_j v_j \hat{y}_j - z^* \sum_{j \in N} v_j \hat{y}_j = \sum_{j \in N} r_j w_j^* / w_0^* - z^* \sum_{j \in N} w_j^* / w_0^* \geq \sum_{j \in N} r_j w_j^* / w_0^* - \zeta^* \sum_{j \in N} w_j^* / w_0^* = (\zeta^* - \zeta^*(1 - w_0^*)) / w_0^* = \zeta^*$ , where the second inequality uses the fact that  $\zeta^* \geq z^*$  shown above and the second equality uses the fact that  $\zeta^* = \sum_{j \in N} r_j w_j^*$  and  $\sum_{j \in N} w_j^* = 1 - w_0^*$  by the first constraint in problem (2.3). So, we have  $z^* \geq \zeta^*$ , establishing that the solutions  $w^*$ ,  $y^*$ ,  $\hat{w}$  and  $\hat{y}$  all provide the objective value  $z^*$  for their respective problems. Given  $w^*$ , the solution  $\hat{y}$  constructed through  $w^*$  is optimal to problem (2.4) and given  $y^*$ , the solution  $\hat{w}$  constructed through  $y^*$  is optimal to problem (2.3).  $\square$

Theorem 2.4.1 shows that problems (2.2) and (2.3) are equivalent and we can obtain an optimal solution to the former problem simply by solving the latter. This result is quite useful in practice since problem (2.3) is a linear program.

## 2.5 Applications

In this section, we give a number of specific cases where we can use Theorem 2.4.1 to obtain the optimal solutions to certain assortment optimization and pricing problems under MNL with a variety of constraints. For majority of these cases, efficient algorithms for obtaining the optimal solution do not exist in the earlier literature. Our results appear to provide the first tractable algorithms to obtain the optimal solution.

### 2.5.1 Cardinality Constraints

Consider the case where the total number of products that can be offered is limited to  $b$ . So, the feasible set of product offer decisions can be written as  $\mathcal{F} = \{x \in \{0, 1\}^{|N|} : \sum_{j \in N} x_j \leq b\}$ , in which case,  $A$  is given by  $(1, \dots, 1)$  with dimensions  $1 \times |N|$ . This matrix is clearly totally unimodular, which implies that we can find the optimal set of products to offer under a cardinality constraint directly by solving problem (2.3) with  $A = (1, \dots, 1)$ . [50] give an efficient algorithm for finding the optimal set of products to offer under MNL with a cardinality constraint, but our use of Theorem 2.4.1 allows us to solve this problem directly by using a linear program. Furthermore, by building on this theorem, we can find the optimal solution under more general cardinality constraints. We proceed to giving two examples of such more general cardinality constraints below.

Consider the case where there are  $K$  nested subsets of products such that

$S_1 \subset S_2 \subset \dots \subset S_K \subset N$  and there are integers  $b_1 \leq b_2 \leq \dots \leq b_K$  associated with each one of these subsets. The total number of products that we can offer in subset  $S_k$  is limited to  $b_k$ . These constraints may arise when, for example,  $S_1$  corresponds to the specialty products, whereas  $S_2$  corresponds to the set of all products and we do not want to offer more than a total of  $b_2$  products, while limiting the number of offered specialty products to  $b_1$ . Thus, the feasible set of product offer decisions can be written as  $\mathcal{F} = \{x \in \{0, 1\}^{|N|} : \sum_{j \in S_k} x_j \leq b_k \forall k = 1, \dots, K\}$ . Since  $S_1 \subset S_2 \subset \dots \subset S_K$ , this constraint matrix includes consecutive ones in each row. Such matrices are called interval matrices and they are known to be totally unimodular; see Corollary 2.10 in Chapter III.1 of [44]. Thus, we can solve a linear program to find the optimal assortment to offer under nested cardinality constraints.

As another example, consider the case where the products are partitioned into  $K$  disjoint subsets  $S_1, S_2, \dots, S_K$ . Without loss of generality, we assume that the products are indexed by the integers  $\{1, \dots, n\}$  and we have  $S_k = \{i_k, \dots, i_{k+1} - 1\}$  with  $i_1 = 1$  and  $i_{K+1} = n + 1$ . The total number of products we can offer in subset  $S_k$  is limited to  $b_k$ . In this case, the feasible set of product offer decisions is  $\mathcal{F} = \{x \in \{0, 1\}^{|N|} : \sum_{j=i_k}^{i_{k+1}-1} x_j \leq b_k \forall k = 1, \dots, K\}$  and this constraint matrix still corresponds to an interval matrix. Such constraints may arise when, for example,  $S_1$  corresponds to the specialty products, whereas  $S_2$  corresponds to the general interest products and we separately want to limit the numbers of offered specialty and general interest products. Furthermore, if  $S_k$  overlaps with only  $S_{k-1}$  and  $S_{k+1}$ , then the constraint matrix is still an interval matrix. So, even if some products may count both as specialty and general interest products, we can still limit the numbers of offered specialty and general interest products. To our knowledge, it is difficult to capture such overlapping cardinality constraints



with earlier frameworks.

## 2.5.2 Display Location Effects

Consider the case where the preference weight of a product depends on where it is displayed. For example, if the products are displayed in a store window or a shelf, then customers may tend to overlook a product when it is displayed at the back of the window or the shelf, which can be captured by a smaller preference weight when the product is displayed at the back. In online retail, customers may be more likely to choose products that are displayed at the top of search results, which can be captured by using preference weights that depend on the display order of the product. It turns out we can build on Theorem 2.4.1 to find the optimal set of products to offer when the preference weights depend on the display location.

We use  $v_{jl}$  to denote the preference weight of item  $j$  when this item is displayed at location  $l$ . Without loss of generality, we assume that there are as many possible locations as the number of items so that we can offer all items at once. In this case, we can index both the items and the locations by  $N$ . If the number of possible locations is smaller than the number of items, then we can define additional locations with  $v_{jl} = 0$  for all  $j \in N$ , for each additional location  $l$ , in which case, using one of these additional locations for an item is equivalent to not displaying the item at all. To capture the product offer decisions, we use  $x = \{x_{jl} : j, l \in N\} \in \{0, 1\}^{|N| \times |N|}$ , where  $x_{jl} = 1$  if we offer item  $j$  in location  $l$ , otherwise  $x_{jl} = 0$ . If the product offer decisions are given by  $x$ , then we obtain an expected revenue of  $\sum_{j,l \in N} r_j v_{jl} x_{jl} / (1 + \sum_{j,l \in N} v_{jl} x_{jl})$ . Therefore, we are interested in solving the

problem

$$\begin{aligned}
\max \quad & \frac{\sum_{j,l \in N} r_j v_{jl} x_{jl}}{1 + \sum_{j,l \in N} v_{jl} x_{jl}} \\
\text{st} \quad & \sum_{l \in N} x_{jl} \leq 1 \quad \forall j \in N \\
& \sum_{j \in N} x_{jl} \leq 1 \quad \forall l \in N \\
& x_{jl} \in \{0, 1\} \quad \forall j, l \in N,
\end{aligned}$$

where the first set of constraints ensure that each item is offered at most in one location and the second set of constraints ensure that each location is used by at most one item. The problem above is a special case of problem (2.2) where each product is indexed by  $(j, l) \in N \times N$ . Its constraint matrix is the constraint matrix of an assignment problem, which is totally unimodular; see Corollary 2.9 in Chapter III.1 of [44]. So, by Theorem 2.4.1, we can use a linear program to find the optimal set of items to offer under display location effects.

### 2.5.3 Pricing with a Finite Price Menu

Consider the case where the price of a product is a decision variable, rather than being fixed. The preference weight of a product depends on its price. Increasing the price of a product is expected to make it less desirable to customers, effectively decreasing its preference weight, but we are not strictly tied to the assumption that higher prices result in lower preference weights. The goal is to choose the prices of the products so as to maximize the expected revenue from each customer. We show how to solve this pricing problem as a linear program as long as the prices are chosen within a finite set of possible price levels.

We let  $K$  be the set of possible price levels for an item. The price corresponding to price level  $k$  for an item is given by  $r_k$ . Therefore,  $\{r_k : k \in K\}$  becomes the

possible prices for an item. If we use the price level  $k$  for item  $j$ , then its preference weight is  $v_{jk}$ . Our notation indicates that the set of possible prices for each item is the same, but it is straightforward to extend our formulation to incorporate different sets of possible prices for different items. To capture our pricing decisions, we use  $x = \{x_{jk} : j \in N, k \in K\} \in \{0, 1\}^{|N| \times |K|}$ , where  $x_{jk} = 1$  if we set the price of item  $j$  at price level  $k$ , otherwise  $x_{jk} = 0$ . Thus, we want to solve the problem

$$\begin{aligned} \max \quad & \frac{\sum_{j \in N} \sum_{k \in K} r_k v_{jk} x_{jk}}{1 + \sum_{j \in N} \sum_{k \in K} v_{jk} x_{jk}} \\ \text{st} \quad & \sum_{k \in K} x_{jk} = 1 \quad \forall j \in N \\ & x_{jk} \in \{0, 1\} \quad \forall j \in N, k \in K, \end{aligned}$$

where the constraints ensure that each item is offered at one price level. Similar to our observations for display location effects, the problem above is a special case of problem (2.2) where each product is indexed by  $(j, k) \in N \times K$ . Each row of the constraint matrix corresponds to an item  $j$  and it includes consecutive ones, corresponding to the different price levels for item  $j$ . Thus, the constraint matrix is an interval matrix, which is totally unimodular. In this case, using Theorem 2.4.1, we can find the optimal prices by solving a linear program. Our formulation above assumes that each product has to be offered. If we need to jointly decide which products to offer and the prices of the offered products, then we can simply replace the equality constraint in the problem above with a “less than” constraint. Furthermore, if we want to impose a limit of  $b$  on the number of products we offer, then we can add the constraint  $\sum_{j \in N} \sum_{k \in K} x_{jk} \leq b$ . The additional constraint amounts to adding a row of ones to the constraint matrix, which does not change the fact that the constraint matrix is an interval matrix.

Pricing models traditionally assume a parametric relationship between the price of an item and its preference weight. In particular, it is usually assumed that if

the price of item  $j$  is  $p$ , then its preference weight is  $e^{\alpha_j - \beta_j p}$ , for some constants  $\alpha_j$  and  $\beta_j$ . Using this parametric form, the pricing problem can be formulated as a smooth optimization problem, involving prices as the decision variables, but the objective function of this problem is not concave. [55] and [12] formulate the problem in terms of the market share of an item to get a concave objective function, but their work is based on the specific parametric relationship between price and preference weight. In our formulation, the preference weight of an item can depend on its price in an arbitrary fashion. Furthermore, since we work with discrete price levels, we can limit attention to operationally appealing prices, such as those in increments of a dollar.

#### 2.5.4 Quality Consistent Pricing

Similar to our pricing model above, consider the case where the prices of the products are decision variables, but there is an inherent ordering between the products in terms of their quality. In particular, the products are indexed such that the first product is lower quality than the second one, the second product is lower quality than the third one and so on. When setting the prices of the products, we need to ensure that the lower quality products have lower prices. Such a pricing scheme is called quality consistent pricing or price laddering. It occurs when products have a clear ordering in terms of quality, richness of features or durability. For example, [48] describe an application where option rich automobiles of the same model have to be priced higher than the option poor ones. The objective is to choose the prices of the products to maximize the expected revenue from each customer, while adhering to the quality consistency constraint.

We index the items by  $N = \{1, 2, \dots, n\}$  and the possible price levels for an item by  $K = \{1, \dots, m\}$ . The price corresponding to price level  $k$  of an item is  $r_k$ .

Therefore, the possible price levels for an item are given by  $\{r_k : k \in K\}$ . Without loss of generality, we assume that the possible price levels are indexed such that  $r_1 \leq r_2 \leq \dots \leq r_m$  so that lower indices correspond to lower prices. Also, we assume that the items are indexed such that the first item is the lowest quality one and the last item is the highest quality one. So, the price of the first item must be smaller than the price of the second item, which, in turn, must be smaller than the price of the third item and so on. We let  $v_{jk}$  be the preference weight of item  $j$  when we price this item at price level  $k$ . To capture our pricing decisions, we use  $x = \{x_{jk} : j \in N, k \in K\} \in \{0, 1\}^{|N| \times |K|}$ , where  $x_{jk} = 1$  if we price item  $j$  at price level  $k$ , otherwise  $x_{jk} = 0$ . To impose the quality consistency constraint, we use the additional decision variables  $z = \{z_{jk} : j \in N, k \in K \setminus \{m\}\} \in \{0, 1\}^{|N| \times (|K|-1)}$ , where  $z_{jk} = 1$  if we price item  $j$  at price level  $k + 1$  or higher, otherwise  $z_{jk} = 0$ . Note that we do not need the decision variables  $\{z_{jm} : j \in N\}$ . We want to solve the problem

$$\begin{aligned}
\max \quad & \frac{\sum_{j \in N} \sum_{k \in K} r_k v_{jk} x_{jk}}{1 + \sum_{j \in N} \sum_{k \in K} v_{jk} x_{jk}} \\
\text{st} \quad & x_{11} + z_{11} = 1 \\
& x_{1k} + z_{1k} = z_{1,k-1} \quad \forall k = 2, \dots, m-1 \\
& x_{1m} = z_{1,m-1} \\
& x_{j1} + z_{j1} = x_{j-1,1} \quad \forall j = 2, \dots, n \\
& x_{jk} + z_{jk} = x_{j-1,k} + z_{j,k-1} \quad \forall j = 2, \dots, n, k = 2, \dots, m-1 \\
& x_{jm} = x_{j-1,m} + z_{j,m-1} \quad \forall j = 2, \dots, n \\
& x_{jk}, z_{jl} \in \{0, 1\} \quad \forall j \in N, k \in K, l \in K \setminus \{m\}.
\end{aligned}$$

We give interpretations for the last three sets of constraints above and the first three sets can be interpreted in a similar fashion. The fourth set of constraints ensure that if item  $j - 1$  is priced at level 1, then item  $j$  is either priced at level 1

or it is priced at level 2 or higher. The fifth set of constraints ensure that if item  $j - 1$  is priced at level  $k$  or if we decide to price item  $j$  at level  $k$  or higher, then item  $j$  is either priced at level  $k$  or it is priced at level  $k + 1$  or higher. The sixth set of constraints ensure that if item  $j - 1$  is priced at level  $m$  or if we decide to price item  $j$  at level  $m$  or higher, then item  $j$  is priced at level  $m$ .

The constraint matrix above corresponds to the constraints of a shortest path problem. To see this result, consider a network composed of the nodes  $\{(j, k) : j \in N, k \in K\}$  and a sink node. Figure 2.5.4 shows a sample network with  $n = 4$  and  $m = 3$ . In the problem above, the decision variable  $x_{jk}$  for  $j \in N \setminus \{n\}$ ,  $k \in K$  corresponds to an arc from node  $(j, k)$  to node  $(j + 1, k)$ . The decision variable  $x_{nk}$  for  $k \in K$  corresponds to an arc from node  $(n, k)$  to the sink node. Finally, the decision variable  $z_{jk}$  for  $j \in N$ ,  $k \in K \setminus \{m\}$  corresponds to an arc from node  $(j, k)$  to node  $(j, k + 1)$ . The first three sets of constraints are the flow balance constraints for the nodes in  $\{(1, k) : k \in K\}$ , whereas the last three sets of constraints are the flow balance constraints for the nodes in  $\{(j, k) : j \in N \setminus \{1\}, k \in K\}$ . The flow balance constraint of the sink node is redundant and it is omitted. The supply of node  $(1, 1)$  is one. So, the horizontal arcs in Figure 2.5.4 correspond to the decision variables  $\{x_{jk} : j \in N, k \in K\}$ , whereas the vertical arcs correspond to the decision variables  $\{z_{jk} : k \in N, k \in K \setminus \{m\}\}$ . The important observation is that if a unit of flow from node  $(1, 1)$  to the sink node follows the arc corresponding to the decision variable  $x_{jk}$ , then it can never follow the arc corresponding to the decision variable  $x_{j+1, k-1}$ , which means that if we choose price level  $k$  for item  $j$ , then we cannot choose price level  $k - 1$  for item  $j + 1$ , which is enough to impose the price ladder. Lastly, we observe that if we add the first three sets of constraints above, then we obtain  $\sum_{k \in K} x_{1k} = 1$ , whereas if we add the last three sets of constraints over  $k \in K$  for a particular item  $j$ , then we obtain  $\sum_{k \in K} x_{jk} = \sum_{k \in K} x_{j-1, k}$ . Thus, the

constraints above ensure that  $\sum_{k \in K} x_{jk} = 1$  for all  $j \in N$ , indicating that we offer each item at one price level.

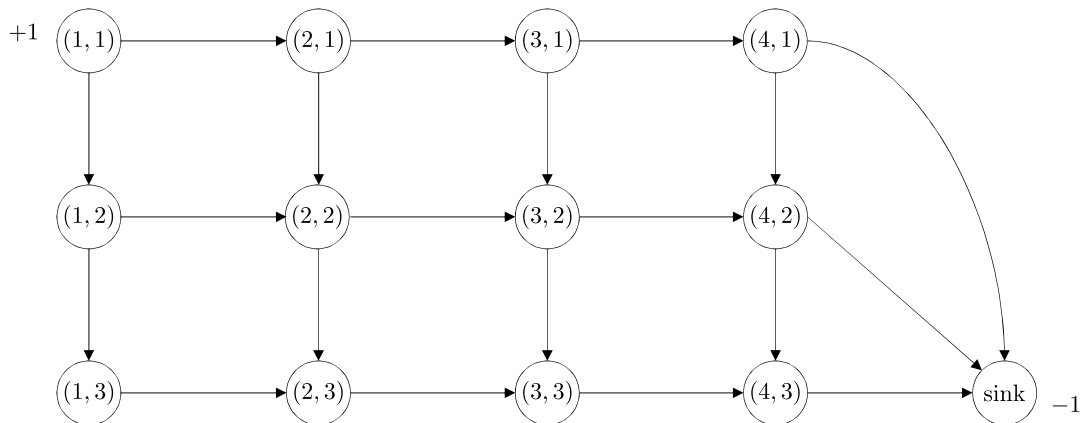


Figure 2.1: Shortest path problem in the quality consistent pricing setting.

Network matrices are totally unimodular by Proposition 3.1 in Chapter III.1 of [44]. So, by Theorem 2.4.1, we can solve quality consistent pricing problems by using a linear program. To our knowledge, tractability of incorporating quality consistency or laddering constraints into pricing problems under MNL was not known previously.

The development in this section assumes that the qualities of the items satisfy the full ordering  $1 \preceq 2 \preceq \dots \preceq n$ , in which case, the prices of the items have to satisfy this ordering as well. By modifying the network in Figure 2.5.4 slightly, we can handle the case where we have a partial ordering between the qualities of the items. For example, assuming that there are five items, the qualities of the items may satisfy the partial ordering  $1 \preceq \{2, 3, 4\} \preceq 5$ , which is to say that the first item is lower quality than the second, third and fourth items, which are, in turn, lower quality than the fifth item, but there is no clear ordering between the qualities of the second, third and fourth items. In this case, the price of the first item should be lower than the prices of the second, third and fourth items. Similarly,

the prices of the second, third and fourth items should be lower than the price of the fifth item, but there is no constraint on how the prices of the second, third and fourth items are ordered. Building on the approach described in this section, it is possible to show that we can still solve a linear program to enforce such a partial quality consistency constraint. Finally, our approach continues to apply when there are disjoint quality consistency constraints in the sense that the items are partitioned into disjoint subsets  $S_1, \dots, S_L$  and we impose a separate quality consistency constraint for the items in each one of the subsets  $S_l$  for  $l = 1, \dots, L$ .

### 2.5.5 Product Precedence Constraints

We consider assortment optimization problems where a particular product cannot be offered to customers unless a certain set of related products are also offered. This kind of a constraint can arise when a company offers multiple versions of a product and company policy or law prohibits offering a more expensive or sophisticated version of the product unless a more inexpensive or basic version is offered. For example, it may not be possible to offer the brand name version of a drug unless the generic version is offered. To model such product precedence constraints, we use  $S_j \subset N$  to denote the set of products that we need to offer to be able to offer product  $j$ . So, the feasible set of product offer decisions is given by  $\mathcal{F} = \{x \in \{0, 1\}^{|N|} : x_j - x_i \leq 0 \ \forall j \in N, i \in S_j\}$ , indicating that we can have  $x_j = 1$  only when  $x_i = 1$  for all  $i \in S_j$ . In this constraint matrix, each row includes only a +1 and a -1. Such matrices are known to be totally unimodular; see Proposition 2.6 in Chapter III.1 of [44], along with [26] and [27]. Thus, by Theorem 2.4.1, it follows that we can find the optimal assortment under product precedence constraints by solving a linear program. We observe that the subsets  $\{S_j : j \in N\}$  in product precedence constraints can be completely arbitrary. In particular, they can be



overlapping and products can have circular dependencies on each other.

Closing this section, we note that if we want to enforce offering a certain product in any of the applications considered in this section, then we can impose a lower bound of one on the sum of the decision variables corresponding to the offer decisions for this product. We can check that the constraint matrix remains totally unimodular under this additional constraint. Furthermore, joining two totally unimodular matrices does not yield a totally unimodular matrix in general, but it may be possible to combine some of the constraints considered in this section without destroying the total unimodularity of the constraint matrix. For example, joining two interval matrices yields an interval matrix. Since a cardinality constraint and pricing with a finite price menu both yield interval constraint matrices, we can impose a cardinality constraint on the offered assortment when solving a pricing problem with a finite price menu.

## CHAPTER 3

# ASSORTMENT OPTIMIZATION UNDER THE NESTED LOGIT MODEL

### 3.1 Introduction

In this chapter, we study a class of assortment optimization problems where the choices of the customers are governed by the nested logit model. Under this model, customers first select a nest, and then, a product within the nest. We assume that there is a revenue associated with each product and the objective is to find a set of products, or an assortment, to offer that maximizes the expected revenue per customer. This assortment optimization problem is combinatorial in nature and the number of possible assortments can be very large, particularly when there are many potential products to offer. In airline and hotel revenue management settings, for example, the number of products can easily exceed 30 or 40, yielding  $2^{30}$  or  $2^{40}$  possible assortments. Therefore, it is important to classify when the problem is polynomially solvable. When not, it is important to find solution methods with worst-case performance guarantees.

To our knowledge, there is no work on assortment optimization under the nested logit model that can deal with a large number of nests. One of our main contributions is to classify the complexity of the assortment problem for nested attraction models. We do this along two dimensions. The first dimension is the magnitude of the nest dissimilarity parameters, which characterize the degree of dissimilarity of the products within a nest. The second dimension is the presence or absence of the no purchase alternative within a nest. This divides the problem into four cases.

We show that the only polynomially solvable case is when the nest dissimilarity

parameters are less than one and the no purchase alternative is only available at the time of selecting a nest. This situation conforms to the standard form of the nested logit model; see [4]. For this case, we show that it is optimal to offer a nested-by-revenue assortment within each nest, but this result does not immediately imply that the problem is polynomially solvable since there are exponentially many combinations of nested-by-revenue assortments we can choose for the different nests. We deal with this difficulty by giving a linear program that finds the best combination of nested-by-revenue assortments for each nest. Thus, the problem is tractable under the standard form of the nested logit model.

If the nest dissimilarity parameters exceed one or the customers can choose a no purchase option after selecting a nest, then we show that the problem is NP-hard. These cases correspond to nonstandard versions of the nested logit model, but we justify the practical importance of these cases. For all of these cases we give parsimonious collections of assortments such that if we focus only on these assortments then we obtain a solution with a certain worst-case performance guarantee.

In particular, the second case we consider focuses on the situation where the dissimilarity parameters can take on any value, but the customers always purchase a product within the selected nest. For this case, we show that if we focus only on nested-by-revenue assortments, then assuming that the revenues of the products in the same nest differ by at most a factor of  $\rho$  and the attractiveness parameters of the products in the same nest differ by at most a factor of  $\kappa$ , the expected revenue from the best nested-by-revenue assortment cannot deviate from the optimal expected revenue by more than a factor of  $\min\{\rho, 2\kappa\}$ . Therefore, we can expect the nested-by-revenue assortments to perform well when the revenues or the attractiveness of the products within a particular nest are not too different from each other. It is

important to emphasize that this result allows the revenues or the attractiveness of the products to differ arbitrarily when the products are in different nests.

In the third case, we consider the situation where the dissimilarity parameters of the nests are less than one, but customers may leave a chosen nest without purchasing. For this case, we construct a small collection of assortments such that the best assortment within this collection provides an expected revenue that deviates from the optimal expected revenue by no more than a factor of two. Finally, the fourth case considers the most general problem instances with no restrictions on the dissimilarity parameters of the nests and the no purchase behavior. For this case, we give a collection of assortments such that the best assortment within this collection has a worst-case performance guarantee of  $2\kappa$ , where  $\kappa$  is as defined in the paragraph above. Furthermore, we exploit the connections of our assortment problem to the partition problem to give a pseudo-polynomial-time algorithm for the most general instances of our problem. Finally, using  $\bar{\gamma}$  to denote the largest dissimilarity parameter for the nests, for any given  $\delta > 1$ , we give a collection of assortments that provides a worst-case performance guarantee of  $\delta^{2\max\{\bar{\gamma}, 1\}+1}$ , but the work required to obtain this collection increases as  $\delta$  gets close to one. Thus, this result is akin to a polynomial-time approximation scheme when  $\bar{\gamma}$  is fixed. Table 3.1 summarizes the four cases and indicates which sections in the paper include each one of these cases.

In addition to the worst-case performance guarantees, we formulate a convex program that yields an upper bound on the optimal expected revenue. By comparing the upper bound on the optimal expected revenue with the expected revenue provided by an assortment, we bound the optimality gap of the assortment we obtain for a particular problem instance. We use this approach in our computational experiments to test the performance of the solutions we obtain by focusing

		NO PURCHASE BEHAVIOR	
		Always Purchase in Nest	May Leave Nest
Dissimilarity	$0 \leq \gamma \leq 1$	Case 1: Section 3.5 Polynomially-solvable	Case 3: Section 3.7 Worst-case guarantee of 2
	Free	Case 2: Section 3.6 Worst-case guarantee of $\min\{\rho, 2\kappa\}$	Case 4: Section 3.8 Worst-case guarantee of $2\kappa$ , pseudo-poly-time algorithm, approximation scheme for fixed $\bar{\gamma}$

Table 3.1: Four cases considered in the chapter. The ratio between the largest and the smallest product revenues in a nest is bounded by  $\rho$ , the ratio between the largest and the smallest attractiveness parameters in a nest is bounded by  $\kappa$  and the dissimilarity parameters are bounded by  $\bar{\gamma}$ .

on the collections of assortments mentioned above. In this way, we characterize the problem parameters that affect the solution quality and empirically demonstrate that the performance of the assortments we propose follows the trends predicted by their worst-case performance guarantees.

## 3.2 Literature Review

Research on pricing in the context of the multinomial logit and nested logit models has been fairly active. In that setting, the problem is to choose a set of prices for the products, where the prices of all products jointly determine the probability that a customer purchases a particular product. The objective is to maximize the expected revenue per customer. For the pricing problem, [25] notice that the expected revenue function fails to be concave in prices for the multinomial logit model, but significant progress was made by formulating the pricing problem in terms of market shares, as this results in a concave expected revenue function; see [55] and [12]. [34] extend the concavity result to the nested logit model by assuming that the price sensitivities of the products are constant within each nest

and the nest dissimilarity parameters are all between zero and one. They show that the expected revenue maximization problem can be reduced to optimizing over a single variable. [21] relax both of the assumptions in [34] and extend the analysis to more general nested attraction models. The key result is that the optimal prices add two terms to the unit costs, where the first term is the inverse of the price sensitivities of the products and the second term is a nest-dependent constant. This result implies that products with the same price sensitivity in a nest have the same markup, irrespective of their quality.

When the price of each product is fixed then we are faced with a pure assortment problem. We address the pure assortment problem when customers choose according to the multinomial logit model (MNL) in Chapter 1. Additionally, if the customers choose according to the MNL then [57] show that the optimal assortment includes a certain number of products with the largest revenues. We refer to assortments that include a certain number of products with the largest revenues as nested-by-revenue assortments. The problem becomes more complicated when more general choice models are considered. [52] study the assortment problem under the mixed multinomial logit model, where there are multiple customer types and customers of different types choose according to different multinomial logit models. They show that the assortment optimization problem is NP-hard in the weak sense even with two customer types and provide a performance guarantee for nested-by-revenue assortments. [5] show that the same problem is NP-hard in the strong sense and [43] give a branch-and-cut algorithm to find the optimal assortment. [53] study the robust assortment problem under the multinomial logit model when some of the parameters of the choice model are not known. [50] consider constraints on the size of the offered assortment when customers choose according to the multinomial logit model. [30] consider joint assortment optimiza-

tion and pricing problems under the nested logit model, where both the set of products offered and their corresponding prices are decision variables. They work with two nest structures. In the first nest structure, customers first select a brand, and then, a product type within the selected brand, whereas in the second nest structure, customers first select a product type, and then, a brand for the selected product type. The authors characterize the structure of the optimal solution, but the problem becomes difficult when the number of brands is large or the product prices are fixed.

In this chapter, we use the nested logit model to capture customer choices. There are several desirable aspects of this choice model. To begin with, the nested logit model alleviates the independence of irrelevant alternatives property suffered by the multinomial logit model; see [2]. In particular, if a product is added to the offered assortment, then the multinomial logit model predicts that the market share of each product in the offered assortment decreases by the same relative amount, which clearly should not occur when different products cannibalize on each other to different extents. Furthermore, it is possible to show that the nested logit model is compatible with a random utility-based choice model, where customers associate random utilities with the products and with the option of not making a purchase and they follow the option providing the largest utility. This feature gives some behavioral justification to the nested logit model; see [4]. Finally, the nested logit model allows correlations between the utilities of the products, capturing the fact that the way a customer evaluates a certain product may help us predict how this customer would evaluate other similar products.

Beside its desirable aspects, the standard form of the nested logit model has some limitations. This choice model works with a fixed nest structure, where customers first select a nest, and then, a product within the selected nest. For

example, nests may correspond to different airlines and products within a nest may correspond to different cabin classes offered by an airline. However, not all customers follow the same nest structure. Some customers may select an airline first, and then, a cabin class offered by this airline, whereas some customers may select a cabin class first, and then, an airline that offers this cabin class. Several extensions to the nested logit model are designed to alleviate this concern; see [59]. In the mixed nested logit model, there are customers of multiple types and customers of different types choose according to different nested logit models, possibly with different nest structures. In paired combinatorial and cross nested logit models, a product may appear in multiple nests. In this paper, we use the standard form of the nested logit model with a fixed nest structure. [5] show that the assortment problem is NP-hard under the mixed multinomial logit model and this result carries over to the mixed nested logit model. [30] characterize the optimal assortment under a mixed nested logit model with identical product prices.

### 3.3 Problem Formulation

In this section, we describe the nested logit model that we use to model the customer choice, and then, formulate our assortment optimization problem. There are  $m$  nests indexed by  $M = \{1, \dots, m\}$ . Depending on the application setting, each nest may represent a different category of products, a different sales channel or a different retail store. There are  $n$  products that we can offer in each nest. We index the products in each nest by  $N = \{1, \dots, n\}$ . We use  $r_{ij}$  to denote the revenue associated with product  $j$  in nest  $i$ . Without loss of generality, we assume that the products in each nest are ordered such that  $r_{i1} \geq r_{i2} \geq \dots \geq r_{in}$  for all  $i \in M$ . We let  $v_{ij}$  be the preference weight of product  $j$  in nest  $i$  and  $v_{i0}$  be the preference weight of the no purchase option in nest  $i$ . We use  $V_i(S_i)$  to denote



the total preference weight of all available options when we offer the assortment  $S_i \subset N$  in nest  $i$ . In other words, we have  $V_i(S_i) = v_{i0} + \sum_{j \in S_i} v_{ij}$ . Under the nested logit model, given that a customer decides to purchase a product in nest  $i$ , if we offer the assortment  $S_i$  in this nest, then the probability that the customer purchases product  $j \in S_i$  is given by

$$P_{ij}(S_i) = \frac{v_{ij}}{v_{i0} + \sum_{k \in S_i} v_{ik}} = \frac{v_{ij}}{V_i(S_i)}.$$

We observe that the assumption that each nest includes the same number of products is without loss of generality because if some nest  $i$  includes fewer than  $n$  products, then we can include additional products  $j$  in this nest with preference weight  $v_{ij} = 0$  and these products would never be purchased. Also, it is possible to have  $v_{i0} = 0$  for some nest  $i$  or even all nests, in which case, given that a customer selects nest  $i$ , he never leaves without purchasing anything. We note that if  $v_{i0} = 0$ , then the expression for  $P_{ij}(\emptyset)$  can evaluate to  $0/0$ , but this does not pose any difficulty since if we offer the empty assortment in nest  $i$  and  $v_{i0} = 0$ , then a customer would never decide to make a purchase in nest  $i$  and the value of  $P_{ij}(\emptyset)$  becomes irrelevant.

Each nest  $i$  has a parameter  $\gamma_i \geq 0$  associated with it that characterizes the degree of the dissimilarity of the products in the nest. Furthermore, we use  $v_0$  to denote the preference weight for the option of not choosing any of the nests. If we offer the assortment  $(S_1, \dots, S_m)$  over all nests, then a customer chooses nest  $i$  with probability

$$Q_i(S_1, \dots, S_m) = \frac{V_i(S_i)^{\gamma_i}}{v_0 + \sum_{l \in M} V_l(S_l)^{\gamma_l}}. \quad (3.1)$$

We note that  $\gamma_i$  serves the purpose of dampening or magnifying the total preference weight of the available options within nest  $i$ . We allow having  $v_0 = 0$ . We note that if we have  $v_0 = 0$ , we offer the empty assortment in all nests and  $v_{i0} = 0$

as well for all  $i \in M$ , then the expression for  $Q_i(\emptyset, \dots, \emptyset)$  evaluates to  $0/0$ , but this does not create any complication since if we offer the empty assortment in all nests, then we trivially make an expected revenue of zero. Therefore, all of our development in the paper applies to the case where some or all of the preference weights  $v_0$  and  $(v_{10}, \dots, v_{m0})$  associated with the no purchase options are equal to zero.

If we offer the assortment  $S_i$  in nest  $i$ , then we can write the expected revenue we obtain from this nest as

$$R_i(S_i) = \sum_{j \in S_i} r_{ij} P_{ij}(S_i) = \frac{\sum_{j \in S_i} r_{ij} v_{ij}}{V_i(S_i)},$$

with the interpretation that  $R_i(\emptyset) = 0$ . Therefore, if we offer the assortment  $(S_1, \dots, S_m)$  over all nests with  $S_i \subset N$  for all  $i \in M$ , then we obtain an expected revenue of

$$\Pi(S_1, \dots, S_m) = \sum_{i \in M} Q_i(S_1, \dots, S_m) R_i(S_i).$$

Our goal is to choose an assortment  $(S_1, \dots, S_m)$  that maximizes the expected revenue over all nests, yielding the assortment optimization problem

$$Z^* = \max_{(S_1, \dots, S_m): S_i \subset N, i \in M} \Pi(S_1, \dots, S_m). \quad (3.2)$$

Throughout this paper, we classify the instances of the assortment optimization problem in (3.2) along two dimensions. The first dimension is based on the values of the dissimilarity parameters  $(\gamma_1, \dots, \gamma_m)$  of the nests. Along this dimension, we separately consider the two cases where (i) we have  $\gamma_i \leq 1$  for all  $i \in M$ , and (ii) there are no restrictions on the dissimilarity parameters. The second dimension of classification is based on the values of the preference weights  $(v_{10}, \dots, v_{m0})$  of the no purchase options within the nests. Along this dimension, we separately consider the two cases where (i) we have  $v_{i0} = 0$  for all  $i \in M$ , and (ii) there

are no restrictions on the preference weights of the no purchase options. Since there are two cases to consider along each one of the two dimensions, we study the assortment optimization problem in (3.2) under four cases. It turns out that while the assortment optimization problem in (3.2) is polynomially solvable when  $\gamma_i \leq 1$  and  $v_{i0} = 0$  for all  $i \in M$ , lifting any one of these restrictions renders the problem NP-hard and we resort to approximation methods.

Along the first dimension, the case with  $\gamma_i \leq 1$  for all  $i \in M$  corresponds to the standard form of the nested logit model studied by [38] and [40]. In these papers, the author notes that having  $\gamma_i \leq 1$  for all  $i \in M$  implies that the nested logit model is always compatible with a random utility-based choice model, irrespective of the values of the preference weights of the products and the preference weights of the no purchase options. In particular, [40] postulates three assumptions that need to be satisfied for the nested logit model to be compatible with a random utility-based choice model and shows that these assumptions are always satisfied when the nest dissimilarity parameters do not exceed one. [4] revisits the compatibility of the nested logit model with utility maximization principle and shows that even if we do not have  $\gamma_i \leq 1$  for all  $i \in M$ , the three assumptions postulated by [40] can be satisfied for certain values of the preference weights  $\{v_{ij} : i \in M, j \in N\}$ ,  $\{v_{i0} : i \in M\}$  and  $v_0$ . Therefore, the nested logit model can still be compatible with a random utility-based choice model when we have  $\gamma_i \geq 1$  for some  $i \in M$ . Similarly, [59] notes that the preference weight  $v_{ij}$  of product  $j$  in nest  $i$  has the form  $e^{\bar{u}_{ij}/\gamma_i}$ , where  $\bar{u}_{ij}$  is the mean utility of product  $j$  in nest  $i$  captured through a linear combination of its attributes, such as price, quality and ease of use. He argues that even if the dissimilarity parameters exceed one, the nested logit model can be compatible with utility maximization principle for certain values of the mean utilities  $\{\bar{u}_{ij} : i \in M, j \in N\}$ . There are a number of empirical

studies that fit the nested logit model to customer choice data and end up with estimates for the nest dissimilarity parameters exceeding one; see [61], [60], [32], [58] and [66] applications in telephone service and housing choice. [61] interpret a nest dissimilarity parameter exceeding one as a signal that substitution across nests happens more readily than substitution within a nest.

Interestingly, using dissimilarity parameters that take on values larger than one allows us to model synergistic, or halo, effects among the products within the same nest. In particular, under the nested logit model, if we offer the assortment  $(S_1, \dots, S_m)$ , then the probability that a customer purchases product  $j \in S_i$  in nest  $i$  is given by

$$Q_i(S_1, \dots, S_m) P_{ij}(S_i) = \frac{V_i(S_i)^{\gamma_i-1}}{v_0 + \sum_{l \in M} V_l(S_l)^\gamma} v_{ij}.$$

From the expression above, we observe that when  $\gamma_i \leq 1$ , adding a product  $k \notin S_i$  to nest  $i$  always decreases the purchase probability of product  $j \in S_i$ . Therefore, when the dissimilarity parameters of the nests do not exceed one, the products in a nest always act as competitors to each other and adding a new product to a nest decreases the probability of purchase for the other products in the nest. In practice, this is not always the case. For example, if the nests correspond to different car dealers, then offering a new luxury car may increase the probability of purchase for other cars in the same dealer because the newly offered luxury car may help attract a larger fraction of customers. We observe that if  $\gamma_i > 1$  and we add a new product to nest  $i$ , then both  $V_i(S_i)^{\gamma_i-1}$  and  $\sum_{l \in M} V_l(S_l)^\gamma$  increase in the expression above. As a result, the probability that a customer purchases product  $j$  may increase or decrease. This feature may allow us to model synergies between different products in a nest. When such synergies exist, it may even be beneficial to include loss leaders in a nest to attract traffic to this nest. Motivated by this observation, we refer to the case with  $\gamma_i \leq 1$  for all  $i \in M$  as the case

with purely *competitive products*, whereas we refer to the case with no restrictions on the dissimilarity parameters of the nests as the case with *possibly synergistic products*.

Along the second dimension, the case with  $v_{i0} = 0$  for all  $i \in M$  corresponds to a situation where if a customer decides to make a purchase within a particular nest, then the customer always makes a purchase within the selected nest. In other words, the demand within a nest is fully captured without loss to the no purchase option. We refer to this situation as the case with *fully-captured nests*. On the other hand, if the preference weight of the no purchase option within a nest is strictly positive, then a customer may leave without purchasing anything in the selected nest. We refer to this situation as the case with *partially-captured nests*. There are a number of reasons to consider the case with partially-captured nests. To begin with, firms do not make their assortment offer decisions in isolation and the preference weight  $v_{i0}$  of the no purchase option in nest  $i$  can be used to capture the attractiveness of the products offered by other firms in nest  $i$ . For example, if each nest corresponds to a particular store, then the total preference weight of the products offered by all firms may serve to attract the customers to a store, but once a customer decides to make a purchase in a particular store, he may choose a product offered by another firm, which is equivalent to not making a purchase within the offered assortment. As mentioned above, the assortment problem becomes NP-hard when we allow partially-captured nests. Game theoretic assortment optimization models is beyond the scope of our paper, but this computational complexity result also indicates that finding the best response of a firm to the assortments already offered by other firms is a computationally difficult problem.

Another reason to consider the case with partially-captured nests is that there

are extensions of the nested logit model that allow offering a particular alternative within multiple nests. These extensions are referred to as generalized nested logit models and [59] shows that generalized nested logit models are consistent with a random utility-based choice model. Our use of the partially-captured nests corresponds to a version of generalized nested logit models where the no purchase option is offered within multiple nests. When we also impose the restriction that the dissimilarity parameters of all of the nests are equal to each other, the resulting choice model is referred to as the cross nested logit model. [13] use the cross nested logit model to include a no purchase option within each nest. The choice model in [13] precisely corresponds to the partially-captured nests that we consider in this paper.

A final reason to consider the case with partially-captured nests is that certain products may have to be included in the offered assortment, in which case, the parameter  $v_{i0}$  can be used to capture the total preference weight of the products that have to be included in the offered assortment. Naturally, there is a revenue contribution associated with the products that have to be included in the offered assortment and the definition of the expected revenue  $R_i(S_i)$  above does not keep track of this revenue contribution. However, it turns out that we can follow the same line of reasoning that we use to deal with partially-captured nests to deal with the case where certain products have to be included in the offered assortment. Thus, by building on our treatment of partially-captured nests, we can address assortment optimization problems where certain products are already included in the offered assortment and we need to decide which additional products should be offered.

In our nested logit model, the dissimilarity parameters  $(\gamma_1, \dots, \gamma_m)$  are assumed to be constants. To understand this assumption, it is useful to view the nested

logit model as a random utility-based choice model. In particular, consider the case where a customer associates random utilities with the products and with the no purchase options. The random utilities have a multi-dimensional generalized extreme value distribution. The utility of product  $j$  in nest  $i$  has mean  $\bar{u}_{ij}$  and unit variance. The utilities of the options in different nests are independent of each other. The customer, being a utility maximizer, follows the option with the largest utility. In this case, we can show that the probability of choosing a particular product under this random utility-based choice model has the same form specified by the nested logit model; see [40]. To obtain the nested logit model corresponding to this random utility-based choice model, we need to set the preference weight  $v_{ij}$  of product  $j$  in nest  $i$  as  $e^{\bar{u}_{ij}/\gamma_i}$ . The correlation structure between the random utilities of the options in nest  $i$  determines the value of the dissimilarity parameter  $\gamma_i$  of nest  $i$ . Thus, the means and correlation structure of the random utilities are the primitives of a random utility-based choice model that is consistent with the nested logit model. In our nested logit model, we assume that the means and correlations of the random utilities are fixed, implying that the preference weights and the dissimilarity parameters are fixed as well. It is possible to consider richer choice models where the means and correlations of the random utilities depend on the offered assortment, in which case, the preference weights and the dissimilarity parameters depend on the offered assortment as well, but this extension makes it difficult to solve the corresponding assortment optimization problem. We expand on this issue in Section 3.11.

### 3.4 Linear Programming Representation

The assortment optimization problem in (3.2) is of a combinatorial nature. In this section, we present a linear programming formulation of this problem. The

linear programming formulation is not too useful directly as a computational tool since its number of constraints grows exponentially with the number of products. However, it turns out that we can build on the linear programming formulation to develop a general approximation result for problem (3.2). The approximation methods that we propose throughout the paper are tightly related to this general approximation result.

To formulate problem (3.2) as a linear program, we first observe that this problem is equivalent to  $\min\{x : x \geq \sum_{i \in M} Q_i(S_1, \dots, S_m) R_i(S_i) \forall (S_1, \dots, S_m) \text{ with } S_i \subset N, i \in M\}$ . By using the definition of  $Q_i(S_1, \dots, S_m)$  in (3.1), we can write the constraints in this problem as

$$v_0 x \geq \sum_{i \in M} (V_i(S_i)^{\gamma_i} R_i(S_i) - V_i(S_i)^{\gamma_i} x) \quad \forall (S_1, \dots, S_m) \text{ with } S_i \subset N, i \in M,$$

which, in turn, are equivalent to the single constraint

$$v_0 x \geq \max_{(S_1, \dots, S_m): S_i \subset N, i \in M} \left\{ \sum_{i \in M} V_i(S_i)^{\gamma_i} (R_i(S_i) - x) \right\}.$$

The key observation is that the optimization problem on the right side of the constraint above decomposes by the nests and the constraint can be written as

$$v_0 x \geq \sum_{i \in M} \max_{S_i \subset N} V_i(S_i)^{\gamma_i} (R_i(S_i) - x).$$

Therefore, problem (3.2) is equivalent to

$$\begin{aligned} \min \quad & x \\ \text{s.t.} \quad & v_0 x \geq \sum_{i \in M} \max_{S_i \subset N} V_i(S_i)^{\gamma_i} (R_i(S_i) - x), \end{aligned}$$

where the only decision variable is  $x$ . Noting that  $Z^*$  is the optimal objective value of problem (3.2), the discussion so far implies that if  $x^*$  is the optimal solution to the problem above, then the optimal objective value of this problem is also  $x^*$  and we have  $x^* = Z^*$ . The constraint of the problem above is nonlinear, but to



linearize this constraint, we can define the decision variables  $y = (y_1, \dots, y_m)$  as  $y_i = \max_{S_i \subset N} V_i(S_i)^{\gamma_i} (R_i(S_i) - x)$  and write the problem as

$$\begin{aligned}
\min \quad & x & (3.3) \\
\text{s.t.} \quad & v_0 x \geq \sum_{i \in M} y_i \\
& y_i \geq V_i(S_i)^{\gamma_i} (R_i(S_i) - x) \quad \forall S_i \subset N, i \in M,
\end{aligned}$$

where the decision variables are  $(x, y)$ . Problem (3.3) is a linear program with  $1 + m$  decision variables and  $1 + m 2^n$  constraints. When  $v_0 = 0$ , the first constraint above reads  $\sum_{i \in M} y_i \leq 0$ , but the second set of constraints prevent  $x$  from becoming arbitrarily small because if  $x$  becomes arbitrarily small, then the decision variables  $(y_1, \dots, y_m)$  would take arbitrarily large values and we cannot satisfy the constraint  $\sum_{i \in M} y_i \leq 0$ . Therefore, problem (3.3) continues to apply when the preference weight  $v_0$  of the no purchase option is zero. Another useful observation is that the number of possible assortments in problem (3.2) is  $2^{mn}$ , which increases exponentially in both the number of nests and the number of products in each nest. In contrast, the numbers of decision variables and constraints in problem (3.3) grow linearly with the number of nests, and problem (3.3) can be tractable when the number of products in each nest is relatively small, irrespective of the number of nests. When the number of products in each nest is large, a possible solution approach for problem (3.3) is to use column generation on its dual, but due to the presence of the dissimilarity coefficient in the second set of constraints, the column generation subproblem is nonlinear and this renders column generation intractable.

Although problem (3.3) is difficult to solve when the number of products in each nest is large, we can build on this problem to develop a general approximation method. Assume that we identify a collection of candidate assortments  $\{A_{it} :$

$t \in \mathcal{T}_i$  that we may consider offering in nest  $i$ , where we have  $A_{it} \subset N$  for all  $t \in \mathcal{T}_i$ . We are interested in finding a combination of these assortments for the different nests so that the combined assortment provides the largest possible expected revenue. In other words, we are interested in finding the assortment that provides the largest expected revenue when we consider assortments of the form  $(S_1, \dots, S_m)$  with  $S_i \in \{A_{it} : t \in \mathcal{T}_i\}$  for all  $i \in M$ . This problem can be formulated as the linear program

$$\begin{aligned}
\min \quad & x & (3.4) \\
\text{s.t.} \quad & v_0 x \geq \sum_{i \in M} y_i \\
& y_i \geq V_i(S_i)^{\gamma_i} (R_i(S_i) - x) \quad \forall S_i \in \{A_{it} : t \in \mathcal{T}_i\}, i \in M.
\end{aligned}$$

The number of decision variables in problem (3.4) is still  $1 + m$ . The number of constraints is  $1 + \sum_{i \in M} |\mathcal{T}_i|$ , which can be reasonable when the collections of candidate assortments are not too large.

We now provide some observations to develop a general approximation result by building on problem (3.4). The constraints in problem (3.4) can succinctly be written as

$$v_0 x \geq \sum_{i \in M} \max_{S_i \in \{A_{it} : t \in \mathcal{T}_i\}} V_i(S_i)^{\gamma_i} (R_i(S_i) - x).$$

We will now argue that the constraint above must be satisfied as equality at an optimal solution. Let  $(\hat{x}, \hat{y})$  be an optimal solution to problem (3.4) and suppose there is a gap. We can then decrease the value of  $\hat{x}$  without violating the constraint, thereby obtaining a strictly better solution to problem (3.4) than  $\hat{x}$ , establishing the claim. Therefore, letting  $\hat{S}_i$  be the solution to the maximization problem on the right side of the constraint above with  $x = \hat{x}$ , it must be the case that  $v_0 \hat{x} =$

$\sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i} (R_i(\hat{S}_i) - \hat{x})$ . Solving for  $\hat{x}$ , we obtain

$$\hat{x} = \frac{\sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i} R_i(\hat{S}_i)}{v_0 + \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i}} = \sum_{i \in M} Q_i(\hat{S}_1, \dots, \hat{S}_m) R_i(\hat{S}_i) = \Pi(\hat{S}_1, \dots, \hat{S}_m).$$

Consequently, if  $(\hat{x}, \hat{y})$  is an optimal solution to problem (3.4) and  $\hat{S}_i$  is an optimal solution to the problem  $\max_{S_i \in \{A_{it} : t \in \mathcal{T}_i\}} V_i(S_i)^{\gamma_i} (R_i(S_i) - \hat{x})$ , then the expected revenue obtained by offering the assortment  $(\hat{S}_1, \dots, \hat{S}_m)$  is precisely  $\hat{x}$ .

We observe that problem (3.4) includes only a subset of the constraints in problem (3.3), which implies that problem (3.4) is a relaxed version of problem (3.3). Therefore, if we let  $(x^*, y^*)$  and  $(\hat{x}, \hat{y})$  respectively be the optimal solutions to problems (3.3) and (3.4), then we have  $Z^* = x^* \geq \hat{x}$ . Furthermore, for some  $\alpha$  and  $\beta$ , if we can show that  $(\alpha \hat{x}, \beta \hat{y})$  is a feasible solution to problem (3.3), then we also obtain  $\alpha \hat{x} \geq Z^* = x^* \geq \hat{x}$ , which implies that the expected revenue obtained by offering the assortment  $(\hat{S}_1, \dots, \hat{S}_m)$  as defined above deviates from the optimal expected revenue by no more than a factor of  $\alpha$ . We collect these observations in the following theorem.

**Theorem 3.4.1.** *Let  $(\hat{x}, \hat{y})$  be an optimal solution to problem (3.4), and for all  $i \in M$ , let  $\hat{S}_i$  be an optimal solution to the problem*

$$\max_{S_i \in \{A_{it} : t \in \mathcal{T}_i\}} V_i(S_i)^{\gamma_i} (R_i(S_i) - \hat{x}). \quad (3.5)$$

*If  $(\alpha \hat{x}, \beta \hat{y})$  is a feasible solution to problem (3.3) for some  $\alpha$  and  $\beta$ , then the expected revenue obtained by offering the assortment  $(\hat{S}_1, \dots, \hat{S}_m)$  deviates from the optimal objective value of problem (3.2) by no more than a factor of  $\alpha$ . In other words, letting  $\hat{Z} = \Pi(\hat{S}_1, \dots, \hat{S}_m)$ , we have  $\alpha \hat{Z} \geq Z^* \geq \hat{Z}$ .*

Theorem 3.4.1 provides sufficient conditions under which we can stitch together a good assortment from the collections of candidate assortments  $\{A_{it} : t \in \mathcal{T}_i\}$  for  $i \in M$ . The thought process we used to reach Theorem 3.4.1 will be critical

throughout the paper. In particular, we will design collections of assortments such that if  $(\hat{x}, \hat{y})$  is the optimal solution to problem (3.4) with these collections of assortments, then  $(\alpha \hat{x}, \beta \hat{y})$  ends up being a feasible solution to problem (3.3) for some  $\alpha$  and  $\beta$ . In that case, we can solve problem (3.4) with these collections of assortments to obtain the optimal solution  $(\hat{x}, \hat{y})$ . Letting  $\hat{S}_i$  be the assortment that solves problem (3.5), Theorem 3.4.1 implies that the expected revenue from the assortment  $(\hat{S}_1, \dots, \hat{S}_m)$  deviates from the optimal expected revenue by at most a factor of  $\alpha$ .

We use various collections of candidate assortments. One possibility is to use the assortments that include a certain number of products with the largest revenues. This class of assortments is known to be optimal when the customer choices are governed by the multinomial logit model and [52] give a performance guarantee for such assortments when the underlying choice model is the multinomial logit model with multiple customer types. In the next section, we show that this class of assortments is still optimal under the nested logit model as long as we only have competitive products and fully-captured nests, but in general, we may need to look beyond this class to find good solutions for the assortment optimization problem we are interested in.

### 3.5 Competitive Products and Fully-Captured Nests

In this section, we focus on instances of the assortment optimization problem in (3.2) with  $\gamma_i \leq 1$  and  $v_{i0} = 0$  for all  $i \in M$ . For this case, we show that there exists an optimal solution  $(S_1^*, \dots, S_m^*)$  to problem (3.2) such that each one of the assortments  $S_1^*, \dots, S_m^*$  is either the empty assortment or is of the form  $S_i^* = \{1, 2, \dots, j\}$  for some  $j \in N$ . Noting that the products in each nest are ordered such that  $r_{i1} \geq r_{i2} \geq \dots \geq r_{in}$ , this result implies that an optimal assortment in

each nest includes a certain number of products with the largest revenues. We call such assortments nested-by-revenue assortments. For notational brevity, we use  $N_{ij}$  to denote the nested-by-revenue assortment that includes the first  $j$  products with the largest revenues in nest  $i$ . In other words, we have  $N_{ij} = \{1, 2, \dots, j\}$  for all  $i \in M, j \in N$ . For notational uniformity, we also let  $N_{i0} = \emptyset$  and  $N_+ = N \cup \{0\}$ , in which case, our goal is to show that an optimal solution  $(S_1^*, \dots, S_m^*)$  to problem (3.2) is of the form  $S_i^* = N_{ij}$  for some  $j \in N_+$ . Throughout this section, we assume without loss of generality that  $v_0 > 0$ . Otherwise, since we have  $v_{i0} = 0$  for all  $i \in M$ , it is optimal to offer only one product with the largest revenue  $\max_{i \in M} r_{i1}$  over all nests and this product would be purchased with probability one.

The following proposition shows that if it is optimal to offer a nonempty assortment in a nest, then the expected revenue from this nest should at least be equal to the optimal expected revenue over all nests.

**Proposition 1.** *If  $(S_1^*, \dots, S_m^*)$  is an optimal solution to problem (3.2) and  $S_i^* \neq \emptyset$ , then  $R_i(S_i^*) \geq Z^*$ .*

*Proof.* To get a contradiction, assume that  $S_i^* \neq \emptyset$  and  $R_i(S_i^*) < Z^*$ . For notational convenience, let  $R_l = R_l(S_l^*)$  and  $q_l = V_l(S_l^*)^{\gamma_l}$  for all  $l \in M$ . Thus, we have  $Q_i(S_1^*, \dots, S_m^*) = q_i / (v_0 + \sum_{l \in M} q_l)$  and we can write the optimal expected revenue as

$$\begin{aligned} Z^* &= \Pi(S_1^*, \dots, S_m^*) = \frac{\sum_{l \in M} q_l R_l}{v_0 + \sum_{l \in M} q_l} \\ &= \frac{v_0 + \sum_{l \in M, l \neq i} q_l}{v_0 + \sum_{l \in M} q_l} \frac{\sum_{l \in M, l \neq i} q_l R_l}{v_0 + \sum_{l \in M, l \neq i} q_l} + \frac{q_i R_i}{v_0 + \sum_{l \in M} q_l} \\ &= \frac{v_0 + \sum_{l \in M, l \neq i} q_l}{v_0 + \sum_{l \in M} q_l} \Pi(S_1^*, \dots, S_{i-1}^*, \emptyset, S_{i+1}^*, \dots, S_m^*) + \frac{q_i}{v_0 + \sum_{l \in M} q_l} R_i. \end{aligned}$$

Noting that  $v_0 > 0$  and  $q_i = V_i(S_i^*)^{\gamma_i} > 0$ , the equality above shows that  $Z^*$  is a nontrivial convex combination of  $\Pi(S_1^*, \dots, S_{i-1}^*, \emptyset, S_{i+1}^*, \dots, S_m^*)$  and  $R_i =$

$R_i(S_i^*)$ . So, having  $R_i < Z^*$  implies that  $\Pi(S_1^*, \dots, S_{i-1}^*, \emptyset, S_{i+1}^*, \dots, S_m^*) > Z^*$  contradicting the fact that  $Z^*$  is the optimal expected revenue.  $\square$

In the following lemma, we show that the products with small revenues can be removed from the assortment without degrading the performance.

**Lemma 3.5.1.** *Assume that  $Z = \Pi(S_1, \dots, S_m)$  for some assortment  $(S_1, \dots, S_m)$  and there exists a product  $j \in S_i$  that satisfies  $r_{ij} < \gamma_i Z + (1 - \gamma_i) R_i(S_i)$  and  $R_i(S_i) \geq Z$ . Then, removing product  $j$  from  $S_i$  yields a strictly larger expected revenue than  $Z$ .*

*Proof.* Let  $\hat{S}_i$  be the assortment constructed by removing product  $j$  from  $S_i$ . We show that the assortment  $(S_1, \dots, S_{i-1}, \hat{S}_i, S_{i+1}, \dots, S_m)$  provides an expected revenue of  $\hat{Z}$  satisfying  $\hat{Z} > Z$ . For notational convenience, let  $\hat{R}_i = R_i(\hat{S}_i)$ ,  $\hat{q}_i = V_i(\hat{S}_i)^{\gamma_i}$ ,  $R_l = R_l(S_l)$  and  $q_l = V_l(S_l)^{\gamma_l}$  for all  $l \in M$ . Following an argument similar to the one in the proof of Proposition 1, we can write the expected revenue from the assortment  $(S_1, \dots, S_m)$  as

$$\begin{aligned} Z = \Pi(S_1, \dots, S_m) &= \frac{\sum_{l \in M} q_l R_l}{v_0 + \sum_{l \in M} q_l} \\ &= \frac{v_0 + \sum_{l \in M, l \neq i} q_l + \hat{q}_i}{v_0 + \sum_{l \in M} q_l} \frac{\sum_{l \in M, l \neq i} q_l R_l + \hat{q}_i \hat{R}_i}{v_0 + \sum_{l \in M, l \neq i} q_l + \hat{q}_i} + \frac{q_i R_i - \hat{q}_i \hat{R}_i}{v_0 + \sum_{l \in M} q_l} \\ &= \frac{v_0 + \sum_{l \in M, l \neq i} q_l + \hat{q}_i}{v_0 + \sum_{l \in M} q_l} \Pi(S_1, \dots, S_{i-1}, \hat{S}_i, S_{i+1}, \dots, S_m) + \frac{q_i - \hat{q}_i}{v_0 + \sum_{l \in M} q_l} \frac{q_i R_i - \hat{q}_i \hat{R}_i}{q_i - \hat{q}_i}. \end{aligned}$$

Therefore,  $Z$  is a convex combination of  $\hat{Z} = \Pi(S_1, \dots, S_{i-1}, \hat{S}_i, S_{i+1}, \dots, S_m)$  and  $(q_i R_i - \hat{q}_i \hat{R}_i)/(q_i - \hat{q}_i)$ . In this case, the desired result follows if we can show that  $Z > (q_i R_i - \hat{q}_i \hat{R}_i)/(q_i - \hat{q}_i)$ . In the rest of the proof, we equivalently show that  $q_i R_i - \hat{q}_i \hat{R}_i < (q_i - \hat{q}_i) Z$ .

Let  $\alpha = V_i(\hat{S}_i)/V_i(S_i)$ , so  $\hat{q}_i = \alpha^{\gamma_i} q_i$ . Using the fact that  $v_{ij} = V_i(S_i) - V_i(\hat{S}_i)$ , we can write  $\hat{R}_i$  as

$$\hat{R}_i = \frac{\sum_{k \in \hat{S}_i} r_{ik} v_{ik}}{V_i(\hat{S}_i)} = \frac{\sum_{k \in S_i} r_{ik} v_{ik} - r_{ij} (V_i(S_i) - V_i(\hat{S}_i))}{\alpha V_i(S_i)} = \frac{1}{\alpha} R_i - \frac{1 - \alpha}{\alpha} r_{ij}.$$

Therefore,  $q_i R_i - \hat{q}_i \hat{R}_i < (q_i - \hat{q}_i) Z$  holds if and only of

$$q_i R_i - \alpha^{\gamma_i} q_i \left[ \frac{1}{\alpha} R_i - \frac{1 - \alpha}{\alpha} r_{ij} \right] < (q_i - \alpha^{\gamma_i} q_i) Z.$$

Arranging the terms in the expression above, we observe that  $q_i R_i - \hat{q}_i \hat{R}_i < (q_i - \hat{q}_i) Z$  holds if and only if  $r_{ij} < g(\alpha) Z + (1 - g(\alpha)) R_i$ , where  $g(\alpha) = (1 - \alpha^{\gamma_i}) / (\alpha^{\gamma_i - 1} - \alpha^{\gamma_i})$ . By the hypothesis, we have  $r_{ij} < \gamma_i Z + (1 - \gamma_i) R_i$ . So, it is enough to show that  $\gamma_i Z + (1 - \gamma_i) R_i \leq g(\alpha) Z + (1 - g(\alpha)) R_i$ . Since both sides of the last inequality are convex combinations of  $Z$  and  $R_i$  and  $Z \leq R_i$  by the hypothesis, the inequality holds as long as  $g(\alpha) \leq \gamma_i$ . However, the last relationship is true because  $g(\alpha)$  is increasing in  $\alpha$  when  $\gamma_i \leq 1$  and by L'Hopital's rule,  $g(\alpha) \leq \lim_{\alpha \uparrow 1} g(\alpha) = \gamma_i$ .  $\square$

Lemma 3.5.1 gives us a mechanism to remove certain products with small revenues without reducing the expected revenue from an assortment. To see a useful implication of Proposition 1 and Lemma 3.5.1, assume that  $(S_1^*, \dots, S_m^*)$  is an optimal solution to problem (3.2) with  $S_i^* \neq \emptyset$ . We must have  $R_i(S_i^*) \geq Z^*$  by Proposition 1. In this case, we must have  $r_{ij} \geq \gamma_i Z^* + (1 - \gamma_i) R_i(S_i^*)$  for all  $j \in S_i^*$  by Lemma 3.5.1. Otherwise, we can remove a product from  $S_i^*$  and obtain an assortment that provides a strictly larger expected revenue than  $Z^*$ . We use this observation in the following theorem to show that nested-by-revenue assortments provide an optimal solution to problem (3.2).

**Theorem 3.5.2.** *There exists an optimal solution  $(S_1^*, \dots, S_m^*)$  to problem (3.2) such that, for all  $i \in M$ , we have  $S_i^* = N_{ij}$  for some  $j \in N_+$ .*

*Proof.* Assume that  $(S_1^*, \dots, S_m^*)$  is an optimal solution to problem (3.2) providing an expected revenue of  $Z^*$  and that there is a nest  $i$  such that  $S_i^*$  contains product  $j$  but not product  $k$  with  $k < j$ . Let  $\hat{S}_i$  be the assortment constructed by adding product  $k$  to  $S_i^*$ . Using  $\hat{Z}$  to denote the expected revenue from the assortment  $(S_1^*, \dots, S_{i-1}^*, \hat{S}_i, S_{i+1}^*, \dots, S_m^*)$ , we show that  $\hat{Z} \geq Z^*$ . Therefore, the assortment

$(S_1^*, \dots, S_{i-1}^*, \hat{S}_i, S_{i+1}^*, \dots, S_m^*)$  must also be optimal. Repeating the argument until the assortments for all nests are of the form  $\{1, 2, \dots, j\}$  for some  $j \in N_+$  establishes the result.

Similar to the notation in the proof of Lemma 3.5.1, let  $\hat{R}_i = R_i(\hat{S}_i)$ ,  $R_i = R_i(S_i^*)$ ,  $\hat{q}_i = V_i(\hat{S}_i)^{\gamma_i}$  and  $q_i = V_i(S_i^*)^{\gamma_i}$ . The main idea is to use an argument similar to the one in the proof of Lemma 3.5.1 to write  $\hat{Z} = \Pi(S_1^*, \dots, S_{i-1}^*, \hat{S}_i, S_{i+1}^*, \dots, S_m^*)$  as a convex combination of  $Z^*$  and  $(\hat{q}_i \hat{R}_i - q_i R_i)/(\hat{q}_i - q_i)$ . In this case, the desired result follows if we can show that  $Z^* \leq (\hat{q}_i \hat{R}_i - q_i R_i)/(\hat{q}_i - q_i)$ . We equivalently show that  $(\hat{q}_i - q_i) Z^* \leq \hat{q}_i \hat{R}_i - q_i R_i$ . Let  $\alpha = V_i(S_i^*)/V_i(\hat{S}_i)$ , so  $\hat{q}_i = q_i/\alpha^{\gamma_i}$ . Using the fact that  $v_{ik} = V_i(\hat{S}_i) - V_i(S_i^*)$ , we write  $\hat{R}_i$  as

$$\hat{R}_i = \frac{\sum_{j' \in \hat{S}_i} r_{ij'} v_{ij'}}{V_i(\hat{S}_i)} = \alpha \frac{\sum_{j' \in S_i^*} r_{ij'} v_{ij'} + r_{ik} (V_i(\hat{S}_i) - V_i(S_i^*))}{V_i(S_i^*)} = \alpha R_i + (1 - \alpha) r_{ik}.$$

Therefore,  $(\hat{q}_i - q_i) Z^* \leq \hat{q}_i \hat{R}_i - q_i R_i$  holds if and only if

$$\left[ \frac{q_i}{\alpha^{\gamma_i}} - q_i \right] Z^* \leq \frac{q_i}{\alpha^{\gamma_i}} (\alpha R_i + (1 - \alpha) r_{ik}) - q_i R_i.$$

Arranging the terms in the expression above, we observe that  $(\hat{q}_i - q_i) Z^* \leq \hat{q}_i \hat{R}_i - q_i R_i$  holds if and only if  $h(\alpha) Z^* + (1 - h(\alpha)) R_i \leq r_{ik}$ , where we let  $h(\alpha) = (1 - \alpha^{\gamma_i})/(1 - \alpha)$ . From the discussion that follows Lemma 3.5.1, since  $j \in S_i^*$ , we know that  $\gamma_i Z^* + (1 - \gamma_i) R_i \leq r_{ij} \leq r_{ik}$ , where the last inequality follows from the fact that  $k < j$ . So, it is enough to show that  $h(\alpha) Z^* + (1 - h(\alpha)) R_i \leq \gamma_i Z^* + (1 - \gamma_i) R_i$ . Since both sides of the last inequality are convex combinations of  $Z^*$  and  $R_i$  and  $Z^* \leq R_i$  by Proposition 1, the inequality holds if and only if  $h(\alpha) \geq \gamma_i$ . However, the last relationship is true because  $h(\alpha)$  is decreasing in  $\alpha$  and by L'Hoptial's rule,  $h(\alpha) \geq \lim_{\alpha \uparrow 1} h(\alpha) = \gamma_i$ .  $\square$

Theorem 3.5.2 shows that we can construct an optimal solution to problem (3.2) by only considering the nested-by-revenue assortments. To find the best combination of such assortments for the different nests, we can make use of problem



(3.4). In particular, we replace the collection of candidate assortments  $\{A_{it} : t \in \mathcal{T}_i\}$  in problem (3.4) with the nested-by-revenue assortments  $\{N_{ij} : j \in N_+\}$  and solve problem (3.4) to find the best combination of nested-by-revenue assortments for the different nests. By Theorem 3.5.2, this best combination has to be an optimal solution to problem (3.2). In this way, we can find an optimal solution to problem (3.2) by solving a linear program with  $1 + m$  decision variables and  $1 + m(1 + n)$  constraints.

### 3.6 Possibly Synergistic Products and Fully-Captured Nests

In this section, we focus on instances of the assortment optimization problem in (3.2), where we do not have any restrictions on the dissimilarity parameters  $(\gamma_1, \dots, \gamma_m)$  of the nests, but we still have  $v_{i0} = 0$  for all  $i \in M$ . We show that allowing the dissimilarity parameters for the nests to take on values larger than one changes the structure of problem (3.2) drastically. In particular, the result that we establish in the previous section does not necessarily hold when the dissimilarity parameters of the nests can take on arbitrary values and the nested-by-revenue assortments are no longer optimal. In the next section, we first characterize the computational complexity of the problem when we have no restrictions on the nest dissimilarity parameters. Following this result, we give a performance guarantee for the nested-by-revenue assortments. Throughout this discussion, we assume that  $\gamma_i > 1$  for some  $i \in M$ . Otherwise, nested-by-revenue assortments are optimal by Theorem 3.5.2.

### 3.6.1 Computational Complexity

We begin by giving an example that shows why nested-by-revenue assortments are no longer optimal when the dissimilarity parameters of the nests can take on values larger than one. This example also demonstrates that nested-by-revenue assortments can perform arbitrarily badly when the revenues and the preference weights of the products in a nest drastically differ from each other. Following this example, we establish that the assortment optimization problem in (3.2) is NP-hard whenever we allow  $\gamma_i > 1$  for some  $i \in M$ .

To give an example where nested-by-revenue assortments do not perform well, we consider an instance of problem (3.2) with a single nest. The preference weight for the option of not choosing any of the nests is  $v_0 = 1$ . The dissimilarity parameter of the nest is  $\gamma_1 = 2$ . There are three products in the nest. Letting  $\varepsilon \leq 1$  be a small positive number, the following table gives the revenues and the preference weights associated with the three products.

Product	1	2	3
Revenue	1	$\varepsilon^4$	0
Preference Weight	$\varepsilon^2$	$3/\varepsilon^2$	$1/\varepsilon$

Since there is only one nest with  $\gamma_1 = 2$ , the expected revenue from an assortment  $S_1 \subset \{1, 2, 3\}$  is

$$\Pi(S_1) = Q_1(S_1) R_1(S_1) = \frac{V_1(S_1)^2}{v_0 + V_1(S_1)^2} \frac{\sum_{j \in S_1} r_{1j} v_{1j}}{V_1(S_1)} = \frac{V_1(S_1) \sum_{j \in S_1} r_{1j} v_{1j}}{v_0 + V_1(S_1)^2}.$$

We compute and bound the expected revenues from the three nested-by-revenue

assortments as

$$\begin{aligned}\Pi(\{1\}) &= \frac{\varepsilon^2 \varepsilon^2}{1 + (\varepsilon^2)^2} \leq \varepsilon^4 \\ \Pi(\{1, 2\}) &= \frac{(\varepsilon^2 + 3/\varepsilon^2)(\varepsilon^2 + 3\varepsilon^2)}{1 + (\varepsilon^2 + 3/\varepsilon^2)^2} \leq \frac{(4/\varepsilon^2)4\varepsilon^2}{9/\varepsilon^4} = \frac{16}{9}\varepsilon^4 \\ \Pi(\{1, 2, 3\}) &= \frac{(\varepsilon^2 + 3/\varepsilon^2 + 1/\varepsilon)(\varepsilon^2 + 3\varepsilon^2)}{1 + (\varepsilon^2 + 3/\varepsilon^2 + 1/\varepsilon)^2} \leq \frac{(5/\varepsilon^2)4\varepsilon^2}{9/\varepsilon^4} = \frac{20}{9}\varepsilon^4,\end{aligned}$$

which implies that the expected revenue from the best nested-by-revenue assortment is no larger than  $\frac{20}{9}\varepsilon^4$ . On the other hand, the expected revenue from the assortment  $\{1, 3\}$  is given by

$$\Pi(\{1, 3\}) = \frac{(\varepsilon^2 + 1/\varepsilon)(\varepsilon^2)}{1 + (\varepsilon^2 + 1/\varepsilon)^2} \geq \frac{(1/\varepsilon)\varepsilon^2}{1 + (1/\varepsilon + 1/\varepsilon)^2} \geq \frac{\varepsilon}{1/\varepsilon^2 + (1/\varepsilon + 1/\varepsilon)^2} = \frac{1}{5}\varepsilon^3.$$

Thus, the optimal expected revenue exceeds the expected revenue from the best nested-by-revenue assortment by at least a factor of  $(\varepsilon^3/5)/(\frac{20}{9}\varepsilon^4) = 9/(100\varepsilon)$ . As  $\varepsilon \rightarrow 0$ , the performance of the best nested-by-revenue assortment becomes arbitrarily poor. The key observation in this problem instance is that the revenue associated with product 1 is quite large when compared with the other product revenues. Therefore, we would like to be able to sell product 1 with high probability. One can check that if we offer product 1 by itself, then the probability of purchase for product 1 is between  $\varepsilon^4/2$  and  $\varepsilon^4$ . On the other hand, if we offer products 1 and 2 together, then the probability of purchase for product 1 is always smaller than  $\varepsilon^4/2$ . Therefore, if we offer product 2 next to product 1, then the probability of purchase for product 1 goes down. In contrast, if we offer products 1 and 3, then it is possible to check that the probability of purchase for product 1 always exceeds  $\varepsilon^3/5$ , which is larger than  $\varepsilon^4$  for small values of  $\varepsilon$ . This observation indicates that product 3 acts as a synergistic product to product 1 and offering product 3 next to product 1 increases the probability of purchase for product 1. We also note that even if the revenue of product 3 was not zero but slightly negative, it would still be beneficial to add this product to the offered assortment, justifying a loss leader.

It turns out that if we allow the dissimilarity parameters of the nests to take on values larger than one, then not only the nested-by-revenue assortments cease to be optimal, but problem (3.2) becomes NP-hard. We devote the rest of this section to showing this result. To show the result we are interested in, we focus on the following decision-theoretic formulation of the assortment optimization problem.

*Assortment Feasibility.* Given a profit threshold  $K$ , is there an assortment  $(S_1, \dots, S_m)$  that provides an expected revenue of  $K$  or more for problem (3.2)?

To establish the NP-hardness of problem (3.2), Theorem 3.6.1 below shows that any instance of the partition problem, which is a well-known NP-hard problem as established in [23], can be reduced to an instance of the assortment feasibility problem. [52] also use a reduction from the partition problem to show the NP-hardness of an assortment optimization problem, but their choice model is the multinomial logit model with multiple customer types, rather than the nested logit model. The partition problem is described as follows.

*Partition.* Given integer-valued sizes  $(c_1, \dots, c_n)$  such that  $\sum_{j=1}^n c_j = 2T$  with  $T$  integer, can we find a subset  $S \subset \{1, \dots, n\}$  such that  $\sum_{j \in S} c_j = \sum_{j \in \{1, \dots, n\} \setminus S} c_j = T$ ?

**Theorem 3.6.1.** *If we allow the dissimilarity parameters for the nests to take on values larger than one, then the assortment feasibility problem is NP-hard.*

*Proof.* Assume that we are given any instance of the partition problem with sizes  $(c_1, \dots, c_n)$  and  $\sum_{j=1}^n c_j = 2T$ . We define an instance of the assortment feasibility problem as follows. There is only one nest. The preference weight for the option of not choosing any of the nests is  $v_0 = (1 + T)^2$ . The dissimilarity parameter of the nest is  $\gamma_1 = 2$ . There are  $n + 1$  products in the nest. The revenue associated with the first  $n$  products is given by  $r_{1j} = 0$  for all  $j = 1, \dots, n$ . The revenue associated with the last product is  $r_{1,n+1} = 2(1 + T)$ . The preference weights of

the first  $n$  products are given by  $v_{1j} = c_j$  for all  $j = 1, \dots, n$ . The preference weight associated with the last product is  $v_{1,n+1} = 1$ . We set the expected revenue threshold in the assortment feasibility problem as  $K = 1$ .

In the rest of the proof, we show that there exists an assortment that provides an expected revenue of  $K$  or more in the assortment feasibility problem if and only if there exists a subset  $S \subset \{1, \dots, n\}$  such that  $\sum_{j \in S} c_j = T$ . The first observation that if we want to get a positive expected revenue in the assortment feasibility problem, then we have to offer the last product with revenue  $2(1+T)$ . Therefore, the only question for the assortment feasibility problem is to choose a subset  $S$  among the products with zero revenues that makes sure that we obtain an expected revenue of  $K = 1$  or more. If we offer a subset  $S$  of the first  $n$  products together with the last product, then the expected revenue is  $Q_1(S \cup \{n+1\}) R_1(S \cup \{n+1\})$ , which evaluates to

$$\frac{(\sum_{j \in S} c_j + 1)^2}{(1+T)^2 + (\sum_{j \in S} c_j + 1)^2} \frac{2(1+T)}{(\sum_{j \in S} c_j + 1)}.$$

Therefore, there exists an assortment with an expected revenue of  $K = 1$  or more if and only if

$$\frac{(\sum_{j \in S} c_j + 1) 2(1+T)}{(1+T)^2 + (\sum_{j \in S} c_j + 1)^2} \geq 1.$$

Arranging the terms in the expression above, the inequality above is equivalent to

$$2(1+T) \sum_{j \in S} c_j + 2(1+T) \geq 1 + 2T + T^2 + 1 + 2 \sum_{j \in S} c_j + \left( \sum_{j \in S} c_j \right)^2,$$

which can equivalently be written as

$$\left( \sum_{j \in S} c_j \right)^2 - 2T \sum_{j \in S} c_j + T^2 \leq 0.$$

Since the last inequality is equivalent to  $(\sum_{j \in S} c_j - T)^2 \leq 0$ , there exists an assortment with an expected revenue of  $K = 1$  or more if and only if there exists

a subset  $S$  with  $(\sum_{j \in S} c_j - T)^2 \leq 0$ . However, the only way for the last inequality to hold is to have  $\sum_{j \in S} c_j = T$ . Therefore, finding an assortment that yields an expected revenue of  $K$  or more is equivalent to finding a subset  $S$  that satisfies  $\sum_{j \in S} c_j = T$  and the latter statement is precisely what the partition problem is interested in.  $\square$

### 3.6.2 Performance of Nested-by-Revenue Assortments

In the previous section, we show that nested-by-revenue assortments may not perform well when we allow the dissimilarity parameters of the nests to take on values larger than one. Our goal in this section is to develop a performance bound for this class of assortments as a function of the problem data. In particular, recalling that we use  $N_{ij}$  to denote the nested-by-revenue assortment that includes the first  $j$  products with the largest revenues in nest  $i$ , we show that by focusing only on the nested-by-revenue assortments, we can construct a solution to problem (3.2) whose expected revenue deviates from the optimal expected revenue by no more than a factor of

$$\max_{i \in M, j=2, \dots, n} \left\{ \frac{R_i(N_{i,j-1})}{R_i(N_{ij})} \wedge \frac{R_i(N_{ij})}{R_i(N_{i,j-1})} \frac{V_i(N_{ij})^{\gamma_i}}{V_i(N_{i,j-1})^{\gamma_i}} \right\}, \quad (3.6)$$

where we let  $a \wedge b = \min\{a, b\}$ . Before we show this result, it is useful to observe the implications of this performance guarantee.

To see the effect of the first term in the minimum operator in (3.6), assume that the revenues of the products within a nest are balanced in the sense that the largest and the smallest product revenues within a nest differ from each other by at most a factor of  $\rho$ . Since the preference weights of the no purchase options within the nests are zero,  $R_i(\cdot)$  is always smaller than the largest product revenue in nest  $i$  and is always larger than the smallest product revenue. Therefore, the

ratio  $R_i(N_{ij})/R_i(N_{i,j-1})$  in the expression above cannot exceed  $\rho$ . This observation implies that we expect the nested-by-revenue assortments to perform well when the revenues of the products within a particular nest are balanced. Note that the revenues of the products in different nests can still differ from each other arbitrarily. On the other hand, if we assume that the preference weights of the products within a nest are balanced in the sense that the largest and the smallest preference weights within a nest differ from each other by at most a factor of  $\kappa$ , then we obtain  $R_i(N_{i,j-1})/R_i(N_{ij}) = (\sum_{k=1}^{j-1} r_{ik} v_{ik} / \sum_{k=1}^j r_{ik} v_{ik}) (\sum_{k=1}^j v_{ik} / \sum_{k=1}^{j-1} v_{ik}) \leq \sum_{k=1}^j v_{ik} / \sum_{k=1}^{j-1} v_{ik} = V_i(N_{ij})/V_i(N_{i,j-1})$ . Since the assortments  $N_{ij}$  and  $N_{i,j-1}$  respectively include  $j$  and  $j-1$  products, the ratio  $V_i(N_{ij})/V_i(N_{i,j-1})$  is bounded from above by  $(j\kappa)/(j-1)$  and the latter expression does not exceed  $2\kappa$  for any  $j = 2, \dots, n$ . Therefore, (3.6) indicates that the nested-by-revenue assortments provide a performance guarantee of  $2\kappa$  as well, implying that these assortments are also expected to perform well when the preference weights of the products within a nest do not differ from each other drastically. Similar to the discussion for the product revenues, the preference weights of the products in different nests can still differ from each other arbitrarily. To see the effect of the second term in the minimum operator in (3.6), we observe that the expression in the curly brackets in (3.6) takes its largest value when the two terms of the minimum operator are equal to each other and this happens when

$$\frac{R_i(N_{i,j-1})}{R_i(N_{ij})} = \frac{V_i(N_{ij})^{\gamma_i/2}}{V_i(N_{i,j-1})^{\gamma_i/2}}.$$

Therefore, the value of the minimum operator in (3.6) is bounded from above by the expression on the right side above. Noting that we have  $V_i(N_{ij})^{\gamma_i/2}/V_i(N_{i,j-1})^{\gamma_i/2} \leq (2\kappa)^{\gamma_i/2}$ , the performance guarantee we give for nested-by-revenue assortments cannot exceed  $\max_{i \in M} (2\kappa)^{\gamma_i/2}$  either. When  $\gamma_i \leq 2$  for all  $i \in M$ , the latter performance guarantee is better than  $2\kappa$ .

We make use of Theorem 3.4.1 to establish the performance guarantee that we give in (3.6). Assume that we solve problem (3.4) after replacing the collection of candidate assortments  $\{A_{it} : t \in \mathcal{T}_i\}$  in the second set of constraints with the nested-by-revenue assortments  $\{N_{ij} : j \in N_+\}$ . Letting  $(\hat{x}, \hat{y})$  be the optimal solution we obtain in this fashion and using  $\alpha$  to denote the expression in (3.6), if we can show that  $(\alpha \hat{x}, \alpha \hat{y})$  is a feasible solution to problem (3.3), then Theorem 3.4.1 implies that we can focus only on nested-by-revenue assortments and still obtain an assortment whose expected revenue deviates from the optimal expected revenue by no more than a factor of  $\alpha$ .

To pursue this line of reasoning, we note that the second set of constraints in problem (3.3) can be written as  $y_i \geq \max_{S_i \subset N} V_i(S_i)^{\gamma_i} (R_i(S_i) - x)$  for all  $i \in M$ . Using the decision variables  $z_i = (z_{i1}, \dots, z_{in}) \in [0, 1]^n$ , we formulate a tighter version of problem (3.3) as

$$\begin{aligned}
\min \quad & x & (3.7) \\
\text{s.t.} \quad & v_0 x \geq \sum_{i \in M} y_i \\
& y_i \geq \max_{z_i \in [0, 1]^n} \left\{ \left( \sum_{j \in N} v_{ij} z_{ij} \right)^{\gamma_i} \left[ \frac{\sum_{j \in N} r_{ij} v_{ij} z_{ij}}{\sum_{j \in N} v_{ij} z_{ij}} - x \right] \right\} \quad \forall i \in M.
\end{aligned}$$

Note that if we imposed the constraint  $z_i \in \{0, 1\}^n$  in the maximization problem on the right side of the second set of constraints above, then problems (3.3) and (3.7) would be equivalent to each other. The way it is formulated, problem (3.3) is a relaxed version of problem (3.7) in the sense that any feasible solution  $(x, y)$  to problem (3.7) is also feasible to problem (3.3). In the following lemma, we study the optimal solution to the maximization problem on the right side of the second set of constraints above.



**Lemma 3.6.2.** *There exists an optimal solution  $z_i^*$  to the problem*

$$\max_{z_i \in [0,1]^n} \left\{ \left( \sum_{j \in N} v_{ij} z_{ij} \right)^{\gamma_i} \left[ \frac{\sum_{j \in N} r_{ij} v_{ij} z_{ij}}{\sum_{j \in N} v_{ij} z_{ij}} - x \right] \right\} \quad (3.8)$$

such that  $z_{i1}^* = 1, z_{i2}^* = 1, \dots, z_{i,k-1}^* = 1, z_{ik}^* \in [0, 1], z_{i,k+1}^* = 0, \dots, z_{in}^* = 0$  for some  $k = 1, \dots, n$ .

*Proof.* Assume that  $\hat{z}_i$  is an optimal solution to problem (3.8) and let  $C = \sum_{j \in N} v_{ij} \hat{z}_{ij}$ . In this case, note that an optimal solution to the continuous knapsack problem

$$\max \left\{ \sum_{j \in N} r_{ij} v_{ij} z_{ij} : \sum_{j \in N} v_{ij} z_{ij} = C, z_i \in [0, 1]^n \right\}$$

is also an optimal solution to problem (3.8). In the continuous knapsack problem above, the utility of item  $j$  is  $r_{ij} v_{ij}$  and the space consumption of item  $j$  is  $v_{ij}$ . Thus, we can solve this problem by sorting the products with respect to their utility-to-space consumption ratios and filling the knapsack starting from the item with the largest utility-to-space consumption ratio. Since the utility-to-space consumption ratio of item  $j$  is  $r_{ij}$  and the products are ordered such that  $r_{i1} \geq r_{i2} \geq \dots \geq r_{in}$ , there exists an optimal solution to the continuous knapsack problem above with the form given in the lemma.  $\square$

Except for at most one possible fractional component, Lemma 3.6.2 shows that a nested-by-revenue assortment is optimal for the maximization problem on the right side of the second set of constraints in problem (3.7). Thus, it is not too surprising that if we solve problem (3.4) after replacing the collection of candidate assortments  $\{A_{it} : t \in \mathcal{T}_i\}$  in the second set of constraints with the nested-by-revenue assortments  $\{N_{ij} : j \in N_+\}$  and obtain the optimal solution  $(\hat{x}, \hat{y})$ , then  $(\hat{x}, \hat{y})$  is almost feasible to problem (3.7). Since problem (3.3) is a relaxed version of problem (3.7), the solution  $(\hat{x}, \hat{y})$  would be almost feasible to problem (3.3) as well. We make this intuitive argument precise in the following theorem and show that

nested-by-revenue assortments provide the performance guarantee that we give in (3.6). We defer the proof of this result to the appendix.

**Theorem 3.6.3.** *Let  $(\hat{x}, \hat{y})$  be an optimal solution to problem (3.4) when we solve this problem after replacing the collection of candidate assortments  $\{A_{it} : t \in \mathcal{T}_i\}$  in the second set of constraints with the nested-by-revenue assortments  $\{N_{ij} : j \in N_+\}$ . Then, using  $\alpha$  to denote the expression in (3.6),  $(\alpha \hat{x}, \alpha \hat{y})$  is a feasible solution to problem (3.3).*

Theorem 3.6.3, along with Theorem 3.4.1, shows that if we only consider the nested-by-revenue assortments as candidate assortments, then we can construct a solution to problem (3.2) whose expected revenue deviates from the optimal expected revenue by at most a factor given in (3.6). Similar to the discussion at the end of Section 3.5, to find the best combination of nested-by-revenue assortments, we can solve problem (3.4) after replacing the collection of candidate assortments  $\{A_{it} : t \in \mathcal{T}_i\}$  with the nested-by-revenue assortments  $\{N_{ij} : j \in N_+\}$ . This amounts to solving a linear program with  $1+m$  decision variables and  $1+m(1+n)$  constraints.

### 3.7 Competitive Products and Partially-Captured Nests

In this section, we consider instances of the assortment optimization problem in (3.2) with  $\gamma_i \leq 1$  for all  $i \in M$ , but the preference weights of the no purchase options within the nests can take on arbitrary values. In other words, we allow  $(v_{10}, \dots, v_{m0})$  to take on strictly positive values so that a customer can leave without purchasing anything even after this customer chooses a particular nest. For this case, nested-by-revenue assortments are no longer optimal. As a matter of fact, problem (3.2) turns out to be NP-hard. However, we are able to characterize

a small class of assortments such that if we focus on this class of assortments, then we can construct a solution to problem (3.2) whose expected revenue deviates from the optimal expected revenue by no more than a factor of two.

In the following theorem, we show that the assortment optimization problem in (3.2) is NP-hard when we have  $v_{i0} > 0$  for some  $i \in M$ . Our proof technique is similar to the one in Section 3.6.1. We consider the assortment feasibility problem as defined in Section 3.6.1 and show that any instance of the partition problem can be reduced to an instance of the assortment feasibility problem. However, the specifics of the reduction are more involved and we give the proof of this theorem in the appendix.

**Theorem 3.7.1.** *If we allow the preference weights of the no purchase options within the nests to take on strictly positive values, then the assortment feasibility problem is NP-hard.*

The difficulty with partially-captured nests is that such nests do not allow Proposition 1 to hold, which implies that it may be optimal to offer a nonempty assortment in a partially-captured nest even if the expected revenue from this nest is below the optimal expected revenue. As a result, our line of reasoning in Section 3.5 does not hold. It is possible to show that if we have a mixture of partially-captured and fully-captured nests, then it is still optimal to offer nested-by-revenue assortments in the fully-captured nests, but it is not clear what to do for the partially-captured ones. Motivated by the computational complexity result in Theorem 3.7.1, we turn our attention to obtaining approximate solutions. In this section, we develop a tractable approach that obtains a solution to problem (3.2) whose expected revenue deviates from the optimal expected revenue by at most a factor of two. To that end, we use an alternative representation of problem (3.3). Using the decision variables  $z_i = (z_{i1}, \dots, z_{in})$ , we define  $K_i(\epsilon_i)$  as the

optimal objective value of the knapsack problem

$$K_i(\epsilon_i) = \max \left\{ \sum_{j \in N} r_{ij} v_{ij} z_{ij} : \sum_{j \in N} v_{ij} z_{ij} \leq \epsilon_i, z_i \in \{0, 1\}^n \right\}. \quad (3.9)$$

In this case, noting that  $V_i(S_i)$  and  $R_i(S_i)$  in the second set of constraints in problem (3.3) are respectively given by  $V_i(S_i) = v_{i0} + \sum_{j \in S_i} v_{ij}$  and  $R_i(S_i) = \sum_{j \in S_i} r_{ij} v_{ij} / V_i(S_i)$ , we consider the problem

$$\begin{aligned} \min \quad & x & (3.10) \\ \text{s.t.} \quad & v_0 x \geq \sum_{i \in M} y_i \\ & y_i \geq \max_{\epsilon_i \geq 0} \left\{ (v_{i0} + \epsilon_i)^{\gamma_i} \left[ \frac{K_i(\epsilon_i)}{v_{i0} + \epsilon_i} - x \right] \right\} \quad \forall i \in M. \end{aligned}$$

The following lemma shows that problem (3.10) is equivalent to problem (3.3).

**Lemma 3.7.2.** *Problems (3.3) and (3.10) are equivalent to each other in the sense that an optimal solution to one problem is also an optimal solution to the other.*

*Proof.* Noting that the second set of constraints in problem (3.3) can equivalently be written as  $y_i \geq \max_{S_i \subset N} V_i(S_i)^{\gamma_i} (R_i(S_i) - x)$  for all  $i \in M$ , the result follows if we can show that

$$\max_{S_i \subset N} V_i(S_i)^{\gamma_i} (R_i(S_i) - x) = \max_{\epsilon_i \geq 0} \left\{ (v_{i0} + \epsilon_i)^{\gamma_i} \left[ \frac{K_i(\epsilon_i)}{v_{i0} + \epsilon_i} - x \right] \right\}$$

for any  $x \geq 0$ . Let  $\zeta_L^*$  and  $\zeta_R^*$  respectively be the optimal objective values of the problems on the left and right side above. First, we show that  $\zeta_L^* \leq \zeta_R^*$ . Assume that  $S_i^*$  is an optimal solution to the problem on the left side above and define  $\epsilon_i^* = \sum_{j \in S_i^*} v_{ij}$ . The solution obtained by setting  $z_{ij} = 1$  for all  $j \in S_i^*$  and  $z_{ij} = 0$  otherwise is a feasible solution to problem (3.9) with  $\epsilon_i = \epsilon_i^*$ , which implies that  $K_i(\epsilon_i^*) \geq \sum_{j \in S_i^*} r_{ij} v_{ij}$ . Thus, if we evaluate the objective value of the problem on the right side above at  $\epsilon_i = \epsilon_i^*$ , then we obtain at least  $\zeta_L^*$ , in which case, we obtain  $\zeta_R^* \geq \zeta_L^*$ .

Second, we show that  $\zeta_L^* \geq \zeta_R^*$ . Let  $\epsilon_i^*$  be an optimal solution to the problem on the right side above and solve problem (3.9) after setting  $\epsilon_i = \epsilon_i^*$ . Letting  $z_i^*$  be the solution we obtain, we observe that we can assume without loss of generality that  $\sum_{j \in N} v_{ij} z_{ij}^* = \epsilon_i^*$ . To see this claim, if we have  $\sum_{j \in N} v_{ij} z_{ij}^* < \epsilon_i^*$ , then we can decrease the value of  $\epsilon_i^*$  to  $\hat{\epsilon}_i = \sum_{j \in N} v_{ij} z_{ij}^*$  while still preserving  $K_i(\epsilon_i^*) = K_i(\hat{\epsilon}_i)$ . In this case, using the fact that  $\gamma_i \leq 1$  and  $x \geq 0$ , we obtain

$$\begin{aligned} (v_{i0} + \epsilon_i^*)^{\gamma_i} \left[ \frac{K_i(\epsilon_i^*)}{v_{i0} + \epsilon_i^*} - x \right] &= \frac{K_i(\epsilon_i^*)}{(v_{i0} + \epsilon_i^*)^{1-\gamma_i}} - (v_{i0} + \epsilon_i^*)^{\gamma_i} x \\ &\leq \frac{K_i(\hat{\epsilon}_i)}{(v_{i0} + \hat{\epsilon}_i)^{1-\gamma_i}} - (v_{i0} + \hat{\epsilon}_i)^{\gamma_i} x = (v_{i0} + \hat{\epsilon}_i)^{\gamma_i} \left[ \frac{K_i(\hat{\epsilon}_i)}{v_{i0} + \hat{\epsilon}_i} - x \right], \end{aligned}$$

which shows that  $\hat{\epsilon}_i$  should also be an optimal solution to the problem on the right side above, establishing the claim. By using the solution  $z_i^*$ , we define the assortment  $S_i^*$  as  $S_i^* = \{j \in N : z_{ij}^* = 1\}$ . Since  $\sum_{j \in S_i^*} v_{ij} = \sum_{j \in N} v_{ij} z_{ij}^* = \epsilon_i^*$  and  $\sum_{j \in S_i^*} r_{ij} v_{ij} = \sum_{j \in N} r_{ij} v_{ij} z_{ij}^* = K_i(\epsilon_i^*)$ , the assortment  $S_i^*$  provides an objective value of  $\zeta_R^*$  for the problem on the left side above and we obtain  $\zeta_L^* \geq \zeta_R^*$ .  $\square$

To exploit the equivalence between problems (3.3) and (3.10) in a tractable fashion, we use the continuous relaxation of the knapsack problem in (3.9), which is given by

$$\hat{K}_i(\epsilon_i) = \max \left\{ \sum_{j \in N} r_{ij} v_{ij} z_{ij} : \sum_{j \in N} v_{ij} z_{ij} \leq \epsilon_i, 0 \leq z_{ij} \leq \mathbf{1}(v_{ij} \leq \epsilon_i) \forall j \in N \right\}, \quad (3.11)$$

where we use  $\mathbf{1}(\cdot)$  to denote the indicator function. Since problem (3.11) is a relaxation of problem (3.9), we have  $\hat{K}_i(\epsilon_i) \geq K_i(\epsilon_i)$ . The problem above is a continuous knapsack problem, where the utility of item  $j$  is  $r_{ij} v_{ij}$ , the space consumption of item  $j$  is  $v_{ij}$  and we can only consider the items whose space consumptions do not exceed  $\epsilon_i$ . Noting that the utility-to-space consumption ratio of item  $j$  is  $r_{ij}$ , we can solve this problem by sorting the products with respect to

their revenues and filling the knapsack starting from the product with the largest revenue, as long as we only consider the products whose preference weights do not exceed  $\epsilon_i$ . We let  $\hat{z}_i(\epsilon_i) = (\hat{z}_{i1}(\epsilon_i), \dots, \hat{z}_{in}(\epsilon_i))$  be an optimal solution to problem (3.11) that we obtain in this fashion. We observe that  $\hat{z}_i(\epsilon_i)$  has at most one fractional component. By using this solution, we define the assortment  $\hat{S}_i(\epsilon_i)$  as  $\hat{S}_i(\epsilon_i) = \{j \in N : \hat{z}_{ij}(\epsilon_i) = 1\}$ , which includes only the strictly positive and integer-valued components of  $\hat{z}_i(\epsilon_i)$ . We use the assortments  $\{\hat{S}_i(\epsilon_i) : \epsilon_i \in [0, \infty]\}$  as a collection of candidate assortments for nest  $i$ . We shortly show in this section that this collection of assortments includes no more than  $1 + n^2$  assortments and each one of these  $1 + n^2$  assortments can be identified in a tractable fashion.

To be able to obtain the performance guarantee of two, we augment the collection of assortments  $\{\hat{S}_i(\epsilon_i) : \epsilon_i \in [0, \infty]\}$  for nest  $i$  by the collection of singleton assortments  $\{\{j\} : j \in N\}$ . We solve problem (3.4) after replacing the collection of candidate assortments  $\{A_{it} : t \in \mathcal{T}_i\}$  in the second set of constraints with the collection of assortments  $\{\hat{S}_i(\epsilon_i) : \epsilon_i \in [0, \infty]\} \cup \{\{j\} : j \in N\}$ . Letting  $(\hat{x}, \hat{y})$  be the optimal solution to problem (3.4) that we obtain in this fashion, if we can show that  $(2\hat{x}, 2\hat{y})$  is a feasible solution to problem (3.3), then Theorem 3.4.1 implies that we can focus only on the assortments  $\{\hat{S}_i(\epsilon_i) : \epsilon_i \in [0, \infty]\} \cup \{\{j\} : j \in N\}$  for nest  $i$  and still obtain an assortment whose expected revenue deviates from the optimal expected revenue by no more than a factor of two. We pursue this result in the following theorem, but defer the proof of this result to the appendix.

**Theorem 3.7.3.** *Let  $(\hat{x}, \hat{y})$  be an optimal solution to problem (3.4) when we solve this problem after replacing the collection of candidate assortments  $\{A_{it} : t \in \mathcal{T}_i\}$  in the second set of constraints with the assortments  $\{\hat{S}_i(\epsilon_i) : \epsilon_i \in [0, \infty]\} \cup \{\{j\} : j \in N\}$ . Then,  $(2\hat{x}, 2\hat{y})$  is a feasible solution to problem (3.3).*

The equivalence between problems (3.3) and (3.10) given in Lemma 3.7.2 lays

out the connection between the assortment optimization problem we are interested in and the knapsack problem, as long as we have  $\gamma_i \leq 1$  for all  $i \in M$ . It is well-known that one can construct a solution to the knapsack problem in (3.9) by using its continuous relaxation in (3.11) and the objective value of this solution would deviate from the optimal objective value of the knapsack problem by no more than a factor of two; see [65]. The proof of Theorem 3.7.3 implicitly makes use of this result. Similarly, there exists a well-known fully polynomial-time approximation scheme for the knapsack problem, which can be found in [65]. Building on this fully polynomial-time approximation scheme, it is indeed possible to develop a fully polynomial-time approximation scheme for the assortment optimization problem we are interested in. We do not pursue the fully polynomial-time approximation scheme because this scheme can be developed by using an argument that is very similar to the preceding discussion in this section. The only difference is that instead of constructing approximate solutions to the knapsack problem in (3.9) by using its continuous relaxation, we construct approximate solutions by using the fully polynomial-time approximation scheme for knapsack problems.

In the remainder of this section, we argue that the collection of assortments  $\{\hat{S}_i(\epsilon_i) : \epsilon_i \in [0, \infty]\}$  includes no more than  $1 + n^2$  assortments and each one of these  $1 + n^2$  assortments can be identified in a tractable fashion. Fix  $\epsilon_i$  and consider the assortment  $\hat{S}_i(\epsilon_i)$ . The definition of  $\hat{z}_i(\epsilon_i)$  implies that  $\hat{S}_i(\epsilon_i)$  is a nested-by-revenue assortment as long as we focus only on the products whose preference weights do not exceed  $\epsilon_i$ . In other words, letting  $k$  be the number of products whose preference weights do not exceed  $\epsilon_i$ ,  $\hat{S}_i(\epsilon_i)$  is a nested-by-revenue assortment as long as we focus only on the products with the  $k$  smallest preference weights. Given that we focus only on the products with the  $k$  smallest preference weights, we use  $N_{ij}^k$  to denote the nested-by-revenue assortment that includes the

first  $j$  products with the largest revenues in nest  $i$ . Therefore,  $\hat{S}_i(\epsilon_i)$  must be one of the assortments  $\{N_{ij}^k : j = 0, \dots, k\}$ , where we let  $N_{i0}^k = \emptyset$  for notational uniformity. Since the only possible values for  $k$  are  $k = 1, \dots, n$ , it follows that  $\{\hat{S}_i(\epsilon_i) : \epsilon_i \in [0, \infty]\} \subset \{N_{ij}^k : k \in N, j = 0, \dots, k\}$ . The latter collection of assortments includes no more than  $1 + n^2$  distinct assortments, all of which can be easily be identified.

Theorem 3.7.3 shows that if we use the assortments  $\{\hat{S}_i(\epsilon_i) : \epsilon_i \in [0, \infty]\} \cup \{\{j\} : j \in N\}$  as a collection of candidate assortments for nest  $i$ , then we can stitch together from these candidate assortments a solution to problem (3.2) whose expected revenue deviates from the optimal expected revenue by at most a factor of two. Since we have  $\{\hat{S}_i(\epsilon_i) : \epsilon_i \in [0, \infty]\} \subset \{N_{ij}^k : k \in N, j = 0, \dots, k\}$ , using the assortments  $\{N_{ij}^k : k \in N, j = 0, \dots, k\} \cup \{\{j\} : j \in N\}$  as candidate assortments for nest  $i$  cannot degrade the performance guarantee of two. To find the best combination of these assortments, we can solve problem (3.4) after replacing the collection of candidate assortments  $\{A_{it} : t \in \mathcal{T}_i\}$  with the assortments  $\{N_{ij}^k : k \in N, j = 0, \dots, k\} \cup \{\{j\} : j \in N\}$ . This amounts to solving a linear program with  $1 + m$  decision variables and  $1 + m(1 + n + n^2)$  constraints.

### 3.8 Possibly Synergistic Products and Partially-Captured Nests

In this section, we consider the most general instances of the assortment optimization problem in (3.2), where we do not have any restrictions on the dissimilarity parameters  $(\gamma_1, \dots, \gamma_m)$  of the nests and the preference weights  $(v_{10}, \dots, v_{m0})$  of the no purchase options within the nests. We establish three results for these most general instances. First, we show that if the largest and the smallest preference



weights within a nest do not differ from each other by more than a factor of  $\kappa$ , then the collection of assortments developed in the previous section provides a performance guarantee of  $2\kappa$ . Second, using the links of our assortment problem to the partition problem, we give a pseudo-polynomial-time algorithm to obtain the optimal assortment. Third, letting  $\bar{\gamma} = \max_{i \in M} \gamma_i$  for notational brevity, for any given  $\delta > 1$ , we construct a collection of assortments that provides a performance guarantee of  $\delta^{2\bar{\gamma}+1}$ , but the computational work required to obtain this collection increases as  $\delta$  gets close to one. The important point about the last result is that it allows us to choose  $\delta$  close to one to obtain an assortment with an arbitrarily good performance guarantee, but choosing  $\delta$  close to one also increases the computational work. So, this result allows us to tradeoff computational work with performance guarantee. Throughout this section, we assume that  $\bar{\gamma} > 1$ . Otherwise, we can use the ideas in the previous section.

### 3.8.1 Performance Guarantee

We begin by showing that if the largest and the smallest preference weights within a nest differ from each other by at most a factor of  $\kappa$ , then the collection of assortments that we develop in the previous section provides a performance guarantee of  $2\kappa$ . It turns out that this result follows by synthesizing the results that are already given in Sections 3.6 and 3.7. In Section 3.6, we show that if we focus only on the nested-by-revenue assortments  $\{N_{ij} : j \in N_+\}$ , then we obtain the performance guarantee in (3.6) when we have possibly synergistic products and fully-captured nests. Noting the definition of  $N_{ij}^k$  in the previous section,  $N_{ij}^n$  is identical to the nested-by-revenue assortment  $N_{ij}$ . Therefore, we have  $\{N_{ij} : j \in N_+\} \subset \{N_{ij}^k : k \in N, j = 0, \dots, k\}$ , which implies that if we focus on the collection of assortments  $\{N_{ij}^k : k \in N, j = 0, \dots, k\}$ , then we still obtain

the performance guarantee in (3.6) when we have possibly synergistic products and fully-captured nests. On the other hand, in Section 3.7, we show that if we focus only on the collection of assortments  $\{N_{ij}^k : k \in N, j = 0, \dots, k\} \cup \{\{j\} : j \in N\}$ , then we obtain a performance guarantee of two when we have competitive products and partially-captured nests. In this case, using  $M^f$  and  $M^p$  to respectively denote the sets of fully-captured and partially-captured nests and letting  $a \vee b = \max\{a, b\}$ , it is possible to synthesize these results to show that if we focus only on the collection of assortments  $\{N_{ij}^k : k \in N, j = 0, \dots, k\} \cup \{\{j\} : j \in N\}$ , then we obtain a performance guarantee of

$$\max_{i \in M^f, j=2, \dots, n} \left\{ \frac{V_i(N_{ij})}{V_i(N_{i,j-1})} \right\} \vee \max_{i \in M^p, j=1, \dots, n} \left\{ \frac{V_i(N_{ij})}{V_i(N_{i,j-1})} \right\} \vee 2 \quad (3.12)$$

for the most general instances of the assortment optimization problem. Note that if  $i$  is a fully-captured nest, then  $V_i(N_{i0}) = V_i(\emptyset) = 0$  and we do not consider the term  $V_i(N_{i1})/V_i(N_{i0})$  in the expression above for fully-captured nests. To see the implication of the performance guarantee in (3.12), we observe that if the largest and the smallest preference weights within a nest differ from each other by at most a factor of  $\kappa$ , then we have  $V_i(N_{ij})/V_i(N_{i,j-1}) \leq (j\kappa)/(j-1)$  for fully-captured nests and  $V_i(N_{ij})/V_i(N_{i,j-1}) \leq (j+1)\kappa/j$  for partially-captured nests. Thus, the performance guarantee in (3.12) cannot exceed  $2\kappa \vee 2 = 2\kappa$ . This observation implies that if we use  $\{N_{ij}^k : k \in N, j = 0, \dots, k\} \cup \{\{j\} : j \in N\}$  as the collection of candidate assortments for nest  $i$ , then the best assortment that we can stitch together from these candidate assortments is expected to perform well when the preference weights within a nest do not differ too much from each other.

We can build on Theorem 3.4.1 to establish the performance guarantee in (3.12). Assume that we solve problem (3.4) after replacing the collection of candidate assortments  $\{A_{it} : t \in \mathcal{T}_i\}$  in the second set of constraints with the collection  $\{N_{ij}^k : k \in N, j = 0, \dots, k\} \cup \{\{j\} : j \in N\}$ . Letting  $(\hat{x}, \hat{y})$  be the optimal

solution we obtain in this fashion and using  $\beta$  to denote the expression in (3.12), if we can show that  $(\beta \hat{x}, \beta \hat{y})$  is a feasible solution to problem (3.3), then Theorem 3.4.1 implies that we can focus only on the collection of assortments  $\{N_{ij}^k : k \in N, j = 0, \dots, k\} \cup \{\{j\} : j \in N\}$  and obtain an assortment whose expected revenue deviates from the optimal expected revenue by no more than a factor of  $\beta$ . The following theorem shows this result. Its proof is given in the appendix.

**Theorem 3.8.1.** *Let  $(\hat{x}, \hat{y})$  be an optimal solution to problem (3.4) when we solve this problem after replacing the collection of candidate assortments  $\{A_{it} : t \in \mathcal{T}_i\}$  in the second set of constraints with the assortments  $\{N_{ij}^k : k \in N, j = 0, \dots, k\} \cup \{\{j\} : j \in N\}$ . Then, using  $\beta$  to denote the expression in (3.12),  $(\beta \hat{x}, \beta \hat{y})$  is a feasible solution to problem (3.3).*

Thus, similar to the discussion at the end of Section 3.7, since there are  $1+n+n^2$  assortments in the collection  $\{N_{ij}^k : k \in N, j = 0, \dots, k\} \cup \{\{j\} : j \in N\}$ , we can solve a linear program with  $1+m$  decision variables and  $1+m(1+n+n^2)$  constraints to obtain an assortment whose expected revenue deviates from the optimal expected revenue by at most a factor given in (3.12).

### 3.8.2 Near-Optimal Assortments

In this section, we give a pseudo-polynomial-time algorithm for our assortment problem. Following this result, we develop an approximation scheme that can tradeoff performance guarantee with computational work. Both of these results are based on viewing problem (3.3) from a knapsack perspective. In particular, we define  $G_i(\epsilon_i)$  as the optimal objective value of the knapsack problem

$$G_i(\epsilon_i) = \max_{S_i \subset N} \left\{ \sum_{j \in S_i} r_{ij} v_{ij} : v_{i0} + \sum_{j \in S_i} v_{ij} = \epsilon_i \right\}, \quad (3.13)$$

where we let  $G_i(\epsilon_i) = -\infty$  when the problem on the right side above is infeasible.

The second set of constraints in problem (3.3) are given by

$$y_i \geq V_i(S_i)^{\gamma_i} \left( \sum_{j \in S_i} r_{ij} v_{ij} / V_i(S_i) - x \right)$$

for all  $S_i \subset N$ ,  $i \in M$  and problem (3.13) finds the largest value of  $\sum_{j \in S_i} r_{ij} v_{ij}$  while keeping  $V_i(S_i)$  constant at  $\epsilon_i$ . Thus, problem (3.3) can equivalently be written as

$$\begin{aligned} \min \quad & x & (3.14) \\ \text{s.t.} \quad & v_0 x \geq \sum_{i \in M} y_i \\ & y_i \geq \epsilon_i^{\gamma_i} \left[ \frac{G_i(\epsilon_i)}{\epsilon_i} - x \right] \quad \forall \epsilon_i \geq 0, i \in M, \end{aligned}$$

where we treat  $0/0$  as zero in the second set of constraints. Letting  $v_i^L = v_{i0} + \min_{j \in N} v_{ij}$  and  $v_i^U = v_{i0} + \sum_{j \in N} v_{ij}$ , the definition of  $G_i(\epsilon_i)$  in (3.13) implies that  $G_i(\epsilon_i) = G_i(v_{i0})$  or  $G_i(\epsilon_i) = -\infty$  for all  $\epsilon_i < v_i^L$ , whereas  $G_i(\epsilon_i) = -\infty$  for all  $\epsilon_i > v_i^U$ . Therefore, it is enough to consider the values of  $\epsilon_i$  with  $\epsilon_i = v_{i0}$  or  $\epsilon_i \in [v_i^L, v_i^U]$  in the second set of constraints in problem (3.14).

Problem (3.13) can be visualized as an optimization version of the partition problem since it tries to find an assortment  $S_i$  that maximizes the objective value among all assortments whose preference weights add up to  $\epsilon_i$ . There is a well-known pseudo-polynomial-time algorithm for the partition problem; see [23]. This pseudo-polynomial-time algorithm operates under the assumption that all preference weights are integer-valued, in which case, the only values of  $\epsilon_i$  that render problem (3.13) feasible are integer-valued. By using a dynamic program, the pseudo-polynomial-time algorithm computes  $G_i(\epsilon_i)$  for all  $\epsilon_i \in [v_i^L, v_i^U]$  in  $O(n v_i^U)$  time. We give such a dynamic program in Appendix 3.12.5. Thus, also noting that  $G_i(v_{i0}) = 0$ , the pseudo-polynomial-time algorithm can compute  $G_i(\epsilon_i)$  for all

values of  $\epsilon_i$  that we need to consider in the second set of constraints in problem (3.14). Once we have  $G_i(\epsilon_i)$  for all values of  $\epsilon_i$ , we can solve problem (3.14) to obtain the optimal objective value of problem (3.3), which is, in turn, equal to the optimal expected revenue in problem (3.2). This approach amounts to solving a linear program with  $1+m$  decision variables and  $1+m(2+v_i^U-v_i^L)$  constraints and it provides a pseudo-polynomial-time algorithm for our assortment optimization problem.

It is also possible to build on problem (3.14) to develop an approximation method that can tradeoff computational work with performance guarantee. This approximation method is based on constructing tractable approximations to  $G_i(\epsilon_i)$ . In particular, instead of considering every single possible value of  $\epsilon_i$  with  $\epsilon_i = v_{i0}$  or  $\epsilon_i \in [v_i^L, v_i^U]$  in the second set of constraints in problem (3.14), we choose some  $\delta > 1$  and consider the values of  $\epsilon_i$  that are close to the powers of  $\delta$ . To do this, we define  $l_i^L$  as  $l_i^L = \min\{l \in \mathbb{Z} : \delta^l \geq v_i^L\}$  and  $l_i^U = \min\{l \in \mathbb{Z} : \delta^l \geq v_i^U\}$  so that we have  $[v_i^L, v_i^U] \subset [\delta^{l_i^L-1}, \delta^{l_i^U}]$ . In this case, whenever  $\epsilon_i$  lies in the interval  $[\delta^{l-1}, \delta^l]$  for some  $l = l_i^L, \dots, l_i^U$ , we approximate  $G_i(\epsilon_i)$  by

$$\hat{G}_{il} = \max_{S_i \subset N} \left\{ \sum_{j \in S_i} r_{ij} v_{ij} : \delta^{l-1} \leq v_{i0} + \sum_{j \in S_i} v_{ij} \leq \delta^l \right\}. \quad (3.15)$$

Problem (3.15) is still difficult to solve exactly as it is a knapsack problem with both upper and lower bounds, but it turns out we can compute approximate solutions to this problem. In particular, Proposition 3 in the appendix shows that we can find an assortment  $\hat{S}_{il}$  that satisfies  $\delta^{l-1} \leq v_{i0} + \sum_{j \in \hat{S}_{il}} v_{ij} \leq \delta^l$  and  $\delta \sum_{j \in \hat{S}_{il}} r_{ij} v_{ij} \geq \hat{G}_{il}$ . In other words, the assortment  $\hat{S}_{il}$  is a feasible solution to problem (3.15) and the objective value provided by this assortment deviates from the optimal objective value of problem (3.15) by no more than a factor of  $\delta$ . Proposition 3 also shows that the computational work required to obtain the assortment  $\hat{S}_{il}$  takes  $O([\delta/(\delta-1)]^2 n^{2\lceil \delta/(\delta-1) \rceil})$  time, where we use  $\lceil \cdot \rceil$  to denote the round up function.

Thus, if we want  $\hat{S}_{il}$  to be a more accurate solution to problem (3.15), then we need to choose  $\delta$  closer to one, but choosing  $\delta$  closer to one also increases the computational work to obtain the assortment  $\hat{S}_{il}$ . In this way, we can use  $\hat{S}_{il}$  as an approximate solution problem (3.15) whenever  $\epsilon_i$  lies in the interval  $[\delta^{l-1}, \delta^l]$ , while balancing computational work with accuracy.

We propose using the assortments  $\{\hat{S}_{il} : l = l_i^L, \dots, l_i^U\} \cup \{\emptyset\}$  as a collection of candidate assortments for nest  $i$ . To find the best combination of such assortments, we solve problem (3.4) after replacing the collection of candidate assortments  $\{A_{it} : t \in \mathcal{T}_i\}$  in the second set of constraints with the collection of assortments  $\{\hat{S}_{il} : l = l_i^L, \dots, l_i^U\} \cup \{\emptyset\}$ . Letting  $(\hat{x}, \hat{y})$  be the optimal solution to problem (3.4) that we obtain in this fashion and recalling that  $\bar{\gamma} = \max_{i \in M} \gamma_i$ , if we can show that  $(\delta^{2\bar{\gamma}+1} \hat{x}, \delta^{\bar{\gamma}+1} \hat{y})$  is a feasible solution to problem (3.3), then Theorem 3.4.1 implies that we can focus only on the assortments  $\{\hat{S}_{il} : l = l_i^L, \dots, l_i^U\} \cup \{\emptyset\}$  for nest  $i$  and still obtain an assortment whose expected revenue deviates from the optimal expected revenue by no more than a factor of  $\delta^{2\bar{\gamma}+1}$ . We establish this result in the following theorem. The proof of this theorem is given in the appendix.

**Theorem 3.8.2.** *Let  $(\hat{x}, \hat{y})$  be an optimal solution to problem (3.4) when we solve this problem after replacing the collection of candidate assortments  $\{A_{it} : t \in \mathcal{T}_i\}$  in the second set of constraints with the assortments  $\{\hat{S}_{il} : l = l_i^L, \dots, l_i^U\} \cup \{\emptyset\}$ . Then,  $(\delta^{2\bar{\gamma}+1} \hat{x}, \delta^{\bar{\gamma}+1} \hat{y})$  is a feasible solution to problem (3.3).*

Theorem 3.8.2 implies that if we use the assortments  $\{\hat{S}_{il} : l = l_i^L, \dots, l_i^U\} \cup \{\emptyset\}$  as candidate assortments for nest  $i$ , then we can combine these assortments for the different nests to obtain a solution to problem (3.2) whose expected revenue deviates from the optimal expected revenue by no more than a factor of  $\delta^{2\bar{\gamma}+1}$ . To find the best combination of these assortments, we need to solve problem (3.4) after replacing the collection of candidate assortments  $\{A_{it} : t \in \mathcal{T}_i\}$  in the second

set of constraints with the collection of assortments  $\{\hat{S}_{il} : l = l_i^L, \dots, l_i^U\} \cup \{\emptyset\}$ . Since we have  $\delta^{l_i^L} \geq v_i^L$  and  $\delta^{l_i^U-1} \leq v_i^U$ , we have  $l_i^U - l_i^L \leq 1 + \log_\delta(v_i^U/v_i^L)$ . Therefore, for the most general instances of the assortment optimization problem we are interested in, we can solve a linear program with  $1+m$  decision variables and at most  $1+m(2 + \log_\delta(v_i^U/v_i^L))$  constraints to find an assortment whose expected revenue deviates from the optimal expected revenue by no more than a factor of  $\delta^{2\bar{\gamma}+1}$ .

### 3.9 Upper Bounds on Optimal Expected Revenue

In the previous three sections, we give collections of candidate assortments such that if we focus on these collections, then we can stitch together an assortment with a certain performance guarantee. The performance guarantees we give reflect the worst-case performance of a given collection of assortments, where the worst-case is taken over all possible problem instances. In this section, our goal is to develop a tractable upper bound on the optimal expected revenue that we can compute for each individual problem instance. We can then use the collections of assortments given in the previous sections to obtain the best possible assortment and compare the expected revenue from this assortment with the problem instance-specific upper bound on the optimal expected revenue. In this way, we can bound the optimality gap of the assortment we obtain for a particular problem instance.

To construct an upper bound on the optimal expected revenue, we use a tighter version of problem (3.3) that is similar to the one in Section 3.6.2. In particular, since the second set of constraints in problem (3.3) can be written as  $y_i \geq \max_{S_i \subset N} V_i(S_i)^{\gamma_i} (R_i(S_i) - x)$  for all  $i \in M$ , using the decision variables

$z_i = (z_{i1}, \dots, z_{in}) \in [0, 1]^n$ , we formulate a tighter version of problem (3.3) as

$$\begin{aligned}
\min \quad & x & (3.16) \\
\text{s.t.} \quad & v_0 x \geq \sum_{i \in M} y_i \\
& y_i \geq \max_{z_i \in [0, 1]^n} \left\{ \left( v_{i0} + \sum_{j \in N} v_{ij} z_{ij} \right)^{\gamma_i} \left[ \frac{\sum_{j \in N} r_{ij} v_{ij} z_{ij}}{v_{i0} + \sum_{j \in N} v_{ij} z_{ij}} - x \right] \right\} \quad \forall i \in M.
\end{aligned}$$

Problem (3.16) is a tighter version of problem (3.3) as any feasible solution to problem (3.16) is a feasible solution to problem (3.3). Therefore, if we can solve problem (3.16) in a tractable fashion, then the optimal objective value of this problem provides an upper bound on the optimal expected revenue.

To see how we can solve problem (3.16) in a tractable fashion, for a fixed value of  $x$ , we use  $F_i(z_i | x)$  to denote the objective function of the maximization problem on the right side of the second set of constraints in problem (3.16) and let  $\hat{F}_i(x) = \max_{z_i \in [0, 1]^n} F_i(z_i | x)$ . So, problem (3.16) is equivalent to

$$\begin{aligned}
\min \quad & x & (3.17) \\
\text{s.t.} \quad & v_0 x \geq \sum_{i \in M} y_i \\
& y_i \geq \hat{F}_i(x) \quad \forall i \in M.
\end{aligned}$$

If we can show that  $\hat{F}_i(x)$  is a convex function of  $x$ , then the feasible region of problem (3.17) ends up being convex. Thus, noting that the objective function is linear, problem (3.17) becomes a convex optimization problem. Furthermore, if we can obtain subgradients of  $\hat{F}_i(x)$  with respect to  $x$  in a tractable fashion, then we can use a standard cutting plane method for convex optimization to solve problem (3.17); see [54]. The following proposition follows a standard argument in nonlinear programming to show that  $\hat{F}_i(x)$  is indeed a convex function of  $x$  and demonstrates how to obtain subgradients of this function with respect to  $x$ .



**Proposition 2.** *The function  $\hat{F}_i(\cdot)$  is convex. Furthermore, if we let  $\hat{z}_i(x)$  be an optimal solution to the problem  $\max_{z_i \in [0,1]^n} F_i(z_i | x)$ , then a subgradient of  $\hat{F}_i(\cdot)$  at  $x$  is given by  $-(v_{i0} + \sum_{j \in N} v_{ij} \hat{z}_{ij}(x))^{\gamma_i}$ .*

*Proof.* By the definitions of  $\hat{F}_i(x)$  and  $\hat{z}_i(x)$ , it follows that  $\hat{F}_i(x) = F_i(\hat{z}_i(x) | x)$  and  $\hat{F}_i(x') \geq F_i(\hat{z}_i(x) | x')$ . Subtracting the equality from the inequality and noting that  $F_i(\hat{z}_i(x) | x') - F_i(\hat{z}_i(x) | x) = -(v_{i0} + \sum_{j \in N} v_{ij} \hat{z}_{ij}(x))^{\gamma_i} (x' - x)$ , we obtain

$$\hat{F}_i(x') \geq \hat{F}_i(x) - \left( v_{i0} + \sum_{j \in N} v_{ij} \hat{z}_{ij}(x) \right)^{\gamma_i} (x' - x),$$

which implies that  $\hat{F}_i(\cdot)$  satisfies the subgradient inequality at  $x$  with a subgradient given by  $-(v_{i0} + \sum_{j \in N} v_{ij} \hat{z}_{ij}(x))^{\gamma_i}$ . In this case, by Theorem 3.2.6 in [1],  $\hat{F}_i(x)$  is a convex function of  $x$  with a subgradient as given in the proposition.  $\square$

Proposition 2 shows that we can obtain a subgradient of  $\hat{F}_i(\cdot)$  at  $x$  by solving the problem

$$\max_{z_i \in [0,1]^n} F_i(z_i | x) = \max_{z_i \in [0,1]^n} \left\{ \left( v_{i0} + \sum_{j \in N} v_{ij} z_{ij} \right)^{\gamma_i} \left[ \frac{\sum_{j \in N} r_{ij} v_{ij} z_{ij}}{v_{i0} + \sum_{j \in N} v_{ij} z_{ij}} - x \right] \right\} \quad (3.18)$$

at a fixed value of  $x$ . Thus, if we can solve the problem above in a tractable fashion, then we can also solve problem (3.17) through a standard cutting plane method. In the rest of this section, we focus on solving problem (3.18) at a fixed value of  $x$ .

Following the same argument in the proof of Lemma 3.6.2, we can show that there exists an optimal solution  $z_i^*$  to problem (3.18) that satisfies  $z_{i1}^* = 1$ ,  $z_{i2}^* = 1, \dots, z_{i,k-1}^* = 1$ ,  $z_{ik}^* \in [0, 1]$ ,  $z_{i,k+1}^* = 0, \dots, z_{in}^* = 0$  for some  $k = 1, \dots, n$ . In other words, if we define the vector  $\eta^k = (\eta_1^k, \dots, \eta_n^k) \in \mathfrak{R}^n$  such that  $\eta_1^k = 1, \dots, \eta_{k-1}^k = 1, \eta_k^k = 0, \dots, \eta_n^k = 0$  and use  $e^k$  to denote the  $k$ -th unit vector in  $\mathfrak{R}^n$ , then an optimal solution to problem (3.18) is of the form  $\eta^k + \rho e^k$  for some  $\rho \in [0, 1]$  and  $k = 1, \dots, n$ . To find the best value for  $\rho$ , we can solve the problem  $\max_{\rho \in [0,1]} F_i(\eta^k + \rho e^k | x)$ , which is a scalar optimization problem. A simple check verifies that the

first derivative of  $F_i(\eta^k + \rho e^k | x)$  with respect to  $\rho$  vanishes at one point and the point at which the first derivative vanishes can be computed in closed-form fashion. Therefore, the optimal objective value of the problem  $\max_{\rho \in [0,1]} F_i(\eta^k + \rho e^k | x)$  is attained either at one of the two end points of the interval  $[0, 1]$  or at the point where the first derivative of  $F_i(\eta^k + \rho e^k | x)$  with respect to  $\rho$  vanishes. By checking the objective value of the problem  $\max_{\rho \in [0,1]} F_i(\eta^k + \rho e^k | x)$  at these three points, we can easily solve this problem. In this case, the optimal objective value of problem (3.18) can be obtained by solving the problem  $\max_{\rho \in [0,1]} F_i(\eta^k + \rho e^k | x)$  for all  $k = 1, \dots, n$  and picking the one that yields the largest optimal objective value.

### 3.10 Computational Experiments

In this section, we provide computational experiments to test the quality of the solutions that we obtain by focusing on various collections of candidate assortments. Our goal is to compare the performance of different collections of candidate assortments under different problem characteristics. We begin by describing the details of our experimental setup. Following this description, we give the findings from our computational experiments.

#### 3.10.1 Experimental Setup

The results in Sections 3.6.2 and 3.8.1 suggest that the performance guarantee provided by the collections of assortments considered in these sections can depend on how much the revenues or the preference weights of the products in the same nest differ. Similarly, the results in Sections 3.6.2 and 3.8.2 show that the performance of the collections of assortments that we develop can depend on the magnitude

of the dissimilarity parameters. In our computational experiments, we investigate how the performance of different collections of candidate assortments changes with various problem parameters.

Throughout this section, we consider test problems with possibly synergistic products and partially-captured nests, which correspond to the most general instances of our assortment problem. We build on the example in Section 3.6.1 to generate our test problems. Noting that  $\varepsilon \leq 1$  is a small positive number, product 1 in this example has a large revenue but a small preference weight. Product 2 has a small revenue but a large preference weight. Product 3 has zero revenue and a moderate preference weight. We observe that although product 1 has a large revenue, its preference weight is too small to attract customers to its nest. In contrast, product 2 has a large preference weight and it can effectively attract customers to the nest, but if a customer is attracted to the nest, then the preference weight of product 2 is so large that the customer ends up buying product 2 with high probability. Since the revenue of product 2 is small, this outcome yields a small revenue. While product 3 has a revenue of zero, its moderate preference weight can attract customers to the nest, but its preference weight is not so large that once a customer is attracted to the nest, there is a reasonable probability that he can end up buying product 1. So, product 3 is essentially a loss leader, whose purpose is to attract customers to the nest. Recall that nested-by-revenue assortments can perform poorly in this example.

We proceed as follows to generate test problems with the same flavor as above, but with reasonably large numbers of nests and products. In all of our test problems, the number of nests is  $m = 5$  and the number of products in each nest is  $n = 25$ . This results in problem sizes that correspond to applications arising in some revenue management and retail settings. We choose a positive parame-

ter  $\varepsilon \leq 1$  that characterizes the degree to which the revenues and the preference weights of the products in the same nest differ. We vary the parameter  $\varepsilon$  in our computational experiments. To come up with the revenues and the preference weights associated with the first  $n - 1$  products in nest  $i$ , we generate  $U_{ij}$  from the uniform distribution over  $[0, 4]$  for all  $j \in N \setminus \{n\}$ . Also, we generate  $X_{ij}$  and  $Y_{ij}$  respectively from the uniform distributions over  $[1, 10]$  and  $[0.2, 1.8]$  for all  $j \in N \setminus \{n\}$ . In this case, we set the revenue of product  $j$  in nest  $i$  as  $r_{ij} = \varepsilon^{U_{ij}} \times X_{ij}$ , whereas we set the preference weight associated with product  $j$  in nest  $i$  as  $v_{ij} = \varepsilon^{2-U_{ij}} \times Y_{ij}$ . The reasoning behind our choice of revenues and preference weights is that if we multiply the revenues and the preference weights of products 1 or 2 in the example in Section 3.6.1, then we end up with a quantity that is of magnitude  $\varepsilon^2$ . If we multiply the revenues and the preference weights of the products we generate, then we obtain  $\varepsilon^{U_{ij}} \times X_{ij} \times \varepsilon^{2-U_{ij}} \times Y_{ij}$ , which is of magnitude  $\varepsilon^2$  as well. If the generated value of  $U_{ij}$  turns out to be closer to zero, then we obtain a product that resembles product 1 in the example in Section 3.6.1, whereas if the generated value of  $U_{ij}$  turns out to be closer to four, then we obtain a product that resembles product 2. The role of  $X_{ij}$  and  $Y_{ij}$  is to introduce some noise in the revenues and the preference weights. To generate the revenue and the preference weight associated with the last product  $n$  in nest  $i$ , we generate  $Y_{in}$  from the uniform distribution over  $[0.2, 1.8]$  and set the revenue and the preference weight of this product respectively as  $r_{in} = 0$  and  $v_{in} = \varepsilon^{-1} \times Y_{in}$ . We observe that the revenue and the preference weight of the last product are of the same magnitude as the revenue and the preference weight of product 3 in the example in Section 3.6.1. With this setup, it is possible to check that as  $\varepsilon$  gets smaller, there are larger differences between the revenues or the preference weights of the products in the same nest.

To come up with the dissimilarity parameter of nest  $i$ , we generate  $\gamma_i$  from the uniform distribution over  $[\gamma^L, \gamma^U]$  for all  $i \in M$ , where  $\gamma^L$  and  $\gamma^U$  are parameters we vary. We set the preference weights of the no purchase options to  $v_0 = 10$  and  $v_{i0} = \varepsilon^{-4}$  for all  $i \in M$ . In our computational experiments, we vary  $\varepsilon$  over  $\{0.6, 0.5, 0.4, 0.3\}$ , whereas we vary  $[\gamma^L, \gamma^U]$  over  $\{[0.5, 1.5], [1.0, 2.0], [1.5, 2.5], [2.0, 3.0]\}$ . In this way, we obtain 16 combinations of problem parameters. Following the approach described in the paragraph above, we generate 5,000 individual problem instances for each combination of problem parameters  $\varepsilon$  and  $[\gamma^L, \gamma^U]$ .

We test the performance of three different collections of assortments on each problem instance. The first collection of assortments is *nested-by-revenue assortments*. This collection for nest  $i$  corresponds to  $\{N_{ij} : j \in N_+\}$ . While nested-by-revenue assortments provide the performance guarantee in (3.6) for fully-captured nests, we are not able to give a performance guarantee for these assortments for the most general instances of our assortment problem. However, since they are intuitively appealing and easy to implement, it is useful to test their performance. We refer to the second collection of assortments as *nested-by-preference-and-revenue assortments*. This collection of assortments for nest  $i$  is given by  $\{N_{ij}^k : k \in N, j = 0, \dots, k\} \cup \{\{j\} : j \in N\}$ . In Section 3.8.1, we show that this collection provides the performance guarantee given in (3.12) for the most general instances of our assortment problem. The third collection that we consider is referred to as *powers-of- $\delta$  assortments* and this collection for nest  $i$  is given by  $\{\hat{S}_{il} : l = l_i^L, \dots, l_i^U\} \cup \{\emptyset\}$ . We work with powers-of- $\delta$  assortments in Section 3.8.2 and show that these assortments provide a performance guarantee of  $\max_{i \in M} \delta^{2\gamma_i+1}$ . The computational work required to obtain the powers-of- $\delta$  assortments depends on the value of  $\delta$  and we set  $\delta = 1.25$  in all of our computational experiments. With this value of  $\delta$ , we can generate all powers-of- $\delta$  assortments for a particular problem

instance and find the best assortment within this class in one tenths of a second. We note that there are  $1 + n$  nested-by-revenue assortments and  $1 + n + n^2$  nested-by-preference-and-revenue assortments in each nest. So, both of these collections grow polynomially with  $n$ . In contrast, there are  $2 + \log_\delta(v_i^U/v_i^L)$  powers-of- $\delta$  assortments in nest  $i$  and noting that  $v_i^U = v_{i0} + \sum_{j \in N} v_{ij}$  and  $v_i^L = v_{i0} + \min_{j \in N} v_{ij}$ , the number of powers-of- $\delta$  assortments grows logarithmically with  $n$ .

### 3.10.2 Computational Results

We give our main computational results in Tables 3.2, 3.3 and 3.4, where Tables 3.2, 3.3 and 3.4 respectively show the performance of the nested-by-revenue, nested-by-preference-and-revenue and powers-of- $\delta$  assortments. In these tables, the first column shows the combination of problem parameters by using the tuple  $([\gamma^L, \gamma^U], \varepsilon)$ . Recall that we generate 5,000 individual problem instances for each combination of problem parameters. For each problem instance we generate, we solve problem (3.16) to obtain an upper bound on the optimal expected revenue. We use  $\mathbf{UB}^k$  to denote the upper bound on the optimal expected revenue we obtain for problem instance  $k$ . Furthermore, given a collection of candidate assortments, we find the best assortment that we can stitch together by focusing on these candidate assortments. In Table 3.2, the candidate assortments are the nested-by-revenue assortments, whereas in Tables 3.3 and 3.4, the candidate assortments are respectively the nested-by-preference-and-revenue and powers-of- $\delta$  assortments. For problem instance  $k$ , we use  $\mathbf{Best}^k$  to denote the expected revenue provided by the best assortment we can find by focusing on a particular collection of candidate assortments. The second column in Tables 3.2, 3.3 and 3.4 shows the number of problem instances  $k$  for which we have  $\mathbf{UB}^k > \mathbf{Best}^k$ . These problem instances correspond to those where we are not able to establish the optimality of the best assortment we

can find. The third column focuses on the problem instances for which we cannot establish the optimality of the best assortment we can find, and reports the average percent gap between  $\text{UB}^k$  and  $\text{Best}^k$  over these problem instances. In other words, using  $\mathcal{K}$  to denote the set of problem instances  $\{k = 1, \dots, 5,000 : \text{UB}^k > \text{Best}^k\}$ , the third column gives

$$\frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} 100 \frac{\text{UB}^k - \text{Best}^k}{\text{UB}^k}.$$

The third column can be interpreted as the estimate of the average optimality gap of the best assortment we can find given that we cannot verify the optimality of this assortment. This column gives only an estimate of the average optimality gap since we do not know the optimal expected revenue for our problem instances and we only have an upper bound on the optimal expected revenue. The fourth column in Tables 3.2, 3.3 and 3.4 gives the 99-th percentile of the gaps between  $\text{UB}^k$  and  $\text{Best}^k$  over all 5,000 problem instances. That is, the fourth column gives the 99-th percentile of the data  $\{100(\text{UB}^k - \text{Best}^k)/\text{UB}^k : k = 1, \dots, 5,000\}$ . The fifth column shows the largest percent gap between  $\text{UB}^k$  and  $\text{Best}^k$  over all 5,000 problem instances. The sixth column shows the average number of products per nest in the best assortment we find. Finally, the last two columns give a feel for how much the revenues and the preference weights of the products in the same nest differ. In particular, the seventh column gives the average ratio between the largest and the smallest revenues in a nest, averaged over all nests and all 5,000 problem instances. Naturally, we do not consider the loss leader product with a revenue of zero when computing the ratio between the largest and the smallest revenues in a nest. The eighth column gives the average ratio between the largest and the smallest preference weights in a nest, averaged over all nests and all 5,000 problem instances.

The results in Table 3.2 indicate that nested-by-revenue assortments can per-

form well. When the dissimilarity parameters of the nests take values over  $[0.5, 1.5]$ , we can verify that nested-by-revenue assortments are optimal in about half of the problem instances. Furthermore, the optimality gaps of these assortments never exceeds 0.81%. As  $[\gamma^L, \gamma^U]$  shifts from  $[0.5, 1.5]$  to  $[2.0, 3.0]$ , the average optimality gaps of nested-by-revenue assortments increase from 0.01% to 0.38%. The overall trend is that the performance of nested-by-revenue assortments can degrade as  $[\gamma^L, \gamma^U]$  increases and the nest dissimilarity parameters become larger. Another trend we observe from Table 3.2 is that the optimality gaps of nested-by-revenue assortments tend to increase as  $\varepsilon$  gets smaller. For the problem instances with  $[\gamma^L, \gamma^U] = [0.5, 1.5]$ , the average optimality gaps of nested-by-revenue assortments increase from 0.01% to 0.05% as  $\varepsilon$  decreases from 0.6 to 0.3, whereas for the problem instances with  $[\gamma^L, \gamma^U] = [2.0, 3.0]$ , the average optimality gaps increase from 0.09% to 0.38% as  $\varepsilon$  decreases from 0.6 to 0.3. We recall that as  $\varepsilon$  gets smaller, the difference between both the revenues and the preference weights of the products in the same nest gets larger. Thus, our results indicate that the performance nested-by-revenue assortments can degrade when we have drastic differences between the revenues and the preference weights. When we have  $[\gamma^L, \gamma^U] = [2.0, 3.0]$  and  $\varepsilon = 0.3$ , there are problem instances where the estimated optimality gap of nested-by-revenue assortments can reach 5.45%. Nevertheless, we note that these problem instances seem unlikely to come up in practice as they involve products in the same nest with revenues differing by a factor of about 234 and preference weights differing by a factor of about 221.

Our findings in Table 3.3 indicate that nested-by-preference-and-revenue assortments provide small improvements over nested-by-revenue assortments when we consider the problem instances with  $[\gamma^L, \gamma^U] = [0.5, 1.5]$ . The performance of nested-by-revenue assortments is already quite satisfactory for these problem



Prob. Param. $([\gamma^L, \gamma^U], \varepsilon)$	Sub. Cnt.	Avg. % Gap in Sub.	99-th Per. % Gap.	Larg. % Gap	Avg. Ass. Size	Avg. Rev. Rat.	Avg. Pref. Rat.
$([0.5, 1.5], 0.6)$	1,999	0.01	0.07	0.34	6.38	24.08	22.59
$([0.5, 1.5], 0.5)$	2,235	0.02	0.12	0.38	6.37	42.58	40.05
$([0.5, 1.5], 0.4)$	2,494	0.03	0.20	0.66	6.55	88.42	83.35
$([0.5, 1.5], 0.3)$	2,832	0.05	0.33	0.81	6.90	234.61	221.63
$([1.0, 2.0], 0.6)$	3,933	0.03	0.20	0.52	5.94	24.08	22.59
$([1.0, 2.0], 0.5)$	4,194	0.05	0.35	0.71	6.36	42.58	40.05
$([1.0, 2.0], 0.4)$	4,447	0.08	0.51	1.43	7.00	88.42	83.35
$([1.0, 2.0], 0.3)$	4,624	0.13	0.87	2.70	7.87	234.61	221.63
$([1.5, 2.5], 0.6)$	4,529	0.06	0.40	1.56	5.17	24.08	22.59
$([1.5, 2.5], 0.5)$	4,773	0.09	0.59	1.63	5.85	42.58	40.05
$([1.5, 2.5], 0.4)$	4,899	0.14	0.85	2.13	6.74	88.42	83.35
$([1.5, 3.0], 0.3)$	4,947	0.25	1.49	4.21	7.87	234.61	221.63
$([2.0, 3.0], 0.6)$	4,696	0.09	0.61	2.00	4.42	24.08	22.59
$([2.0, 3.0], 0.5)$	4,871	0.14	0.86	2.47	5.28	42.58	40.05
$([2.0, 3.0], 0.4)$	4,960	0.22	1.25	3.70	6.33	88.42	83.35
$([2.0, 3.0], 0.3)$	4,984	0.38	2.07	5.45	7.61	234.61	221.63

Table 3.2: Performance of nested-by-revenue assortments.

instances and it turns out to be difficult to improve over these assortments. As  $[\gamma^L, \gamma^U]$  increases, however, nested-by-preference-and-revenue assortments can provide noticeable improvements over nested-by-revenue assortments. For the problem instances with  $[\gamma^L, \gamma^U] = [2.0, 3.0]$ , nested-by-preference-and-revenue assortments can decrease the largest optimality gaps of nested-by-revenue assortments by more than 2%. Finally, Table 3.4 shows that the performance of powers-of- $\delta$  assortments is not as good as the performance of the other two collections of assortments we consider. For almost all of the problem instances, we cannot verify the optimality of the best powers-of- $\delta$  assortment. Nevertheless, the largest optimality gaps of powers-of- $\delta$  assortments are still below 5%. Also, for the problem instances with  $[\gamma^L, \gamma^U] = [2.0, 3.0]$  and  $\varepsilon = 0.3$ , comparing the largest optimality gaps of nested-by-revenue and powers-of- $\delta$  assortments indicates that there are problem instances where powers-of- $\delta$  assortments improve over nested-by-revenue assortments.

Our results suggest two problem parameters that can affect the performance

Prob. Param. $([\gamma^L, \gamma^U], \varepsilon)$	Sub. Cnt.	Avg. % Gap in Sub.	99-th Per. % Gap.	Larg. % Gap	Avg. Ass. Size	Avg. Rev. Rat.	Avg. Pref. Rat.
$([0.5, 1.5], 0.6)$	1,999	0.01	0.06	0.20	6.38	24.08	22.59
$([0.5, 1.5], 0.5)$	2,235	0.02	0.11	0.31	6.37	42.58	40.05
$([0.5, 1.5], 0.4)$	2,494	0.03	0.18	0.48	6.55	88.42	83.35
$([0.5, 1.5], 0.3)$	2,832	0.04	0.28	0.78	6.90	234.61	221.63
$([1.0, 2.0], 0.6)$	3,933	0.03	0.18	0.51	5.94	24.08	22.59
$([1.0, 2.0], 0.5)$	4,194	0.04	0.27	0.71	6.37	42.58	40.05
$([1.0, 2.0], 0.4)$	4,447	0.07	0.39	0.92	7.02	88.42	83.35
$([1.0, 2.0], 0.3)$	4,624	0.11	0.65	1.31	7.89	234.61	221.63
$([1.5, 2.5], 0.6)$	4,529	0.05	0.34	0.79	5.18	24.08	22.59
$([1.5, 2.5], 0.5)$	4,773	0.08	0.47	1.03	5.87	42.58	40.05
$([1.5, 2.5], 0.4)$	4,899	0.12	0.68	2.13	6.78	88.42	83.35
$([1.5, 3.0], 0.3)$	4,947	0.19	1.03	2.42	7.92	234.61	221.63
$([2.0, 3.0], 0.6)$	4,696	0.08	0.51	1.24	4.43	24.08	22.59
$([2.0, 3.0], 0.5)$	4,871	0.12	0.68	1.82	5.31	42.58	40.05
$([2.0, 3.0], 0.4)$	4,960	0.18	0.88	2.20	6.38	88.42	83.35
$([2.0, 3.0], 0.3)$	4,984	0.29	1.33	3.26	7.68	234.61	221.63

Table 3.3: Performance of nested-by-preference-and-revenue assortments.

Prob. Param. $([\gamma^L, \gamma^U], \varepsilon)$	Sub. Cnt.	Avg. % Gap in Sub.	99-th Per. % Gap.	Larg. % Gap	Avg. Ass. Size	Avg. Rev. Rat.	Avg. Pref. Rat.
$([0.5, 1.5], 0.6)$	4,996	0.93	2.22	3.20	6.46	24.08	22.59
$([0.5, 1.5], 0.5)$	4,991	0.89	2.13	3.09	6.43	42.58	40.05
$([0.5, 1.5], 0.4)$	4,969	0.80	2.06	2.90	6.61	88.42	83.35
$([0.5, 1.5], 0.3)$	4,937	0.68	1.82	2.86	6.96	234.61	221.63
$([1.0, 2.0], 0.6)$	4,994	1.01	2.42	3.68	6.01	24.08	22.59
$([1.0, 2.0], 0.5)$	4,998	0.97	2.33	3.72	6.42	42.58	40.05
$([1.0, 2.0], 0.4)$	4,998	0.88	2.27	3.95	7.08	88.42	83.35
$([1.0, 2.0], 0.3)$	4,997	0.78	2.05	3.14	7.91	234.61	221.63
$([1.5, 2.5], 0.6)$	4,999	1.12	2.87	4.40	5.25	24.08	22.59
$([1.5, 2.5], 0.5)$	4,999	1.07	2.67	3.90	5.92	42.58	40.05
$([1.5, 2.5], 0.4)$	5,000	0.99	2.53	3.80	6.81	88.42	83.35
$([1.5, 3.0], 0.3)$	4,999	0.91	2.37	3.26	7.91	234.61	221.63
$([2.0, 3.0], 0.6)$	4,997	1.22	3.42	4.37	4.53	24.08	22.59
$([2.0, 3.0], 0.5)$	5,000	1.18	3.08	4.64	5.37	42.58	40.05
$([2.0, 3.0], 0.4)$	5,000	1.10	2.84	4.92	6.40	88.42	83.35
$([2.0, 3.0], 0.3)$	4,999	1.03	2.73	4.02	7.66	234.61	221.63

Table 3.4: Performance of powers-of- $\delta$  assortments.

of the collections of assortments we propose. The first parameter is the dissimilarity parameters of the nests. The second parameter is the degree to which the revenues or the preference weights of the products in the same nest differ. To make such trends clear, Figure 3.1 plots the optimality gaps for nested-by-preference-and-revenue assortments as a function of the nest dissimilarity parameters and the difference between the revenues and the preference weights of the products in the same nest. Our earlier results indicate that nested-by-preference-and-revenue assortments generally provide better performance than the other two collections and to conserve space, we focus only on these assortments in Figure 3.1. The horizontal axis in this figure shows the values of  $[\gamma^L, \gamma^U]$  and  $\varepsilon$  for the 16 combinations of problem parameters. The thin data series plot the average optimality gap of nested-by-preference-and-revenue assortments, whereas the thick data series plot the 99-th percentile of the optimality gaps over the 5,000 problems instances generated for a certain combination of problem parameters. From the figure, we observe that as  $\varepsilon$  shifts from 0.6 to 0.3, keeping  $[\gamma^L, \gamma^U]$  constant, the optimality gaps of nested-by-preference-and-revenue assortments get larger. Similarly, as  $[\gamma^L, \gamma^U]$  shifts from  $[0.5, 1.5]$  to  $[2.0, 3.0]$ , keeping  $\varepsilon$  constant, the optimality gaps of nested-by-preference-and-revenue assortments get larger as well.

In Tables 3.2, 3.3 and 3.4, we keep the width of the interval  $[\gamma^L, \gamma^U]$  constant at one and systematically increase the value of  $\gamma^L$ . An interesting question is how our collections of assortments perform as we increase the width of the interval  $[\gamma^L, \gamma^U]$ . To answer this question, we fix  $\gamma^L$  at one and vary  $\gamma^U$  over  $\{1.5, 2.0, 2.5, 3.0\}$ . We fix  $\varepsilon$  at 0.3, which corresponds to the case with the largest optimality gaps in our earlier results. Table 3.5 shows our findings. The layout of this table is similar to that of Tables 3.2, 3.3 and 3.4. The three portions of this table focus on the performance of nested-by-revenue, nested-by-preference-

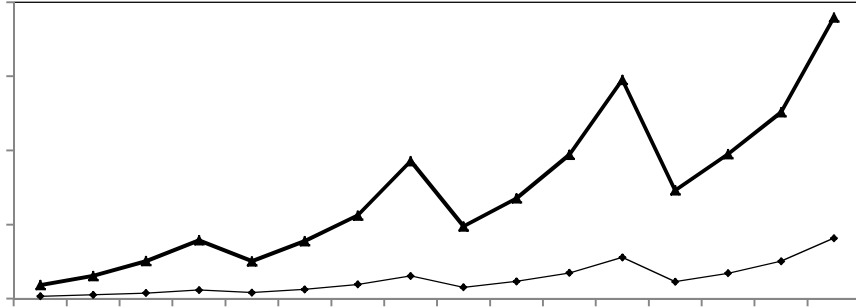


Figure 3.1: Optimality gaps for nested-by-preference-and-revenue assortments.

and-revenue and powers-of- $\delta$  assortments. The results in Table 3.5 indicates that as  $\gamma^U$  increases and the nest dissimilarity parameters tend to take values significantly larger than one, the optimality gaps of all three collections of assortments get larger. Nevertheless, even with  $\varepsilon = 0.3$ , which corresponds to a case where the revenues and the preference weights of the products in the same nest differ respectively by factors of about 234 and 221, the 99-th percentile of the optimality gaps for the best collections of assortments do not exceed 1.40% and the largest optimality gaps do not exceed 3.85%.

### 3.11 Conclusions

In this paper, we studied a class of assortment optimization problems under variants of the nested logit model. We showed that the problem is polynomially solvable when the dissimilarity parameters of the nests are less than one, and the customers

Nested-by-revenue assortments

Prob. Param. $([\gamma^L, \gamma^U], \varepsilon)$	Sub. Cnt.	Avg. % Gap in Sub.	99-th Per. % Gap.	Larg. % Gap	Avg. Ass. Size	Avg. Rev. Rat.	Avg. Pref. Rat.
$([1.0, 1.5], 0.3)$	3,866	0.05	0.37	0.88	7.81	234.61	221.63
$([1.0, 2.0], 0.3)$	4,624	0.13	0.87	2.70	7.87	234.61	221.63
$([1.0, 2.5], 0.3)$	4,827	0.22	1.53	4.62	7.74	234.61	221.63
$([1.0, 3.0], 0.3)$	4,898	0.32	2.12	6.73	7.55	234.61	221.63

Nested-by-preference-and-revenue assortments

Prob. Param. $([\gamma^L, \gamma^U], \varepsilon)$	Sub. Cnt.	Avg. % Gap in Sub.	99-th Per. % Gap.	Larg. % Gap	Avg. Ass. Size	Avg. Rev. Rat.	Avg. Pref. Rat.
$([1.0, 1.5], 0.3)$	3,866	0.05	0.31	0.76	7.82	234.61	221.63
$([1.0, 2.0], 0.3)$	4,624	0.11	0.65	1.31	7.89	234.61	221.63
$([1.0, 2.5], 0.3)$	4,827	0.18	1.00	1.93	7.78	234.61	221.63
$([1.0, 3.0], 0.3)$	4,898	0.24	1.40	3.85	7.60	234.61	221.63

Powers-of- $\delta$  assortments

Prob. Param. $([\gamma^L, \gamma^U], \varepsilon)$	Sub. Cnt.	Avg. % Gap in Sub.	99-th Per. % Gap.	Larg. % Gap	Avg. Ass. Size	Avg. Rev. Rat.	Avg. Pref. Rat.
$([1.0, 1.5], 0.3)$	4,981	0.69	1.84	2.46	7.86	234.61	221.63
$([1.0, 2.0], 0.3)$	4,997	0.78	2.05	3.14	7.91	234.61	221.63
$([1.0, 2.5], 0.3)$	4,998	0.87	2.31	3.32	7.78	234.61	221.63
$([1.0, 3.0], 0.3)$	4,999	0.97	2.69	3.96	7.60	234.61	221.63

Table 3.5: Performance of the three different collections of assortments as a function of  $\gamma^U$ .

always purchase a product within their selected nest. Relaxing either one of these assumptions renders the problem NP-hard. To deal with the NP-hard cases, we developed collections of assortments with worst-case performance guarantees. Furthermore, we formulated a tractable convex program whose optimal objective value provides an upper bound on the optimal expected revenue. In this case, we can compare the expected revenue provided by an assortment with the upper bound on the optimal expected revenue to get a feel for the optimality gap of the assortment. By following this approach, our computational experiments tested the performance of the collections of candidate assortments that we develop.

There are still interesting research questions within the context of assortment optimization under the nested logit model. In this paper, we assume that the dissimilarity parameters  $(\gamma_1, \dots, \gamma_m)$  of the nests and the preference weights  $\{v_{ij} : i \in M, j \in N\}$ ,  $\{v_{i0} : i \in M\}$  and  $v_0$  of the products and the no purchase options are constants. As we explain at the end of Section 3.3, this assumption can be justified by deriving the nested logit model through a random utility-based choice model, where the means and correlation structure of the random utilities are assumed to be fixed. Naturally, we can obtain richer choice models by allowing the means and correlation structure of the random utilities to depend on the offered assortment. For example, if a certain product is offered along with some others, then it may appear more attractive to customers, which can be modeled by allowing the mean utility of the product to depend on the whole set of offered products. Similarly, the correlation structure of the random utilities may also depend on which assortment is offered. When we allow the means or correlation structure of the random utilities to depend on the offered assortment, the preference weights or the dissimilarity parameters become dependent on the offered assortment as well.

The approach that we use in this paper has difficulties when working with

such assortment-dependent preference weights or dissimilarity parameters. For example, consider a case where the preference weights of the products in nest  $i$  or the dissimilarity parameter of nest  $i$  depends on the assortment offered within this nest. In this case, one difficulty that we run into is that the performance guarantees in Sections 3.6.2, 3.7 and 3.8.1 are obtained by using various continuous relaxations, such as those on the right side of the second set of constraints in problems (3.7) and (3.10), but it is not clear how to construct and work with such continuous relaxations when we have assortment-dependent preference weights or dissimilarity parameters. Furthermore, the proofs of the results in these sections use the monotonicity, concavity and convexity properties of  $V_i(S_i)^{\gamma_i}$  when viewed as a function of  $V_i(S_i)$ . For example, the proof of Theorem 3.7.3 uses the fact that  $V_i(S_i)^{\gamma_i}$  and  $V_i(S_i)^{1-\gamma_i}$  are both increasing in  $V_i(S_i)$  when  $\gamma_i \leq 1$ , whereas the proof of Theorem 3.8.1 uses the fact that  $V_i(S_i)^{1-\gamma_i}$  is a convex function of  $V_i(S_i)$  when  $\gamma_i \geq 1$ . Similar monotonicity, concavity and convexity properties are used in the proofs of Theorems 3.5.2 and 3.6.3 as well. These properties naturally hold when the dissimilarity parameters are constants but it is not clear how to ensure the analogous monotonicity, concavity and convexity properties when the preference weights or the dissimilarity parameters depend on the offered assortment.

Another important research direction is to work with more general forms of the nested logit model, such as the mixed or cross nested logit models. In Section 3.4, we exploit the fact that the problem  $\max_{(S_1, \dots, S_m): S_i \subset N, i \in M} \sum_{i \in M} V_i(S_i)^{\gamma_i} (R_i(S_i) - x)$  decomposes by the nests. As a result, for a fixed value of the decision variable  $x$ , the second set of constraints in problem (3.3) separates by the nests and we can construct candidate assortments by considering each nest separately. Unfortunately, this separable structure is lost under more general forms of the nested logit model. Therefore, dealing with assortment-dependent preference weights or dis-

similarity parameters and solving assortment problems under more general forms of the nested logit model remain open for further research.

## 3.12 Appendix: Omitted Proofs

### 3.12.1 Proof of Theorem 3.6.3

Noting that problem (3.3) is a relaxed version of problem (3.7), it is enough to show that  $(\alpha \hat{x}, \alpha \hat{y})$  is a feasible solution to problem (3.7). We observe that since  $(\hat{x}, \hat{y})$  is an optimal solution to problem (3.4) after replacing the collection of assortments  $\{A_{it} : t \in \mathcal{T}_i\}$  in the second set of constraints with the nested-by-revenue assortments  $\{N_{ij} : j \in N_+\}$ , this solution satisfies the second set of constraints for nest  $i$  and the nested-by-revenue assortment  $N_{i0} = \emptyset$ . Noting that  $V_i(\emptyset) = 0$ , it follows that  $\hat{y}_i \geq 0$  for all  $i \in M$ . In this case, the first constraint in problem (3.4) implies that  $\hat{x} \geq 0$ .

We fix an arbitrary nest  $i$  and let  $\hat{z}_i$  be an optimal solution to the maximization problem on the right side of the second set of constraints in problem (3.7) when this maximization problem is solved at  $x = \alpha \hat{x}$ . By Lemma 3.6.2,  $\hat{z}_i$  is of the form  $\hat{z}_{i1} = 1, \hat{z}_{i2} = 1, \dots, \hat{z}_{i,k-1} = 1, \hat{z}_{ik} \in [0, 1], \hat{z}_{i,k+1} = 0, \dots, \hat{z}_{in} = 0$  for some  $k = 1, \dots, n$ . We define  $\rho$  as  $\rho = \hat{z}_{ik} \in [0, 1]$  and consider two cases.

*Case 1.* Assume that  $k \geq 2$ . We branch into two subcases.

*Case 1.a.* Noting that  $(\hat{x}, \hat{y})$  is the optimal solution to problem (3.4) after replacing the collection of assortments  $\{A_{it} : t \in \mathcal{T}_i\}$  in the second set of constraints with the nested-by-revenue assortments  $\{N_{ij} : j \in N_+\}$ , this solution satisfies the second set of constraints in problem (3.4) for nest  $i$  and the nested-by-revenue assortment



$N_{ik} = \{1, 2, \dots, k\}$ . Thus, it holds that

$$\hat{y}_i \geq \left( \sum_{j=1}^k v_{ij} \right)^{\gamma_i} \left[ \frac{\sum_{j=1}^k r_{ij} v_{ij}}{\sum_{j=1}^k v_{ij}} - \hat{x} \right].$$

For notational convenience, let  $R_{ik'} = \sum_{j=1}^{k'} r_{ij} v_{ij}$  and  $q_{ik'} = \sum_{j=1}^{k'} v_{ij}$  for all  $k' = 1, \dots, n$ . Multiplying the inequality above by  $\frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{q_{i,k-1} + v_{ik} \rho} \frac{q_{ik}}{R_{ik}}$ , we obtain

$$\frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{q_{i,k-1} + v_{ik} \rho} \frac{q_{ik}}{R_{ik}} \hat{y}_i \geq q_{ik}^{\gamma_i} \left[ \frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{q_{i,k-1} + v_{ik} \rho} - \frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{q_{i,k-1} + v_{ik} \rho} \frac{q_{ik}}{R_{ik}} \hat{x} \right]. \quad (3.19)$$

It is simple to check that the first derivative of  $\frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{q_{i,k-1} + v_{ik} \rho}$  with respect to  $\rho$  has the same sign as  $r_{ik} q_{i,k-1} - R_{i,k-1}$  and we have  $r_{ik} q_{i,k-1} - R_{i,k-1} = \sum_{j=1}^{k-1} (r_{ik} - r_{ij}) v_{ij} \leq 0$ , where the last inequality is by the fact that  $r_{i1} \geq r_{i2} \geq \dots \geq r_{in}$ . Thus, it follows that  $\frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{q_{i,k-1} + v_{ik} \rho}$  is decreasing in  $\rho$  so that is it bounded from above by  $R_{i,k-1}/q_{i,k-1}$ . Also, noting the definitions of  $R_i(S_i)$ ,  $R_{ik'}$  and  $q_{ik'}$ , we have  $R_{ik'}/q_{ik'} = R_i(N_{ik'})$ . In this case, we can bound the expression that multiplies  $\hat{y}_i$  and  $\hat{x}$  in (3.19) as

$$\frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{q_{i,k-1} + v_{ik} \rho} \frac{q_{ik}}{R_{ik}} \leq \frac{R_{i,k-1}}{q_{i,k-1}} \frac{q_{ik}}{R_{ik}} = \frac{R_i(N_{i,k-1})}{R_i(N_{ik})}.$$

Letting  $\alpha_{ik}^1 = R_i(N_{i,k-1})/R_i(N_{ik})$  for notational brevity, using the upper bound above in (3.19) and noting that  $\hat{y}_i \geq 0$  and  $\hat{x} \geq 0$ , the inequality in (3.19) implies that

$$\alpha_{ik}^1 \hat{y}_i \geq q_k^{\gamma_i} \left[ \frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{q_{i,k-1} + v_{ik} \rho} - \alpha_{ik}^1 \hat{x} \right].$$

Finally, if we multiply the right side of the inequality above by  $(q_{k-1} + v_{ik} \rho)^{\gamma_i} / q_k^{\gamma_i} \leq 1$ , but not the left side, then the inequality is above still preserved and we have

$$\alpha_{ik}^1 \hat{y}_i \geq (q_{i,k-1} + v_{ik} \rho)^{\gamma_i} \left[ \frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{q_{i,k-1} + v_{ik} \rho} - \alpha_{ik}^1 \hat{x} \right]. \quad (3.20)$$

*Case 1.b.* Noting that  $(\hat{x}, \hat{y})$  is the optimal solution to problem (3.4) after replacing the collection of assortments  $\{A_{it} : t \in \mathcal{T}_i\}$  in the second set of constraints with

the nested-by-revenue assortments  $\{N_{ij} : j \in N_+\}$ , this solution also satisfies the second set of constraints in problem (3.4) for nest  $i$  and the nested-by-revenue assortment  $N_{i,k-1} = \{1, 2, \dots, k-1\}$ . Thus, we obtain

$$\hat{y}_i \geq \left( \sum_{j=1}^{k-1} v_{ij} \right)^{\gamma_i} \left[ \frac{\sum_{j=1}^{k-1} r_{ij} v_{ij}}{\sum_{j=1}^{k-1} v_{ij}} - \hat{x} \right] = q_{i,k-1}^{\gamma_i} \left[ \frac{R_{i,k-1}}{q_{i,k-1}} - \hat{x} \right].$$

Multiplying the inequality above by  $\frac{(q_{i,k-1} + v_{ik} \rho)^{\gamma_i}}{q_{i,k-1}^{\gamma_i}} \frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{q_{i,k-1} + v_{ik} \rho} \frac{q_{i,k-1}}{R_{i,k-1}}$  and arranging the terms, we have

$$\begin{aligned} & \frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{(q_{i,k-1} + v_{ik} \rho)^{1-\gamma_i}} \frac{q_{i,k-1}^{1-\gamma_i}}{R_{i,k-1}} \hat{y}_i \\ & \geq (q_{i,k-1} + v_{ik} \rho)^{\gamma_i} \left[ \frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{q_{i,k-1} + v_{ik} \rho} - \frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{q_{i,k-1} + v_{ik} \rho} \frac{q_{i,k-1}}{R_{i,k-1}} \hat{x} \right]. \end{aligned}$$

Since  $\hat{x} \geq 0$ , if we make the expression that multiplies  $\hat{x}$  in the inequality above even larger by multiplying it by  $\frac{(q_{i,k-1} + v_{ik} \rho)^{\gamma_i}}{q_{i,k-1}^{\gamma_i}} \geq 1$ , then the inequality above is still preserved and we have

$$\begin{aligned} & \frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{(q_{i,k-1} + v_{ik} \rho)^{1-\gamma_i}} \frac{q_{i,k-1}^{1-\gamma_i}}{R_{i,k-1}} \hat{y}_i \\ & \geq (q_{i,k-1} + v_{ik} \rho)^{\gamma_i} \left[ \frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{q_{i,k-1} + v_{ik} \rho} - \frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{(q_{i,k-1} + v_{ik} \rho)^{1-\gamma_i}} \frac{q_{i,k-1}^{1-\gamma_i}}{R_{i,k-1}} \hat{x} \right]. \quad (3.21) \end{aligned}$$

It is simple to check that whenever the first derivative of  $\frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{(q_{i,k-1} + v_{ik} \rho)^{1-\gamma_i}}$  with respect to  $\rho$  vanishes, the second derivative takes a positive value. Therefore, this expression is maximized at either  $\rho = 0$  or  $\rho = 1$ . In this case, we can bound the expression that multiplies  $\hat{y}_i$  and  $\hat{x}$  in (3.21) as

$$\frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{(q_{i,k-1} + v_{ik} \rho)^{1-\gamma_i}} \frac{q_{i,k-1}^{1-\gamma_i}}{R_{i,k-1}} \leq 1 \vee \frac{R_{ik}}{q_{ik}^{1-\gamma_i}} \frac{q_{i,k-1}^{1-\gamma_i}}{R_{i,k-1}} = 1 \vee \frac{R_i(N_{ik})}{R_i(N_{i,k-1})} \frac{V_i(N_{ik})^{\gamma_i}}{V_i(N_{i,k-1})^{\gamma_i}},$$

where we use  $a \vee b = \max\{a, b\}$ . The two terms in the maximum operator on the right side of the first inequality are obtained by evaluating the expression on the left side of the inequality at  $\rho = 0$  and  $\rho = 1$ . The equality above follows by noting

that  $R_{ik'}/q_{ik'} = R_i(N_{ik'})$  and  $q_{ik'} = V_i(N_{ik'})$ . Letting  $\alpha_{ik}^2 = \frac{R_i(N_{ik})}{R_i(N_{i,k-1})} \frac{V_i(N_{ik})^{\gamma_i}}{V_i(N_{i,k-1})^{\gamma_i}}$  for notational convenience, using the upper bound above in (3.21) and noting that  $\hat{y}_i \geq$  and  $\hat{x} \geq 0$ , the inequality in (3.21) implies that

$$(1 \vee \alpha_{ik}^2) \hat{y}_i \geq (q_{i,k-1} + v_{ik} \rho)^{\gamma_i} \left[ \frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{q_{i,k-1} + v_{ik} \rho} - (1 \vee \alpha_{ik}^2) \hat{x} \right]. \quad (3.22)$$

Putting Cases 1.a and 1.b together, we observe that  $\hat{y}_i$  and  $\hat{x}$  satisfy both (3.20) and (3.22), in which case, they must also satisfy

$$[\alpha_{ik}^1 \wedge (1 \vee \alpha_{ik}^2)] \hat{y}_i \geq (q_{i,k-1} + v_{ik} \rho)^{\gamma_i} \left[ \frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{q_{i,k-1} + v_{ik} \rho} - [\alpha_{ik}^1 \wedge (1 \vee \alpha_{ik}^2)] \hat{x} \right]. \quad (3.23)$$

Lemma 3.12.1 below shows that  $\alpha \geq 1$  for the value of  $\alpha$  given in (3.6). The proof of that lemma also shows that  $\alpha_{ik}^1 = R_i(N_{ik})/R_i(N_{i,k-1}) \geq 1$ . By the definitions of  $\alpha$ ,  $\alpha_{ik}^1$  and  $\alpha_{ik}^2$ , we obtain  $\alpha \geq \alpha_{ik}^1 \wedge \alpha_{ik}^2$  as well. In this case, we have

$$\alpha_{ik}^1 \wedge (1 \vee \alpha_{ik}^2) = (\alpha_{ik}^1 \wedge 1) \vee (\alpha_{ik}^1 \wedge \alpha_{ik}^2) = 1 \vee (\alpha_{ik}^1 \wedge \alpha_{ik}^2) \leq \alpha.$$

Thus, replacing the expression that multiplies  $\hat{y}_i$  and  $\hat{x}$  in (3.23) with an even larger expression  $\alpha$ , the inequality is still preserved and we obtain

$$\alpha \hat{y}_i \geq (q_{i,k-1} + v_{ik} \rho)^{\gamma_i} \left[ \frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{q_{i,k-1} + v_{ik} \rho} - \alpha \hat{x} \right]. \quad (3.24)$$

*Case 2.* Assume that  $k = 1$ . Since  $(\hat{x}, \hat{y})$  is the optimal solution to problem (3.4) after replacing the collection of assortments  $\{A_{it} : t \in \mathcal{T}_i\}$  in the second set of constraints with the nested-by-revenue assortments  $\{N_{ij} : j \in N_+\}$ , this solution satisfies the second set of constraints in problem (3.4) for nest  $i$  and the nested-by-revenue assortment  $N_{i1} = \{1\}$ . Therefore, we have  $\hat{y}_i \geq v_{i1}^{\gamma_i} \left[ \frac{r_{i1} v_{i1}}{v_{i1}} - \hat{x} \right]$ . Since  $\alpha \geq 1$ ,  $\hat{y}_i \geq 0$  and  $\hat{x} \geq 0$ , this inequality yields  $\alpha \hat{y}_i \geq v_{i1}^{\gamma_i} \left[ \frac{r_{i1} v_{i1}}{v_{i1}} - \alpha \hat{x} \right]$ . If we multiply the right side of the last inequality by  $\rho^{\gamma_i} \leq 1$ , but not the left side, then the inequality is still preserved and we obtain

$$\alpha \hat{y}_i \geq (v_{i1} \rho)^{\gamma_i} \left[ \frac{r_{i1} v_{i1} \rho}{v_{i1} \rho} - \alpha \hat{x} \right]. \quad (3.25)$$

Putting Cases 1 and 2 together, we succinctly write the inequalities in (3.24) and (3.25) as

$$\begin{aligned} \alpha \hat{y}_i &\geq \left( \sum_{j=1}^{k-1} v_{ij} \hat{z}_{ij} + v_{ik} \hat{z}_{ik} \right)^{\gamma_i} \left[ \frac{\sum_{j=1}^{k-1} r_{ij} v_{ij} \hat{z}_{ij} + r_{ik} v_{ik} \hat{z}_{ik}}{\sum_{j=1}^{k-1} v_{ij} \hat{z}_{ij} + v_{ik} \hat{z}_{ik}} - \alpha \hat{x} \right] \\ &= \max_{z_i \in [0,1]^n} \left\{ \left( \sum_{j \in N} v_{ij} z_{ij} \right)^{\gamma_i} \left[ \frac{\sum_{j \in N} r_{ij} v_{ij} z_{ij}}{\sum_{j \in N} v_{ij} z_{ij}} - \alpha \hat{x} \right] \right\}, \end{aligned}$$

where the equality follows from the definition of  $\hat{z}_i$ . Since the choice of nest  $i$  is arbitrary, the inequality above holds for all  $i \in M$ , which implies that the solution  $(\alpha \hat{x}, \alpha \hat{y})$  satisfies the second set of constraints in problem (3.7). Since  $(\hat{x}, \hat{y})$  is an optimal solution to problem (3.4), we have  $v_0 \hat{x} \geq \sum_{i \in M} \hat{y}_i$ , which implies that  $v_0 \alpha \hat{x} \geq \sum_{i \in M} \alpha \hat{y}_i$ . Therefore, the solution  $(\alpha \hat{x}, \alpha \hat{y})$  satisfies the first constraint in problem (3.7) as well and we obtain the desired result.  $\square$

**Lemma 3.12.1.** *Using  $\alpha$  to denote the expression in (3.6), if we have  $\gamma_i > 1$  for some  $i \in M$ , then  $\alpha \geq 1$ .*

*Proof.* Noting that  $R_i(N_{ij}) = \sum_{k=1}^j r_{ik} v_{ik} / \sum_{k=1}^j v_{ik}$ ,  $R_{ij}(N_{ij})$  is the weighted average of the revenues of the first  $j$  products in nest  $i$ . Since  $r_{i1} \geq r_{i2} \geq \dots \geq r_{in}$ , it follows that  $R_i(N_{i,j-1}) \geq R_i(N_{ij})$ . On the other hand, since  $\gamma_i > 1$ , we have

$$\begin{aligned} R_i(N_{ij}) V_i(N_{ij})^{\gamma_i} &= \frac{\sum_{k=1}^j r_{ik} v_{ik}}{\sum_{k=1}^j v_{ik}} \left( \sum_{k=1}^j v_{ik} \right)^{\gamma_i} \\ &\geq \frac{\sum_{k=1}^{j-1} r_{ik} v_{ik}}{\sum_{k=1}^{j-1} v_{ik}} \left( \sum_{k=1}^{j-1} v_{ik} \right)^{\gamma_i} \\ &= R_i(N_{i,j-1}) V_i(N_{i,j-1})^{\gamma_i}. \end{aligned}$$

Therefore, both terms of the minimum operator in the expression in (3.6) for nest  $i$  are at least one, which implies that  $\alpha$  is also at least one.  $\square$

### 3.12.2 Proof of Theorem 3.7.1

Assume that we are given any instance of the partition problem with sizes  $(c_1, \dots, c_n)$  and  $\sum_{j=1}^n c_j = 2T$ . We define an instance of the assortment feasibility problem as follows. There are two nests. The preference weight for the option of not choosing any of the nests is  $v_0 = 0$ . The dissimilarity parameters of the two nests are  $\gamma_1 = \gamma_2 = 1/2$ . For the first nest, the preference weight of the no purchase option is  $v_{10} = 2$ . This nest has only one product in it. The revenue and the preference weight associated with this product are  $r_{11} = 2(T+1)(T+3)$  and  $v_{11} = 2(2T+1)$ . For the second nest, the preference weight of the no purchase option is  $v_{20} = 1$ . The second nest has  $n$  products in it. The revenues of the products in the second nest are identical and they are given by  $r_{2j} = (T+1)(2T+1)$  for all  $j = 1, \dots, n$ . The preference weights of the products in the second nest are given by  $v_{2j} = c_j$  for all  $j = 1, \dots, n$ . We set the expected revenue threshold in the assortment feasibility problem as  $K = (T+2)(2T+1)$ .

The first observation that if we only offer the product in the first nest, then the expected revenue we generate from the first nest is

$$R_1(\{1\}) = \frac{r_{11} v_{11}}{v_{01} + v_{11}} = \frac{2(T+1)(T+3)2(2T+1)}{2 + 2(2T+1)} = (T+3)(2T+1),$$

which is larger than the revenues of the products in the second nest. Thus, if we want to get the largest possible expected revenue, then it is always optimal to offer the product in the first nest. Therefore, the only question for the assortment feasibility problem is to choose a subset  $S$  among the products in the second nest that makes sure that we obtain an expected revenue of  $K = (T+2)(2T+1)$  or more. If we offer a subset  $S$  of the products in the second nest together with the product in the first nest, then the expected revenue is  $Q_1(\{1\}, S) R_1(\{1\}) + Q_2(\{1\}, S) R_2(S)$ ,

which evaluates to

$$\frac{\sqrt{2+2(2T+1)}}{\sqrt{2+2(2T+1)}+\sqrt{1+\sum_{j\in S}c_j}}\frac{2(T+1)(T+3)2(2T+1)}{2+2(2T+1)} + \frac{\sqrt{1+\sum_{j\in S}c_j}}{\sqrt{2+2(2T+1)}+\sqrt{1+\sum_{j\in S}c_j}}\frac{(T+1)(2T+1)\sum_{j\in S}c_j}{1+\sum_{j\in S}c_j}.$$

Thus, arranging the terms in the expression above, the assortment feasibility problem asks the question of whether there is a subset  $S$  such that

$$\frac{\frac{2(T+1)(T+3)2(2T+1)}{\sqrt{2+2(2T+1)}}+(T+1)(2T+1)\frac{\sum_{j\in S}c_j}{\sqrt{1+\sum_{j\in S}c_j}}}{\sqrt{2+2(2T+1)}+\sqrt{1+\sum_{j\in S}c_j}}\geq(T+2)(2T+1).$$

If we cancel the terms in the first fraction in the numerator on the left side and move the denominator to the right, then the inequality above is equivalent to

$$2(T+3)(2T+1)\sqrt{T+1}+(T+1)(2T+1)\frac{\sum_{j\in S}c_j}{\sqrt{1+\sum_{j\in S}c_j}}\geq 2(T+2)(2T+1)\sqrt{T+1}+(T+2)(2T+1)\sqrt{1+\sum_{j\in S}c_j}.$$

Canceling the term  $2T+1$  from both sides of the inequality above, multiplying the inequality by  $\sqrt{1+\sum_{j\in S}c_j}$  and adding and subtracting one from the term  $\sum_{j\in S}c_j$ , the inequality above can be written as

$$2(T+3)\sqrt{T+1}\sqrt{1+\sum_{j\in S}c_j}+(T+1)\left(1+\sum_{j\in S}c_j\right)-(T+1)\geq 2(T+2)\sqrt{T+1}\sqrt{1+\sum_{j\in S}c_j}+(T+2)\left(1+\sum_{j\in S}c_j\right).$$

Finally, collecting all of the terms to the right, the last inequality becomes

$$\left(1+\sum_{j\in S}c_j\right)-2\sqrt{T+1}\sqrt{1+\sum_{j\in S}c_j}+(T+1)\leq 0.$$

Since the last inequality is equivalent to  $(\sqrt{1 + \sum_{j \in S} c_j} - \sqrt{T + 1})^2 \leq 0$ , there exists an assortment with an expected revenue of  $K = (T + 2)(2T + 1)$  or more if and only if there exists a subset  $S$  with  $(\sqrt{1 + \sum_{j \in S} c_j} - \sqrt{T + 1})^2 \leq 0$ . However, the only way for the last inequality to hold is to have  $\sum_{j \in S} c_j = T$ . Therefore, finding an assortment that yields an expected revenue of  $K$  or more is equivalent to finding a subset  $S$  that satisfies  $\sum_{j \in S} c_j = T$  and the latter statement is precisely what the partition problem is interested in.  $\square$

### 3.12.3 Proof of Theorem 3.7.3

Since problem (3.10) is equivalent to problem (3.3), it is enough to show that  $(2\hat{x}, 2\hat{y})$  is a feasible solution to problem (3.10). First, note that  $\hat{x} \geq 0$ . To see this claim, if  $\hat{x} < 0$ , then the right sides of the second set of constraints in problem (3.4) are strictly positive for nonempty assortments so that  $\hat{y}_i > 0$  for all  $i \in M$ . In this case,  $(\hat{x}, \hat{y})$  cannot satisfy the first constraint in problem (3.4), establishing the claim.

We fix an arbitrary nest  $i$  and let  $\hat{\epsilon}_i$  be the optimal solution to the maximization problem on the right side of the second set of constraints in problem (3.10) when this maximization problem is solved at  $x = 2\hat{x}$ . Finally, let  $\hat{z}_i(\hat{\epsilon}_i)$  be the optimal solution to problem (3.11) when this continuous knapsack problem is solved at  $\epsilon_i = \hat{\epsilon}_i$ . We consider two cases.

*Case 1.* Assume that the solution  $\hat{z}_i(\hat{\epsilon}_i)$  has exactly one fractional component. We denote this fractional component by  $k \in N$ . Since  $(\hat{x}, \hat{y})$  is the optimal solution to problem (3.4) after replacing the collection assortments  $\{A_{it} : t \in \mathcal{T}_i\}$  in the second set of constraints with  $\{\hat{S}_i(\epsilon_i) : \epsilon_i \in [0, \infty]\} \cup \{\{j\} : j \in N\}$ , the solution  $(\hat{x}, \hat{y})$  satisfies the second set of constraints in problem (3.4) for nest  $i$  and the assortment

$\hat{S}_i(\hat{\epsilon}_i)$  and we obtain

$$\begin{aligned} \hat{y}_i &\geq V_i(\hat{S}_i(\hat{\epsilon}_i))^{\gamma_i} (R_i(\hat{S}_i(\hat{\epsilon}_i)) - \hat{x}) \\ &= \frac{\sum_{j \in \hat{S}_i(\hat{\epsilon}_i)} r_{ij} v_{ij}}{(v_{i0} + \sum_{j \in \hat{S}_i(\hat{\epsilon}_i)} v_{ij})^{1-\gamma_i}} - \left( v_{i0} + \sum_{j \in \hat{S}_i(\hat{\epsilon}_i)} v_{ij} \right)^{\gamma_i} \hat{x} \geq \frac{\sum_{j \in \hat{S}_i(\hat{\epsilon}_i)} r_{ij} v_{ij}}{(v_{i0} + \hat{\epsilon}_i)^{1-\gamma_i}} - (v_{i0} + \hat{\epsilon}_i)^{\gamma_i} \hat{x}, \end{aligned} \quad (3.26)$$

where the second inequality above follows by  $\gamma_i \leq 1$  and noting that we have  $\sum_{j \in \hat{S}_i(\hat{\epsilon}_i)} v_{ij} \leq \sum_{j \in N} v_{ij} \hat{z}_{ij}(\hat{\epsilon}_i) \leq \hat{\epsilon}_i$  by the definitions of  $\hat{S}_i(\hat{\epsilon}_i)$  and  $\hat{z}_i(\hat{\epsilon}_i)$ . Similarly, the solution  $(\hat{x}, \hat{y})$  satisfies the second set of constraints in problem (3.4) for nest  $i$  and the singleton assortment  $\{k\}$  so that

$$\begin{aligned} \hat{y}_i &\geq V_i(\{k\})^{\gamma_i} (R_i(\{k\}) - \hat{x}) \\ &= \frac{r_{ik} v_{ik}}{(v_{i0} + v_{ik})^{1-\gamma_i}} - (v_{i0} + v_{ik})^{\gamma_i} \hat{x} \geq \frac{r_{ik} v_{ik}}{(v_{i0} + \hat{\epsilon}_i)^{1-\gamma_i}} - (v_{i0} + \hat{\epsilon}_i)^{\gamma_i} \hat{x}, \end{aligned} \quad (3.27)$$

where the second inequality above follows from the fact that we must have  $v_{ik} \leq \hat{\epsilon}_i$  for  $\hat{z}_{ik}(\hat{\epsilon}_i)$  to take a fractional value. Since  $\hat{S}_i(\hat{\epsilon}_i)$  includes all strictly positive and integer-valued components of  $\hat{z}_i(\hat{\epsilon}_i)$  and  $k$  is the only component of  $\hat{z}_i(\hat{\epsilon}_i)$  that takes a fractional value, we have  $\sum_{j \in \hat{S}_i(\hat{\epsilon}_i)} r_{ij} v_{ij} + r_{ik} v_{ik} \geq \sum_{j \in N} r_{ij} v_{ij} \hat{z}_{ij}(\hat{\epsilon}_i) = \hat{K}_i(\hat{\epsilon}_i)$ , where the equality follows by the definition of  $\hat{z}_i(\hat{\epsilon}_i)$ . Using this relationship and adding (3.26) and (3.27), we have

$$\begin{aligned} 2\hat{y}_i &\geq \frac{\sum_{j \in \hat{S}_i(\hat{\epsilon}_i)} r_{ij} v_{ij} + r_{ik} v_{ik}}{(v_{i0} + \hat{\epsilon}_i)^{1-\gamma_i}} - (v_{i0} + \hat{\epsilon}_i)^{\gamma_i} 2\hat{x} \geq \frac{\hat{K}_i(\hat{\epsilon}_i)}{(v_{i0} + \hat{\epsilon}_i)^{1-\gamma_i}} - (v_{i0} + \hat{\epsilon}_i)^{\gamma_i} 2\hat{x} \\ &\geq \frac{K_i(\hat{\epsilon}_i)}{(v_{i0} + \hat{\epsilon}_i)^{1-\gamma_i}} - (v_{i0} + \hat{\epsilon}_i)^{\gamma_i} 2\hat{x} = \max_{\epsilon_i \geq 0} \left\{ (v_{i0} + \epsilon_i)^{\gamma_i} \left[ \frac{K_i(\epsilon_i)}{v_{i0} + \epsilon_i} - 2\hat{x} \right] \right\}, \end{aligned} \quad (3.28)$$

where the last inequality follows from the fact that problem (3.11) is a relaxation of problem (3.9) and the last equality follows from the definition of  $\hat{\epsilon}_i$ .

*Case 2.* Assume that the solution  $\hat{z}_i(\hat{\epsilon}_i)$  does not have any fractional components. In this case,  $\hat{S}_i(\hat{\epsilon}_i)$  includes all strictly positive components of  $\hat{z}_i(\hat{\epsilon}_i)$  and we obtain  $\sum_{j \in \hat{S}_i(\hat{\epsilon}_i)} r_{ij} v_{ij} = \sum_{j \in N} r_{ij} v_{ij} \hat{z}_{ij}(\hat{\epsilon}_i) = \hat{K}_i(\hat{\epsilon}_i)$ . Using this relationship and



following the same argument that we used to obtain (3.26) in the first case, we have

$$\hat{y}_i \geq \frac{\sum_{j \in \hat{S}_i(\hat{\epsilon}_i)} r_{ij} v_{ij}}{(v_{i0} + \hat{\epsilon}_i)^{1-\gamma_i}} - (v_{i0} + \hat{\epsilon}_i)^{\gamma_i} \hat{x} \geq \frac{K_i(\hat{\epsilon}_i)}{(v_{i0} + \hat{\epsilon}_i)^{1-\gamma_i}} - (v_{i0} + \hat{\epsilon}_i)^{\gamma_i} \hat{x}.$$

Multiplying the inequality above by two, we obtain

$$\begin{aligned} 2\hat{y}_i &\geq 2 \frac{K_i(\hat{\epsilon}_i)}{(v_{i0} + \hat{\epsilon}_i)^{1-\gamma_i}} - (v_{i0} + \hat{\epsilon}_i)^{\gamma_i} 2\hat{x} \\ &\geq \frac{K_i(\hat{\epsilon}_i)}{(v_{i0} + \hat{\epsilon}_i)^{1-\gamma_i}} - (v_{i0} + \hat{\epsilon}_i)^{\gamma_i} 2\hat{x} = \max_{\epsilon_i \geq 0} \left\{ (v_{i0} + \epsilon_i)^{\gamma_i} \left[ \frac{K_i(\epsilon_i)}{v_{i0} + \epsilon_i} - 2\hat{x} \right] \right\}, \end{aligned} \quad (3.29)$$

where the last inequality follows from the definition of  $\hat{\epsilon}_i$ .

Collecting (3.28) and (3.29) in the two cases, the solution  $(2\hat{x}, 2\hat{y})$  satisfies the second set of constraints for nest  $i$  in problem (3.10). Noting that our choice of nest  $i$  is arbitrary, the second set of constraints in problem (3.10) is satisfied by the solution  $(2\hat{x}, 2\hat{y})$ . Finally, since the solution  $(\hat{x}, \hat{y})$  is optimal to problem (3.4), we have  $v_0 \hat{x} \geq \sum_{i \in M} \hat{y}_i$ , which implies that  $v_0 2\hat{x} \geq \sum_{i \in M} 2\hat{y}_i$ . Therefore, the solution  $(2\hat{x}, 2\hat{y})$  satisfies the first constraint in problem (3.10) as well and we obtain the desired result.  $\square$

### 3.12.4 Proof of Theorem 3.8.1

The proof follows from a reasoning similar to those in the proofs of Theorems 3.6.3 and 3.7.3, but there are some key new points. Following an argument similar to the one at the beginning of the proof of Theorem 3.7.3, we have  $\hat{x} \geq 0$ . We fix an arbitrary nest  $i$  and consider three cases.

*Case 1.* Assume that  $\gamma_i > 1$  and  $\hat{y}_i \geq 0$ . We let  $\hat{z}_i$  be an optimal solution to the problem

$$\max_{z_i \in [0,1]^n} \left\{ \left( v_{i0} + \sum_{j \in N} v_{ij} z_{ij} \right)^{\gamma_i} \left[ \frac{\sum_{j \in N} r_{ij} v_{ij} z_{ij}}{v_{i0} + \sum_{j \in N} v_{ij} z_{ij}} - \beta \hat{x} \right] \right\}.$$

Using the same idea in the proof of Lemma 3.6.2, we can show that  $\hat{z}_i$  is of the form  $\hat{z}_{i1} = 1, \hat{z}_{i2} = 1, \dots, \hat{z}_{i,k-1} = 1, \hat{z}_{ik} \in [0, 1], \hat{z}_{i,k+1} = 0, \dots, \hat{z}_{in} = 0$  for some  $k = 1, \dots, n$ . We define  $\rho$  as  $\rho = \hat{z}_{ik} \in [0, 1]$  and branch into two subcases.

*Case 1.a.* Assume that  $k \geq 2$ . Noting that  $(\hat{x}, \hat{y})$  is the optimal solution to problem (3.4) after replacing the collection of assortments  $\{A_{it} : t \in \mathcal{T}_i\}$  in the second set of constraints with  $\{N_{ij}^{k'} : k' \in N, j = 0, \dots, k'\} \cup \{\{j\} : j \in N\}$ , this solution satisfies the second set of constraints in problem (3.4) for nest  $i$  and the assortment  $N_{ik}^n = \{1, 2, \dots, k\}$ . Thus, it holds that

$$\hat{y}_i \geq \left( v_{i0} + \sum_{j=1}^k v_{ij} \right)^{\gamma_i} \left[ \frac{\sum_{j=1}^k r_{ij} v_{ij}}{v_{i0} + \sum_{j=1}^k v_{ij}} - \hat{x} \right].$$

For notational convenience, let  $R_{ik'} = \sum_{j=1}^{k'} r_{ij} v_{ij}$  and  $q_{ik'} = \sum_{j=1}^{k'} v_{ij}$  for all  $k' = 1, \dots, n$ . The expression in the numerator of the fraction above corresponds to  $R_{ik}$  and if we replace the expression  $R_{ik}$  with the smaller expression  $R_{i,k-1} + r_{ik} v_{ik} \rho$ , then the inequality above still holds. Also, since  $\beta \geq 1$ , we can increase  $\hat{y}_i$  on the left side to  $\beta \hat{y}_i$  and the inequality still holds. Thus, the inequality above yields  $\beta \hat{y}_i \geq (v_{i0} + q_{ik})^{\gamma_i - 1} (R_{i,k-1} + r_{ik} v_{ik} \rho - (v_{i0} + q_{ik}) \hat{x})$ . Since  $\hat{y}_i \geq 0$ , if we multiply the right side of the last inequality by  $(v_{i0} + q_{i,k-1} + v_{ik} \rho)^{\gamma_i - 1} / (v_{i0} + q_{ik})^{\gamma_i - 1} \leq 1$ , but not the left side, then the last inequality is preserved and we obtain  $\beta \hat{y}_i \geq (v_{i0} + q_{i,k-1} + v_{ik} \rho)^{\gamma_i - 1} (R_{i,k-1} + r_{ik} v_{ik} \rho - (v_{i0} + q_{ik}) \hat{x})$ . We write this inequality as

$$\begin{aligned} \beta \hat{y}_i &\geq (v_{i0} + q_{i,k-1} + v_{ik} \rho)^{\gamma_i} \left[ \frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{v_{i0} + q_{i,k-1} + v_{ik} \rho} - \frac{v_{i0} + q_{ik}}{v_{i0} + q_{i,k-1} + v_{ik} \rho} \hat{x} \right] \\ &\geq (v_{i0} + q_{i,k-1} + v_{ik} \rho)^{\gamma_i} \left[ \frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{v_{i0} + q_{i,k-1} + v_{ik} \rho} - \beta \hat{x} \right], \end{aligned} \quad (3.30)$$

where the second inequality above follows by  $(v_{i0} + q_{ik}) / (v_{i0} + q_{i,k-1} + v_{ik} \rho) \leq V_i(N_{ik}^n) / V_i(N_{i,k-1}^n) \leq \beta$ .

*Case 1.b.* Assume that  $k = 1$ . In this case, if nest  $i$  is a partially-captured nest with  $v_{i0} > 0$ , then we can use the same approach in Case 1.a to show that the

inequality in (3.30) is satisfied with  $k = 1$ . On the other hand, if nest  $i$  is a fully-captured nest with  $v_{i0} = 0$ , then we can use the same approach in Case 2 in the proof of Theorem 3.6.3 to show that the inequality in (3.30) is satisfied with  $k = 1$ . Thus, the inequality in (3.30) holds under Case 1.b as well. Putting Cases 1.a and 1.b together, we have

$$\begin{aligned} \beta \hat{y}_i &\geq (v_{i0} + q_{i,k-1} + v_{ik} \rho)^{\gamma_i} \left[ \frac{R_{i,k-1} + r_{ik} v_{ik} \rho}{v_{i0} + q_{i,k-1} + v_{ik} \rho} - \beta \hat{x} \right] \\ &= \left( v_{i0} + \sum_{j \in N} v_{ij} \hat{z}_{ij} \right)^{\gamma_i} \left[ \frac{\sum_{j \in N} r_{ij} v_{ij} \hat{z}_{ij}}{v_{i0} + \sum_{j \in N} v_{ij} \hat{z}_{ij}} - \beta \hat{x} \right] \geq V_i(S_i)^{\gamma_i} (R_i(S_i) - \beta \hat{x}) \end{aligned}$$

for all  $S_i \subset N$ , where the second inequality follows from the definition of  $\hat{z}_i$ . Therefore, the solution  $(\beta \hat{x}, \beta \hat{y})$  satisfies the second set of constraints for nest  $i$  in problem (3.3).

*Case 2.* Assume that  $\gamma_i > 1$  and  $\hat{y}_i < 0$ . We let  $\hat{z}_i$  be an optimal solution to the problem

$$\max_{z_i \in [0,1]^n} \left\{ \sum_{j \in N} r_{ij} v_{ij} z_{ij} - \beta \hat{x} \left( v_{i0} + \sum_{j \in N} v_{ij} z_{ij} \right) - \beta \hat{y}_i \left( v_{i0} + \sum_{j \in N} v_{ij} z_{ij} \right)^{1-\gamma_i} \right\}. \quad (3.31)$$

Following the idea in the proof of Lemma 3.6.2, we can show that  $\hat{z}_i$  is of the form  $\hat{z}_{i1} = 1, \hat{z}_{i2} = 1, \dots, \hat{z}_{i,k-1} = 1, \hat{z}_{ik} \in [0, 1], \hat{z}_{i,k+1} = 0, \dots, \hat{z}_{in} = 0$  for some  $k = 1, \dots, n$ . We claim that we have either  $\hat{z}_{ik} = 0$  or  $\hat{z}_{ik} = 1$ , implying that the optimal solution to the problem above corresponds to a nested-by-revenue assortment. To get a contradiction, we define  $\rho$  as  $\rho = \hat{z}_{ik}$  and assume that  $\rho \in (0, 1)$ . Since setting  $\hat{z}_{i1} = 1, \hat{z}_{i2} = 1, \dots, \hat{z}_{i,k-1} = 1, \hat{z}_{ik} = \rho, \hat{z}_{i,k+1} = 0, \dots, \hat{z}_{in} = 0$  yields an optimal solution to problem (3.31), if we fix all of the decision variables except for  $z_{ik}$  at their optimal values in the problem above and optimize only over the decision variable  $z_{ik}$ , then  $\rho \in (0, 1)$  must be an optimal solution. However, since  $\gamma_i > 1$  and  $\hat{y}_i < 0$ , the objective function of the problem above is strictly

convex and its optimal solution must occur at either  $\rho = 0$  or  $\rho = 1$ . Thus, the claim holds, implying that the optimal solution to problem (3.31) corresponds to a nested-by-revenue assortment and we denote this nested by revenue assortment by  $\{1, 2, \dots, k\}$  for some  $k \in N$ .

Noting that  $(\hat{x}, \hat{y})$  is the optimal solution to problem (3.4) after replacing the collection of assortments  $\{A_{it} : t \in \mathcal{T}_i\}$  in the second set of constraints with  $\{N_{ij}^{k'} : k' \in N, j = 0, \dots, k'\} \cup \{\{j\} : j \in N\}$ , this solution satisfies the second set of constraints in problem (3.4) for nest  $i$  and the assortment  $N_{ik}^n = \{1, 2, \dots, k\}$ . Therefore, it follows that  $\hat{y}_i \geq V_i(N_{ik}^n)^{\gamma_i} (R_i(N_{ik}^n) - \hat{x})$ . Multiplying both sides of this inequality with  $\beta$ , we obtain

$$\beta \hat{y}_i \geq \beta V_i(N_{ik}^n)^{\gamma_i} (R_i(N_{ik}^n) - \hat{x}) \geq V_i(N_{ik}^n)^{\gamma_i} (R_i(N_{ik}^n) - \beta \hat{x}),$$

where the second inequality uses the fact that  $\beta \geq 1$ . Arranging the terms in the inequality  $\beta \hat{y}_i \geq V_i(N_{ik}^n)^{\gamma_i} (R_i(N_{ik}^n) - \beta \hat{x})$  by using the definitions of  $V_i(S_i)$  and  $R_i(S_i)$ , we obtain  $\sum_{j \in N_{ik}^n} r_{ij} v_{ij} - V_i(N_{ik}^n) \beta \hat{x} - V_i(N_{ik}^n)^{1-\gamma_i} \beta \hat{y}_i \leq 0$ . Noting that the optimal solution to problem (3.31) corresponds to the nested-by-revenue assortment  $N_{ik}^n = \{1, 2, \dots, k\}$ , the last inequality shows that the optimal objective value of problem (3.31) is negative. Therefore, for all  $S_i \subset N$ , we have

$$\begin{aligned} & \sum_{j \in S_i} r_{ij} v_{ij} - V_i(S_i) \beta \hat{x} - V_i(S_i)^{1-\gamma_i} \beta \hat{y}_i \\ & \leq \max_{z_i \in [0,1]^n} \left\{ \sum_{j \in N} r_{ij} v_{ij} z_{ij} - \beta \hat{x} \left( v_{i0} + \sum_{j \in N} v_{ij} z_{ij} \right) - \beta \hat{y}_i \left( v_{i0} + \sum_{j \in N} v_{ij} z_{ij} \right)^{1-\gamma_i} \right\} \leq 0. \end{aligned}$$

Arranging the terms in the inequality  $\sum_{j \in S_i} r_{ij} v_{ij} - V_i(S_i) \beta \hat{x} - V_i(S_i)^{1-\gamma_i} \beta \hat{y}_i \leq 0$  by using the definitions of  $V_i(S_i)$  and  $R_i(S_i)$ , we obtain  $\beta \hat{y}_i \geq V_i(S_i)^{\gamma_i} (R_i(S_i) - \beta \hat{x})$  for all  $S_i \subset N$ , which shows that the solution  $(\beta \hat{x}, \beta \hat{y})$  satisfies the second set of constraints for nest  $i$  in problem (3.3).

*Case 3.* Assume that  $\gamma_i \leq 1$ . In this case, by using a line of reasoning that is similar to the one in the proof of Theorem 3.7.3, we can show that  $2 \hat{y}_i \geq$

$V_i(S_i)^{\gamma_i} (R_i(S_i) - 2\hat{x})$  for all  $S_i \subset N$ . Multiplying this inequality by  $\beta/2 \geq 1$ , we obtain  $\beta \hat{y}_i \geq V_i(S_i)^{\gamma_i} (\beta R_i(S_i)/2 - \beta \hat{x}) \geq V_i(S_i)^{\gamma_i} (R_i(S_i) - \beta \hat{x})$  for all  $S_i \subset N$ . The last inequality shows that the solution  $(\beta \hat{x}, \beta \hat{y})$  satisfies the second set of constraints for nest  $i$  in problem (3.3).

Collecting the three cases together, the solution  $(\beta \hat{x}, \beta \hat{y})$  satisfies the second set of constraints for nest  $i$  in problem (3.3). Noting that our choice of nest  $i$  is arbitrary, the second set of constraints in problem (3.3) is satisfied by the solution  $(\beta \hat{x}, \beta \hat{y})$ . Finally, since the solution  $(\hat{x}, \hat{y})$  is optimal to problem (3.4), we have  $v_0 \hat{x} \geq \sum_{i \in M} \hat{y}_i$ , which implies that  $v_0 \beta \hat{x} \geq \sum_{i \in M} \beta \hat{y}_i$ . Therefore, the solution  $(\beta \hat{x}, \beta \hat{y})$  satisfies the first constraint in problem (3.3) as well and we obtain the desired result.  $\square$

### 3.12.5 Knapsack Problems with Equality Constraints

In this section, we begin by giving a dynamic program that obtains the optimal solution to problem (3.13) in  $O(n v_i^U)$  time. Following this result, we develop a tractable method to obtain approximate solutions to problem (3.15). To obtain the optimal solution to problem (3.13) through a dynamic program, we let  $\zeta_i(k, b_i)$  be the optimal objective value of problem (3.13) when we focus only on the first  $k$  products in this problem and replace the right side of the constraint with  $b_i$ . In other words, we have

$$\zeta_i(k, b_i) = \max_{S_i \subset \{1, \dots, k\}} \left\{ \sum_{j \in S_i} r_{ij} v_{ij} : v_{i0} + \sum_{j \in S_i} v_{ij} = b_i \right\}$$

with the convention that  $\zeta_i(k, b_i) = -\infty$  when the problem on the right side above is infeasible. We note that  $\zeta_i(n, \epsilon_i)$  corresponds to the optimal objective value of problem (3.13). In this case,  $\zeta_i(k, b_i)$  satisfies the dynamic programming recursion

$$\zeta_i(k, b_i) = \max \left\{ r_{ik} v_{ik} + \zeta_i(k-1, b_i - v_{ik}), \zeta_i(k-1, b_i) \right\}$$

with the boundary condition that  $\zeta_i(0, v_{i0}) = 0$  and  $\zeta_i(0, b_i) = -\infty$  for all  $b_i \in \{0, 1, \dots, v_i^U\} \setminus \{v_{i0}\}$ . We can use the dynamic programming recursion above to compute  $\zeta_i(n, b_i)$  for all  $b_i \in \{0, 1, \dots, v_i^U\}$  in  $O(n v_i^U)$  time. In this case, the values in the set  $\{\zeta_i(n, b_i) : b_i = 0, 1, \dots, v_i^U\}$  correspond to the values  $\{G_i(\epsilon_i) : \epsilon_i = 0, 1, \dots, v_i^U\}$ , as desired. The dynamic program above is similar to the one that is used for solving the partition and knapsack problems; see [23].

In the rest of this section, we focus on obtaining approximate solutions to problem (3.15). For notational brevity, we omit the subscripts for the nest and use the decision variables  $z = (z_1, \dots, z_n)$  to consider the problem

$$\hat{G}_l = \max \left\{ \sum_{j=1}^n r_j v_j z_j : \delta^{l-1} \leq v_0 + \sum_{j=1}^n v_j z_j \leq \delta^l, z \in \{0, 1\}^n \right\}. \quad (3.32)$$

We are interested in finding a feasible solution to the problem above whose objective value deviates from the optimal objective value by no more than a factor of  $\delta$ . To that end, we begin by classifying the products in the problem above into two categories. A product  $j$  satisfying  $v_j > (\delta - 1) \delta^{l-1}$  is called a large product, whereas a product  $j$  satisfying  $v_j \leq (\delta - 1) \delta^{l-1}$  is called a small product. We use  $N^L$  and  $N^S$  to respectively denote the sets of large and small products, with  $N = N^L \cup N^S$ . We observe that an optimal solution to problem (3.32) cannot include  $\lceil \delta/(\delta - 1) \rceil$  or more large products. Otherwise, the constraint in problem (3.32) evaluates to more than  $\lceil \delta/(\delta - 1) \rceil (\delta - 1) \delta^{l-1} \geq \delta^l$ , violating its upper bound. For notational brevity, we set  $q = \lceil \delta/(\delta - 1) \rceil$  throughout this section so that an optimal solution to problem (3.32) cannot include  $q$  or more large products.

To obtain a tractable approximation to problem (3.32), we consider a special linear programming relaxation of this problem. In particular, we choose a subset  $J^L \subset N^L$  of large products and a subset  $J^S \subset N^S$  of small products and solve the

problem

$$\begin{aligned}
\max \quad & \sum_{j=1}^n r_j v_j z_j & (3.33) \\
\text{s.t.} \quad & \sum_{j=1}^n v_j z_j \leq \delta^l \\
& z_j = 1 & \forall j \in J^L \cup J^S \\
& z_j = 0 & \forall j \in N^L \setminus J^L \\
& 0 \leq z_j \leq \mathbf{1}(r_j v_j \leq \min_{k \in J^S} r_k v_k) & \forall j \in N^S \setminus J^S.
\end{aligned}$$

The problem above is a continuous knapsack problem with only an upper bound constraint and the values of some of the variables are fixed at zero or one. In particular, the decision variables corresponding to the products in  $J^L$  and  $J^S$  are set at one. The decision variables corresponding to the large products in  $N^L \setminus J^L$  are set to zero. If the objective function coefficient of a small product in  $N^S \setminus J^S$  is smaller than the smallest of the objective function coefficient of the small products in  $J^S$ , then the decision variable corresponding to this small product is allowed to take values between zero and one. Otherwise, the decision variable corresponding to this product is fixed at zero. Noting that the utility of product  $j$  in problem (3.33) is given by  $r_j v_j$  and the utility-to-space consumption ratio of product  $j$  is given by  $r_j$ , we can solve problem (3.33) by using the following procedure. We put all of the products in  $J^L \cup J^S$  into the knapsack and drop these products from consideration. Also, we drop the products in  $N^L \setminus J^L$  from consideration immediately. We order the products in  $N^S$  with respect to their utilities. If there are any products in  $N^S \setminus J^S$  whose utilities exceed the smallest of the utilities in  $J^S$ , then we drop these products from consideration as well. Considering the remaining products in  $N^S \setminus J^S$ , we order these products with respect to their utility-to-space consumption ratios and fill the knapsack starting from the products with the largest utility-to-space consumption ratios. Therefore, assuming that we

already have the orderings of the items with respect to their utilities and utility-to-space consumption ratios, we can solve problem (3.33) in  $O(n)$  time. It is also useful to observe that the optimal solution to problem (3.33) that we obtain in this fashion includes at most one fractional component. The continuous knapsack problem in (3.33) is inspired by [17], where the authors use a similar continuous knapsack problem to construct polynomial-time approximation schemes for multi-dimensional knapsack problems. We use  $\hat{z}(J^L, J^S)$  to denote the optimal solution to problem (3.33), where our notation emphasizes the fact that the solution to this problem depends on the choice of  $J^L$  and  $J^S$ .

Using the solution  $\hat{z}(J^L, J^S)$ , we define the assortment  $\hat{S}(J^L, J^S) = \{j \in N : \hat{z}_j(J^L, J^S) = 1\}$ , including to the products that take strictly positive and integer values in the solution  $\hat{z}(J^L, J^S)$ . By using the discussion above, for given  $J^L$  and  $J^S$ , we can compute  $\hat{S}(J^L, J^S)$  in  $O(n)$  time. We use  $\wp^L$  to denote the set of subsets of  $N^L$  with cardinality not exceeding  $q$ . Similarly, we use  $\wp^S$  to denote the set of subsets of  $N^S$  with cardinality not exceeding  $q$ . In this case, the next proposition shows that the collection of assortments  $\{\hat{S}(J^L, J^S) : J^L \in \wp^L, J^S \in \wp^S\}$  includes a feasible solution to problem (3.32) such that the objective value provided by this feasible solution deviates from the optimal objective value of problem (3.32) by at most a factor of  $\delta$ .

**Proposition 3.** *Assuming that problem (3.32) has a feasible solution, there exists an assortment in the collection  $\{\hat{S}(J^L, J^S) : J^L \in \wp^L, J^S \in \wp^S\}$  such that this assortment yields a feasible solution to problem (3.32) and the objective value provided by this assortment deviates from the optimal objective value of problem (3.32) by at most a factor of  $\delta$ . Furthermore, all of the assortments in the collection  $\{\hat{S}(J^L, J^S) : J^L \in \wp^L, J^S \in \wp^S\}$  can be constructed in  $O(q^2 n^{2q+1})$  time.*

*Proof.* By the definitions of  $\wp^L$  and  $\wp^S$ , we have  $|\wp^L| = |\wp^S| = O(q n^q)$ . Therefore,



there are  $O(q^2 n^{2q})$  assortments in the collection  $\{\hat{S}(J^L, J^S) : J^L \in \wp^L, J^S \in \wp^S\}$ . Noting the discussion right before the proposition, each one of these assortments can be constructed in  $O(n)$  time. Thus, all of the assortments in the collection  $\{\hat{S}(J^L, J^S) : J^L \in \wp^L, J^S \in \wp^S\}$  can be constructed in  $O(q^2 n^{2q+1})$  time.

Letting  $\tilde{z}$  be the optimal solution to problem (3.32), we define the assortment  $\tilde{S}$  corresponding to this solution as  $\tilde{S} = \{j \in N : \tilde{z}_j = 1\}$ . We let  $\tilde{J}^L$  and  $\tilde{J}^S$  to respectively be the large and small products in the assortment  $\tilde{S}$ . By the discussion that follows problem (3.32), we must have  $|\tilde{J}^L| \leq q$ , implying that  $\tilde{J}^L \in \wp^L$ . If we assume that  $|\tilde{J}^S| \leq q$ , then we have  $\tilde{J}^S \in \wp^S$  as well. Thus, the assortment  $\hat{S}(\tilde{J}^L, \tilde{J}^S)$  is included in the collection  $\{\hat{S}(J^L, J^S) : J^L \in \wp^L, J^S \in \wp^S\}$ . Furthermore, by the definitions of  $\hat{z}(J^L, J^S)$  and  $\hat{S}(J^L, J^S)$ , the assortment  $\hat{S}(\tilde{J}^L, \tilde{J}^S)$  includes all of the products in  $\tilde{J}^L$  and  $\tilde{J}^S$ , which implies that  $\hat{S}(\tilde{J}^L, \tilde{J}^S) \supset \tilde{J}^L \cup \tilde{J}^S = \tilde{S}$  so that  $\hat{S}(\tilde{J}^L, \tilde{J}^S)$  includes all of the products in  $\tilde{S}$ . Thus,  $\hat{S}(\tilde{J}^L, \tilde{J}^S)$  must provide an objective value for problem (3.32) that is at least as large as the one provided by  $\tilde{S}$  and we conclude that  $\hat{S}(\tilde{J}^L, \tilde{J}^S)$  is an optimal solution to problem (3.32). This establishes the desired result under the assumption that  $|\tilde{J}^S| \leq q$ . In the rest of the proof, we assume that  $|\tilde{J}^S| > q$ .

We let  $A^S$  be the subset of  $\tilde{J}^S$  that includes the  $q$  products in  $\tilde{J}^S$  with the largest utilities. In other words, we have  $A^S \subset \tilde{J}^S$ ,  $|A^S| = q$  and  $r_j v_j \leq \min_{k \in A^S} r_k v_k$  for all  $j \in \tilde{J}^S \setminus A^S$ . Consider the solution  $\hat{z}(\tilde{J}^L, A^S)$  that we obtain by solving problem (3.33) with  $J^L = \tilde{J}^L$  and  $J^S = A^S$ . If this solution has a fractional component  $j'$ , then by the fourth set of constraints in problem (3.33), this component must satisfy  $r_{j'} v_{j'} \leq r_k v_k$  for all  $k \in A^S$ . Also, the component  $j'$  must be in  $N^S \setminus A^S$  so that the product  $j'$  is a small product. In this case, noting that the optimal objective value of problem (3.32) is given by  $\hat{G}_l$ , we have  $\hat{G}_l = \sum_{j \in \tilde{J}^L} r_j v_j + \sum_{j \in \tilde{J}^S} r_j v_j = \sum_{j \in \tilde{J}^L} r_j v_j + \sum_{j \in A^S} r_j v_j + \sum_{j \in \tilde{J}^S \setminus A^S} r_j v_j \geq \sum_{j \in A^S} r_j v_j \geq q r_{j'} v_{j'}$ , where the first

equality is by the fact that  $\tilde{S} = \tilde{J}^L \cup \tilde{J}^S$  is an optimal solution to problem (3.32) and the second inequality is by the fact that  $r_{j'} v_{j'} \leq r_k v_k$  for all  $k \in A^S$  and  $|A^S| = q$ . The last chain of inequalities yields  $r_{j'} v_{j'} \leq \hat{G}_l/q$ .

We claim that the assortment  $\hat{S}(\tilde{J}^L, A^S)$  provides a feasible solution to problem (3.32). First, we show this claim under the assumption that the solution  $\hat{z}(\tilde{J}^L, A^S)$  consumes all of the knapsack capacity in problem (3.33) when we solve this problem with  $J^L = \tilde{J}^L$  and  $J^S = A^S$ . We use  $j'$  to denote the fractional component of  $\hat{z}(\tilde{J}^L, A^S)$  when there is one. By the discussion in the paragraph above,  $j'$  must be a small product. Since the solution  $\hat{z}(\tilde{J}^L, A^S)$  consumes all of the knapsack capacity, we have  $\delta^l = \sum_{j=1}^n v_j \hat{z}_j(\tilde{J}^L, A^S) \leq \sum_{j \in \hat{S}(\tilde{J}^L, A^S)} v_j + v_{j'} \leq \sum_{j \in \hat{S}(\tilde{J}^L, A^S)} v_j + (\delta - 1) \delta^{l-1}$ , where the first inequality follows from the fact that  $\hat{S}(\tilde{J}^L, A^S)$  includes all components of  $\hat{z}(\tilde{J}^L, A^S)$  with the exception of  $j'$  and the second inequality follows from the fact that product  $j'$  is a small product. From the last chain of inequalities, it follows that  $\sum_{j \in \hat{S}(\tilde{J}^L, A^S)} v_j \geq \delta^l - (\delta - 1) \delta^{l-1} = \delta^{l-1}$  so that the assortment  $\hat{S}(\tilde{J}^L, A^S)$  satisfies the lower bound constraint in problem (3.32). Furthermore, noting that  $\delta^l \geq \sum_{j=1}^n v_j \hat{z}_j(\tilde{J}^L, A^S) = \sum_{j \in \hat{S}(\tilde{J}^L, A^S)} v_j + v_{j'} z_{j'}(\tilde{J}^L, A^S) \geq \sum_{j \in \hat{S}(\tilde{J}^L, A^S)} v_j$ , the assortment  $\hat{S}(\tilde{J}^L, A^S)$  satisfies the upper bound constraint in problem (3.32) as well and the claim follows.

Second, we show the claim under the assumption that the solution  $\hat{z}(\tilde{J}^L, A^S)$  does not consume all of the knapsack capacity in problem (3.33) when we solve this problem with  $J^L = \tilde{J}^L$  and  $J^S = A^S$ . We recall that the discussion at the beginning of the third paragraph of the proof shows that  $r_j v_j \leq \min_{k \in A^S} r_k v_k$  for all  $j \in \tilde{J}^S \setminus A^S$ . Thus, if we solve problem (3.33) with  $J^L = \tilde{J}^L$  and  $J^S = A^S$ , then by the fourth set of constraints in this problem, the decision variables corresponding to the products in  $\tilde{J}^S \setminus A^S$  are free to take values between zero and one. In this case, since the solution  $\hat{z}(\tilde{J}^L, A^S)$  does not consume all of the knapsack capacity

in problem (3.33) and all of the objective function coefficients are positive, the decision variables corresponding to the products in  $\tilde{J}^S \setminus A^S$  must take value one. Furthermore, the decision variables corresponding to the products in  $\tilde{J}^L$  and  $A^S$  are fixed at one when we solve problem (3.33) with  $J^L = \tilde{J}^L$  and  $J^S = A^S$ . Therefore, if we solve problem (3.33) with  $J^L = \tilde{J}^L$  and  $J^S = A^S$ , then the decision variables corresponding to the products in  $\tilde{J}^L$ ,  $A^S$  and  $\tilde{J}^S \setminus A^S$  take value one, which implies that  $\delta^{l-1} \leq \sum_{j \in \tilde{S}} v_j = \sum_{j \in \tilde{J}^L} v_j + \sum_{j \in A^S} v_j + \sum_{j \in \tilde{J}^S \setminus A^S} v_j \leq \sum_{j=1}^n v_j z_j(\tilde{J}^L, A^S) = \sum_{j \in \hat{S}(\tilde{J}^L, A^S)} v_j$ , where the first inequality follows from the fact that  $\tilde{S}$  is an optimal solution to problem (3.32), the first equality uses the fact that  $\tilde{S} = \tilde{J}^L \cup \tilde{J}^S$ , the second inequality follows from the fact that the products in  $\tilde{J}^L$ ,  $A^S$  and  $\tilde{J}^S \setminus A^S$  all take value one when we solve problem (3.33) with  $J^L = \tilde{J}^L$  and  $J^S = A^S$  and the last equality follows from the fact that if the solution  $\hat{z}(\tilde{J}^L, A^S)$  does not consume all of the knapsack capacity, then it cannot have any fractional components. On the other hand, since the solution  $\hat{z}(\tilde{J}^L, A^S)$  does not have any fractional components and it is an optimal solution to problem (3.33), we obtain  $\delta^l \geq \sum_{j=1}^n v_j z_j(\tilde{J}^L, A^S) = \sum_{j \in \hat{S}(\tilde{J}^L, A^S)} v_j$ . The last two chains of inequalities show that the assortment  $\hat{S}(\tilde{J}^L, A^S)$  satisfies the constraint in problem (3.32) and the claim holds.

We proceed to checking the objective function value provided by the assortment  $\hat{S}(\tilde{J}^L, A^S)$ . To that end, letting  $\zeta(J^L, J^S)$  be the optimal objective value of problem (3.33), we begin by arguing that  $\hat{G}_l \leq \zeta(\tilde{J}^L, A^S)$ . To establish this inequality, we observe that by the definitions of  $\tilde{J}^L$  and  $\tilde{J}^S$ , the products in  $\tilde{J}^L$  and  $\tilde{J}^S$  take value one in the optimal solution to problem (3.32). On the other hand, if we solve problem (3.33) with  $J^L = \tilde{J}^L$  and  $J^S = A^S$ , then the products in  $\tilde{J}^L$  and  $A^S$  take value one in the optimal solution. Furthermore, as discussed at the beginning of the previous paragraph, the products in  $\tilde{J}^S \setminus A^S$  are free to take values between

zero and one when we solve problem (3.33). Thus, the optimal solution to problem (3.32) is a feasible solution to problem (3.33) when we solve this problem with  $J^L = \tilde{J}^L$  and  $J^S = A^S$ , which implies that  $\hat{G}_l \leq \zeta(\tilde{J}^L, A^S)$  as desired. In this case, using  $j'$  to denote the fractional component of  $\hat{z}(\tilde{J}^L, A^S)$  when there is one, we obtain  $\hat{G}_l \leq \zeta(\tilde{J}^L, A^S) = \sum_{j=1}^n r_j v_j \hat{z}_j(\tilde{J}^L, A^S) \leq \sum_{j \in \hat{S}(\tilde{J}^L, A^S)} r_j v_j + r_{j'} v_{j'} \leq \sum_{j \in \hat{S}(\tilde{J}^L, A^S)} r_j v_j + \hat{G}_l/q$ , where the last inequality follows by noting that  $r_{j'} v_{j'} \leq \hat{G}_l/q$ , which is shown in the third paragraph of the proof. Focusing on the first and last expressions in the last chain of inequalities and noting that  $q = \lceil \delta/(\delta - 1) \rceil$ , we get  $\sum_{j \in \hat{S}(\tilde{J}^L, A^S)} r_j v_j \geq ((q - 1)/q)\hat{G}_l \geq \hat{G}_l/\delta$ . So, the assortment  $\hat{S}(\tilde{J}^L, A^S)$  corresponds to a feasible solution to problem (3.32), providing an objective value to this problem that deviates from the optimal objective value by no more than a factor of  $\delta$ . Furthermore, noting that  $|\tilde{J}^L| \leq q$  and  $|A^S| = q$ , we have  $\hat{S}(\tilde{J}^L, A^S) \in \{\hat{S}(J^L, J^S) : J^L \in \wp^L, J^S \in \wp^S\}$  and the result follows.  $\square$

### 3.12.6 Proof of Theorem 3.8.2

By using the same argument at the beginning of the proof of Theorem 3.7.3, it follows that  $\hat{x} \geq 0$ . Fix an arbitrary nest  $i$ . Choose any assortment  $S_i \subset N$  within this nest. First, we consider the case where  $S_i \neq \emptyset$ . Fix  $l = l_i^L, \dots, l_i^U$  such that  $\delta^{l-1} \leq v_{i0} + \sum_{j \in S_i} v_{ij} \leq \delta^l$ . Since  $(\hat{x}, \hat{y})$  is the optimal solution to problem (3.4) after replacing the collection of assortments  $\{A_{it} : t \in \mathcal{T}_i\}$  in the second set of constraints with the assortments  $\{\hat{S}_{il} : \ell = l_i^L, \dots, l_i^U\} \cup \{\emptyset\}$ , this solution satisfies the second set of constraints in problem (3.4) for nest  $i$  and the assortment  $\hat{S}_{il}$ . Therefore, we have

$$\hat{y}_i \geq V_i(\hat{S}_{il})^{\gamma_i} (R_i(\hat{S}_{il}) - \hat{x}) = V_i(\hat{S}_{il})^{\gamma_i - 1} \left( \sum_{j \in \hat{S}_{il}} r_{ij} v_{ij} \right) - V_i(\hat{S}_{il})^{\gamma_i} \hat{x}.$$

Multiplying both sides of this inequality by  $\delta^{\bar{\gamma}+1}$ , we obtain

$$\delta^{\bar{\gamma}+1} \hat{y}_i \geq \delta^{\bar{\gamma}} V_i(\hat{S}_{il})^{\gamma_i-1} \left( \delta \sum_{j \in \hat{S}_{il}} r_{ij} v_{ij} \right) - \delta^{\bar{\gamma}+1} V_i(\hat{S}_{il})^{\gamma_i} \hat{x}. \quad (3.34)$$

We proceed to bound each one of the terms  $\delta \sum_{j \in \hat{S}_{il}} r_{ij} v_{ij}$ ,  $V_i(\hat{S}_{il})^{\gamma_i-1}$  and  $V_i(\hat{S}_{il})^{\gamma_i}$  in the inequality above. By the definition of  $\hat{S}_{il}$ , we have

$$\delta \sum_{j \in \hat{S}_{il}} r_{ij} v_{ij} \geq \hat{G}_{il} = \max_{S'_i \subset N} \left\{ \sum_{j \in S'_i} r_{ij} v_{ij} : \delta^{l-1} \leq v_{i0} + \sum_{j \in S'_i} v_{ij} \leq \delta^l \right\} \geq \sum_{j \in S_i} r_{ij} v_{ij}, \quad (3.35)$$

where the second inequality follows by noting that  $l$  is chosen such that  $\delta^{l-1} \leq v_{i0} + \sum_{j \in S_i} v_{ij} \leq \delta^l$ . The definition of  $\hat{S}_{il}$  also implies that  $\delta^{l-1} \leq v_{i0} + \sum_{j \in \hat{S}_{il}} v_{ij} = V_i(\hat{S}_{il}) \leq \delta^l$ . In this case, if we have  $\gamma_i \leq 1$ , then  $V_i(\hat{S}_{il})^{\gamma_i-1} \geq (\delta^l)^{\gamma_i-1}$ . If, on the other hand, we have  $\gamma_i > 1$ , then  $V_i(\hat{S}_{il})^{\gamma_i-1} \geq (\delta^{l-1})^{\gamma_i-1} = (\delta^l)^{\gamma_i-1} \delta^{-(\gamma_i-1)}$ . So, combining the two cases, we bound  $V_i(\hat{S}_{il})^{\gamma_i-1}$  by

$$V_i(\hat{S}_{il})^{\gamma_i-1} \geq (\delta^l)^{\gamma_i-1} \delta^{-[\gamma_i-1]^+},$$

where we use  $[a]^+ = \max\{a, 0\}$ . Noting also that  $V_i(\hat{S}_{il})^{\gamma_i} \leq (\delta^l)^{\gamma_i}$ , using these bounds on  $V_i(\hat{S}_{il})^{\gamma_i-1}$  and  $V_i(\hat{S}_{il})^{\gamma_i}$  together with the inequality in (3.35) in (3.34), we obtain

$$\begin{aligned} \delta^{\bar{\gamma}+1} \hat{y}_i &\geq \delta^{\bar{\gamma}} (\delta^l)^{\gamma_i-1} \delta^{-[\gamma_i-1]^+} \left( \sum_{j \in S_i} r_{ij} v_{ij} \right) - \delta^{\bar{\gamma}+1} (\delta^l)^{\gamma_i} \hat{x} \\ &= \delta^{\bar{\gamma}} (\delta^l)^{\gamma_i-1} \delta^{-[\gamma_i-1]^+} \left( \sum_{j \in S_i} r_{ij} v_{ij} \right) - \delta^{\bar{\gamma}+\gamma_i+1} (\delta^{l-1})^{\gamma_i} \hat{x}. \end{aligned} \quad (3.36)$$

Our choice of  $l$  at the beginning of the proof implies that  $\delta^{l-1} \leq v_{i0} + \sum_{j \in S_i} v_{ij} = V_i(S_i) \leq \delta^l$ . In this case, if we have  $\gamma_i \leq 1$ , then  $V_i(S_i)^{\gamma_i-1} \leq (\delta^{l-1})^{\gamma_i-1} = (\delta^l)^{\gamma_i-1} \delta^{1-\gamma_i}$ . If, on the other hand, we have  $\gamma_i > 1$ , then  $V_i(S_i)^{\gamma_i-1} \leq (\delta^l)^{\gamma_i-1}$ . Combining the two cases yields  $V_i(S_i)^{\gamma_i-1} \leq (\delta^l)^{\gamma_i-1} \delta^{[1-\gamma_i]^+}$  so that we can bound  $(\delta^l)^{\gamma_i-1}$  by

$$(\delta^l)^{\gamma_i-1} \geq \delta^{-[1-\gamma_i]^+} V_i(S_i)^{\gamma_i-1}.$$

Furthermore, noting that  $(\delta^{l-1})^{\gamma_i} \leq V_i(S_i)^{\gamma_i}$ , we use these bounds on  $(\delta^l)^{\gamma_i-1}$  and  $(\delta^{l-1})^{\gamma_i}$  in the two terms on the right side of (3.36) to obtain

$$\delta^{\bar{\gamma}+1} \hat{y}_i \geq \delta^{\bar{\gamma}} \delta^{-[\gamma_i-1]^+} \delta^{-[1-\gamma_i]^+} V_i(S_i)^{\gamma_i-1} \left( \sum_{j \in S_i} r_{ij} v_{ij} \right) - \delta^{\bar{\gamma}+\gamma_i+1} V_i(S_i)^{\gamma_i} \hat{x}.$$

If we have  $\gamma_i \leq 1$ , then  $\bar{\gamma} - [\gamma_i - 1]^+ - [1 - \gamma_i]^+ = \bar{\gamma} - 1 + \gamma_i \geq 0$ , where we use the fact that  $\bar{\gamma} > 1$ . On the other hand, if we have  $\gamma_i > 1$ , then  $\bar{\gamma} - [\gamma_i - 1]^+ - [1 - \gamma_i]^+ = \bar{\gamma} - \gamma_i + 1 \geq 0$  since  $\bar{\gamma} \geq \gamma_i$ . Therefore,  $\delta^{\bar{\gamma}} \delta^{-[\gamma_i-1]^+} \delta^{-[1-\gamma_i]^+} \geq 1$ . We also have  $\delta^{\bar{\gamma}+\gamma_i+1} \leq \delta^{2\bar{\gamma}+1}$ . Thus, the last inequality above yields

$$\delta^{\bar{\gamma}+1} \hat{y}_i \geq V_i(S_i)^{\gamma_i-1} \left( \sum_{j \in S_i} r_{ij} v_{ij} \right) - \delta^{2\bar{\gamma}+1} V_i(S_i)^{\gamma_i} \hat{x}.$$

Since  $\sum_{j \in S_i} r_{ij} v_{ij} / V_i(S_i) = R_i(S_i)$ , the inequality above shows that the solution  $(\delta^{2\bar{\gamma}+1} \hat{x}, \delta^{\bar{\gamma}+1} \hat{y})$  satisfies the second set of constraints in problem (3.3) for the assortment  $S_i$  and nest  $i$ , as long as  $S_i \neq \emptyset$ .

Second, we consider the case where  $S_i = \emptyset$ . The solution  $(\hat{x}, \hat{y})$  satisfies the second set of constraints in problem (3.4) for the empty assortment within nest  $i$ , in which case, we obtain  $\hat{y}_i \geq V_i(\emptyset)^{\gamma_i} (R_i(\emptyset) - \hat{x})$ . If we multiply this inequality by  $\delta^{\bar{\gamma}+1}$  and note that  $R_i(\emptyset) = 0$ , then we have  $\delta^{\bar{\gamma}+1} \hat{y}_i \geq \delta^{\bar{\gamma}+1} V_i(\emptyset)^{\gamma_i} R_i(\emptyset) - \delta^{\bar{\gamma}+1} V_i(\emptyset)^{\gamma_i} \hat{x} = V_i(\emptyset)^{\gamma_i} R_i(\emptyset) - \delta^{\bar{\gamma}+1} V_i(\emptyset)^{\gamma_i} \hat{x}$ . Replacing the term  $\delta^{\bar{\gamma}+1}$  on the right side of the last inequality with an even larger term  $\delta^{2\bar{\gamma}+1}$ , it follows that

$$\delta^{\bar{\gamma}+1} \hat{y}_i \geq V_i(\emptyset)^{\gamma_i} R_i(\emptyset) - \delta^{2\bar{\gamma}+1} V_i(\emptyset)^{\gamma_i} \hat{x}.$$

Therefore, the solution  $(\delta^{2\bar{\gamma}+1} \hat{x}, \delta^{\bar{\gamma}+1} \hat{y})$  satisfies the second set of constraints in problem (3.3) for assortment  $S_i = \emptyset$  and nest  $i$ . Combining the two cases above and noting that our choice of nest  $i$  and assortment  $S_i$  is arbitrary, we conclude that the solution  $(\delta^{2\bar{\gamma}+1} \hat{x}, \delta^{\bar{\gamma}+1} \hat{y})$  satisfies the second set of constraints in problem (3.3).

Since the solution  $(\hat{x}, \hat{y})$  is optimal to problem (3.4), we have  $v_0 \hat{x} \geq \sum_{i \in M} \hat{y}_i$ . This implies that  $v_0 \delta^{2\bar{\gamma}+1} \hat{x} \geq v_0 \delta^{\bar{\gamma}+1} \hat{x} \geq \sum_{i \in M} \delta^{\bar{\gamma}+1} \hat{y}_i$ , in which case, the solution

$(\delta^{2\bar{\gamma}+1} \hat{x}, \delta^{\bar{\gamma}+1} \hat{y})$  satisfies the first constraint in problem (3.3) as well and we obtain the desired result.  $\square$

We hope that it will be possible to extend the techniques developed in this paper to solve assortment optimization problems under richer choice models. One possible extension is to consider the case where the dissimilarity parameter of a particular nest depends on the assortment offered within this nest, in which case, the dissimilarity parameter of nest  $i$  takes the form  $\gamma_i(S_i)$ . It turns out that there are certain cases where we can deal with such assortment-dependent dissimilarity parameters. For example, consider the case where  $\gamma_i(S_i)$  depends on  $S_i$  only through the cardinality of  $S_i$  so that  $\gamma_i(S_i) = f_i(|S_i|)$  for some  $f_i(\cdot)$ . We assume that  $f_i(\cdot)$  takes values over  $[0, 1]$ , indicating that  $\gamma_i(S_i) \leq 1$  for all  $S_i \subset N$  so that we have purely competitive products. Furthermore, we assume that  $v_{i0} = 0$  for all  $i \in M$ , which implies that the nests are fully-captured. To obtain the optimal solution under assortment-dependent dissimilarity parameters, we can solve problem (3.3) after replacing  $\gamma_i$  in the second set of constraints with  $\gamma_i(S_i)$ . In this case, we observe that these constraints are equivalent to  $y_i \geq \max_{S_i \subset N} V_i(S_i)^{\gamma_i(S_i)} (R_i(S_i) - z)$  for all  $i \in M$ . Using the fact that  $\gamma_i(S_i)$  depends only on the cardinality of  $S_i$ , we can alternatively write the last set of constraints as

$$y_i \geq \max_{S_i \subset N: |S_i|=c_i} V_i(S_i)^{f_i(c_i)} (R_i(S_i) - z) \quad \forall c_i = 1, \dots, n, \quad i \in M.$$

Subsequent to our work, [20] followed up on this paper by considering constrained assortment optimization problems under the nested logit model. In particular, they consider the case where there is a cardinality constraint on the assortment offered in each nest, the products are purely competitive and the nests are fully-captured. The maximization problem on the right side of the constraint above appears when they deal with an assortment optimization problem where the cardinality of the assortment offered in nest  $i$  is constrained to be  $c_i$  and  $f_i(c_i)$  is the dissimilarity

parameter of this nest. Using a proof technique that is entirely different than our approach in this paper, Theorem 5 in [20] shows that for a given value of  $c_i$ , it is possible to construct a collection of candidate assortments  $\{A_{it} : t \in \mathcal{T}_i^{c_i}\}$  such that this collection always includes the optimal solution to the maximization problem on the right side of the constraints above. Furthermore, the collection  $\{A_{it} : t \in \mathcal{T}_i^{c_i}\}$  includes at most  $n^2$  assortments. Therefore, we can equivalently write the constraints above as  $y_i \geq V_i(S_i)^{f_i(c_i)} (R_i(S_i) - z)$  for all  $c_i = 1, \dots, n$ ,  $S_i \in \{A_{it} : t \in \mathcal{T}_i^{c_i}\}$ ,  $i \in M$ . This discussion implies that if the dissimilarity parameter of a nest depends on the cardinality of the assortment offered in this nest and we have purely competitive products and fully-captured nests, then we can obtain the optimal assortment by replacing the second set of constraints in problem (3.3) with the constraints  $y_i \geq V_i(S_i)^{f_i(c_i)} (R_i(S_i) - z)$  for all  $c_i = 1, \dots, n$ ,  $S_i \in \{A_{it} : t \in \mathcal{T}_i^{c_i}\}$ ,  $i \in M$  and solving the corresponding linear program. Since there are at most  $n^2$  assortments in the collection  $\{A_{it} : t \in \mathcal{T}_i^{c_i}\}$ , this linear program has  $1 + m$  decision variables and at most  $1 + mn^3$  constraints.

Another possible case where we can deal with assortment-dependent dissimilarity parameters occurs when a certain subset of products  $C_i \subset N$  in nest  $i$  is designated as a critical subset and if any product in  $C_i$  is offered, then the dissimilarity parameter of nest  $i$  takes value  $\Gamma_i^1$ , whereas if no product in  $C_i$  is offered, then the dissimilarity parameter of nest  $i$  takes value  $\Gamma_i^0$ . In this case, the second set of constraints in problem (3.3) are equivalent to the two sets of constraints

$$y_i \geq \max_{S_i \subset N: |S_i \cap C_i| \geq 1} V_i(S_i)^{\Gamma_i^1} (R_i(S_i) - z) \quad \forall i \in M$$

$$y_i \geq \max_{S_i \subset N: |S_i \cap C_i| = 0} V_i(S_i)^{\Gamma_i^0} (R_i(S_i) - z) \quad \forall i \in M.$$

[20] consider parent product constraints in assortment optimization problems under the nested logit model. Parent product constraints designate a subset of products in each nest as parent products. Each parent product has a set of child products.



A child product cannot be offered unless its parent product is offered. They show that it is tractable to solve assortment optimization problems under the nested logit model with parent product constraints. By using their approach, it is possible to show that we can construct a collection of candidate assortments  $\{A_{it} : t \in \mathcal{T}_i^1\}$  such that this collection always includes the optimal solution to the maximization problem on the right side of the first set of constraints above. Furthermore, the collection  $\{A_{it} : t \in \mathcal{T}_i^1\}$  includes at most  $n^2$  assortments. On the other hand, the maximization problem on the right side of the second set of constraints can be solved by dropping the products in  $C_i$  from consideration and focusing only on the products in  $N \setminus C_i$ . In this case, by using the approach used in this paper, we can show that a nested-by-revenue assortment solves the maximization problem on right side of the second set of constraints above, as long as we focus on the products in  $N \setminus C_i$ . Therefore, we can construct a collection of candidate assortments  $\{A_{it} : t \in \mathcal{T}_i^0\}$  such that this collection always includes the optimal solution to the maximization problem on the right side of the second set of constraints above and there are at most  $1 + n$  assortments in this collection. Thus, the two sets of constraints above can succinctly be written as  $y_i \geq V_i(S_i)^{\Gamma_i^1} (R_i(S_i) - z)$  for all  $S_i \in \{A_{it} : t \in \mathcal{T}_i^1\}$ ,  $i \in M$  and  $y_i \geq V_i(S_i)^{\Gamma_i^0} (R_i(S_i) - z)$  for all  $S_i \in \{A_{it} : t \in \mathcal{T}_i^0\}$ ,  $i \in M$ . In this case, we can obtain the optimal assortment replacing the second set of constraints in problem (3.3) with the constraints  $y_i \geq V_i(S_i)^{\Gamma_i^1} (R_i(S_i) - z)$  for all  $S_i \in \{A_{it} : t \in \mathcal{T}_i^1\}$ ,  $i \in M$  and  $y_i \geq V_i(S_i)^{\Gamma_i^0} (R_i(S_i) - z)$  for all  $S_i \in \{A_{it} : t \in \mathcal{T}_i^0\}$ ,  $i \in M$  and solving the corresponding linear program. This linear program has  $1+m$  decision variables and  $1 + m(1 + n + n^2)$  constraints since there are respectively at most  $n^2$  and  $1 + n$  constraints in the collections  $\{A_{it} : t \in \mathcal{T}_i^1\}$  and  $\{A_{it} : t \in \mathcal{T}_i^0\}$ . We can extend this approach to cover the case where the number of possible values of the dissimilarity parameter is more than two. For example, we can extend this

approach to cover the case where the number of possible values of the dissimilarity parameter of nest  $i$  are  $\{\Gamma_0, \Gamma_1, \dots, \Gamma_K\}$  and we have  $\gamma_i(S_i) = \Gamma_i^0$  when  $|S_i \cap C_i| = 0$ , whereas  $\gamma_i(S_i) = \Gamma_i^k$  when  $|S_i \cap C_i| = k$  for  $k = 1, \dots, K - 1$  and  $\gamma_i(S_i) = \Gamma_i^K$  when  $|S_i \cap C_i| \geq K$ .

The previous two paragraphs describe some cases where we can deal with assortment-dependent dissimilarity parameters, as long as we have purely competitive products and fully-captured nests. One direction for further research is to investigate other cases where we can allow the dissimilarity parameter of a nest to depend on the assortment offered within this nest. Such cases will clearly provide more flexibility in modeling customer choices. Another possible direction for further investigation is to work with more general forms of the nested logit model, such as a mixture of nested logit models or the cross nested logit models. It is not immediately clear how the approach that we used in this paper can be extended to solve assortment optimization problems under the more general forms of the nested logit model. In particular, our approach exploits the fact that the second set of constraints in problem (3.3) separates by the nests for a fixed value of the decision variable  $x$ . We lose this separable structure when solving assortment optimization problems under more general forms of the nested logit model. Thus, solving assortment optimization problems under general forms of the nested logit model is still open for further research.

In Section 3.5, we show that nested-by-revenue assortments are optimal when the dissimilarity parameters are less than one and the nests are fully-captured, but by the discussion in the previous paragraph, this result does not hold under assortment-dependent dissimilarity parameters. In Section 3.6.2, we show that nested-by-revenue assortments provide the performance guarantee given by the

expression in (3.6) when we have fully-captured nests. For the problem instance above, one can check that the expression in (3.6) is no larger than  $10/9$ , irrespective of the value of  $k$ . However, the discussion in the previous paragraph shows that nested-by-revenue assortments perform arbitrarily badly when we choose a large value for  $k$ , indicating that they cannot provide a performance guarantee of  $10/9$ . In Section 3.7, we show that the collection of assortments  $\{N_{ij}^k : k \in N, j = 0, \dots, k\} \cup \{\{j\} : j \in N\}$  provide a performance guarantee of two when the dissimilarity parameters are less than one. Noting the definition of  $N_{ij}^k$  at the end of Section 3.7, the collection of assortments  $\{N_{ij}^k : k \in N, j = 0, \dots, k\} \cup \{\{j\} : j \in N\}$  is given by  $\{\emptyset, \{1\}, \{1, 2\}, \{1, 2, 3\}, \{2\}, \{2, 3\}, \{3\}\}$ , but as we describe in the previous paragraph, all of the assortments in this collection can perform arbitrarily badly when we choose a large value for  $k$ . So, they cannot provide a performance guarantee of two under assortment-dependent dissimilarity parameters. Finally, in Section 3.8.1, we show that the same collection of assortments provide the performance guarantee given by the expression in (3.12). For the problem instance above, we can check that the expression in (3.12) is equal to two, irrespective of the value of  $k$ . However, as we just described, these assortments cannot provide a performance guarantee of two.

## CHAPTER 4

# QUALITY CONSISTENT PRICING UNDER THE NESTED LOGIT MODEL

### 4.1 Introduction

In this chapter, we consider pricing problems when customers choose according to the nested logit model and there is a quality consistency constraint on the prices charged for the products. In the quality consistency constraint that we impose on the prices, there is an intrinsic ordering between the qualities of the products. The quality consistency constraint ensures that the prices charged for the products of higher quality are also larger. The goal is to find the prices to charge for the products to maximize the expected revenue obtained from a customer, while making sure that the prices satisfy the quality consistency constraint.

We consider two types of quality consistency constraint. In the first type of constraint, there is an intrinsic ordering between the qualities of the products in each nest. We refer to this quality consistency constraint as *price ladders inside nests*. Figure 4.1.b illustrates this quality consistency constraint with three products in each nest and the price of product  $j$  in nest  $i$  is denoted by  $p_{ij}$ . The products in each nest are indexed such that the third product is of higher quality than the second product in the same nest, which is, in turn, of higher quality than the first product. Therefore, the price of the third product should be larger than the price of the second product, which should, in turn, be larger than the price of the first product. There is no dictated ordering between the qualities or prices of the products in different nest. As an example of a situation where this type of quality consistency constraint becomes relevant, we consider the case where the nests correspond to different brands and the products within a particular nest correspond

to the variants of a particular brand with different qualities. There is a verifiable ordering between the qualities of the different variants of a particular brand and the customers expect that the prices for the variants of higher quality should also be larger. On the other hand, it is difficult to compare the variants of different brands in terms of quality and there is no reason for the customers to expect a particular ordering between the prices for the variants of different brands.

In the second type of constraint, there is an intrinsic ordering between the qualities of the nests, but there is no clear ordering between the qualities of the products in the same nest. We refer to this quality consistency constraint as *price ladders between nests*. Figure 4.1.c illustrates this quality consistency constraint with three nests. The nests are indexed such that the third nest corresponds to a higher quality level than the second nest, which, in turn, corresponds to a higher quality level than the first nest. So, the price for any product in the third nest should be larger than the price for any product in the second nest, which should, in turn, be larger than the price for any product in the first nest. As an example of a situation where this type of quality consistency constraint becomes relevant, we consider the case where the nests correspond to different quality levels and the products within a particular nest correspond to products that differ in cosmetic or personal features, such as color. The customers expect that the prices for the products in a nest corresponding to a higher quality level are larger than the prices for the products in a nest corresponding to a lower quality level, but there is no reason for the customers to expect a particular ordering between the prices for the products in a particular nest since these products differ in cosmetic or personal features.

Although the two types of quality consistency constraints cover a variety of useful situations, we observe that there can be other quality consistency constraints

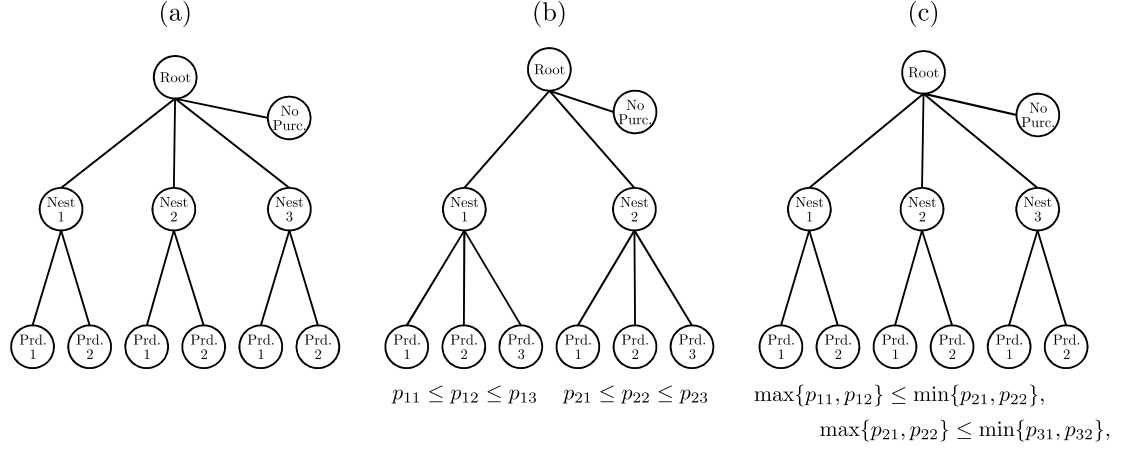


Figure 4.1: Nested logit model, price ladders inside nests and price ladders between nests.

that are not covered by our results. In our conclusions, we discuss some possible extensions of our quality consistency constraints.

In this chapter, we give algorithms to find the optimal prices to charge under both types of quality consistency constraints. In our setting, there are  $m$  nests and  $n$  products in each nest. The price of each product is chosen within a finite set of possible prices and there are  $q$  possible prices for each product. Therefore, the vector of prices charged for the products in a nest takes values in  $\mathfrak{R}^n$  and each component of this vector takes one of the  $q$  possible values, which implies that there are  $O(q^n)$  possible price vectors that we can charge for the products in a nest. Under price ladders within nests, we show that the optimal price vector to charge in a nest is one of at most  $nq$  candidate price vectors and all of these candidate price vectors can be constructed by solving a linear program through the parametric simplex method. The linear program that we use to come up with the candidate price vectors has  $O(nq)$  decision variables and  $O(nq^2)$  constraints. This result reduces the number of possible price vectors to consider for each nest from

$O(q^n)$  to  $O(nq)$ . However, although the optimal price vector to charge in each nest is one of  $O(nq)$  candidate price vectors, computing the optimal prices to charge over all nests can still be challenging, since there are  $O((nq)^m)$  different ways of combining  $nq$  candidate price vectors from  $m$  nests. To deal with this difficulty, we give a linear program with  $O(m)$  decision variables and  $O(mnq)$  constraints that finds the optimal combination of price vectors to charge in different nests. Thus, we solve a linear program with  $O(nq)$  decision variables and  $O(nq^2)$  constraints by using the parametric simplex method to come up with  $O(nq)$  candidate price vectors for each nest. We find the optimal combination of candidate price vectors to charge in different nests by solving another linear program with  $O(m)$  decision variables and  $O(mnq)$  constraints.

Pricing problems under price ladders between nests are considerably more difficult than the ones under price ladders inside nests since price ladders between nests create interactions between the prices charged for the products in different nests. Under price ladders between nests, we show that the optimal price vector to charge in a nest is one of at most  $nq^3$  candidate price vectors and all of these candidate price vectors can be constructed by solving a linear program through the parametric simplex method. The linear program that we use to come up with the candidate price vectors has  $O(nq)$  decision variables and  $O(n)$  constraints. To find the optimal combination of price vectors to charge in different nests, we give a linear program with  $O(mq)$  decision variables and  $O(mnq^4)$  constraints. In our numerical experiments, we consider test problems with as many as  $m = 6$  nests,  $n = 30$  products in each nest and  $q = 30$  possible prices for each product, yielding a total of 180 products. Under price ladders inside nests, we can compute the optimal prices to charge for the products in about two seconds, whereas under price ladders between nests, we can compute the optimal prices in about two minutes.

In addition to providing algorithms that find the optimal prices under price ladders inside nests and between nests, we make practically useful contributions through our formulation of the pricing problem. In our formulation, the price of each product is chosen within a finite set of possible prices and the set of possible prices for a product is defined by the modeler. The modeler can design the set of possible prices for a product to correspond to the prices that are commonly used in retail, such as prices that end in 99 cents or prices that are in increments of 10 dollars. Furthermore, the nested logit model commonly assumes that there is a parametric relationship between the attractiveness of a product and its price. For example, it is common to assume that if the price charged for product  $j$  in nest  $i$  is  $p_{ij}$ , then the attractiveness of this product is given by  $\exp(\alpha_{ij} - \beta_{ij} p_{ij})$ , where  $\alpha_{ij}$  and  $\beta_{ij}$  are fixed parameters; see [34] and [21]. Our formulation of the pricing problem does not assume a parametric relationship between the attractiveness of a product and its price, allowing the attractiveness of a product to depend on its price in an arbitrary fashion.

## 4.2 Literature Review

There is a significant amount of work on solving pricing problems under variants of the multinomial and nested logit models. Under the multinomial logit model, [25] observe that the expected revenue is not a concave function of the prices for the products. [55] and [12] solve the pricing problem by expressing the expected revenue as a function of the market shares of the products and showing that the expected revenue is a concave function of the market shares. [8] and [63] give tractable solution methods for joint assortment optimization and pricing problems under the multinomial logit model, where the set of products offered to customers, as well as the prices of the offered products, are decision variables. [68] discuss



revenue management problems, where the prices charged for the products are dynamically adjusted over time as a function of the remaining time in the selling horizon and the remaining inventory for the products. [10] show that pricing problems under the multinomial logit model with a finite set of possible prices can be formulated as a linear program. [29] study pricing problems, where the attractiveness of a product depends on its price in a general fashion and there are constraints on the expected number of sales for the products.

The literature on solving pricing problems under the nested logit model has recently started growing. [34] study pricing problems under the assumption that the products in the same nest have the same price sensitivity. They show that if the products in the same nest have the same price sensitivity, then the pricing problem can be formulated as a convex program. [21] consider the case where the products in the same nest do not necessarily have the same price sensitivity. They show that the expected revenue function can have multiple local maxima and show how to find a local maximum of the expected revenue function. The authors also give sufficient conditions that eliminate multiple local maxima. [47] develop algorithms for computing solutions with a specified performance guarantee even when there are multiple local maxima of the expected revenue function. [35] and [33] consider pricing problems under the nested logit model, where the choice process proceeds in more than two stages. The earlier work on pricing problems under the nested logit model does not consider quality consistency constraints.

Charging quality consistent prices is practically important since such prices convey a sense of fairness to customers. [48] consider quality consistent pricing problems. Their work is motivated by a pricing problem in the automobile industry, where the prices of the automobiles with richer features should also be larger. They use a nonparametric choice model, show that the corresponding pricing problem

with a quality consistency constraint is NP hard and provide an approximation algorithm. [19] discuss fairness issues when providing upgrades to customers. In the context of airline industry, they point out that if the customers need to be upgraded, then the customers with low fare class reservations should be upgraded to a relatively lower fare class, when compared with the customers with high fare class reservations. They develop fluid models for revenue management problems with upgrade possibilities. They show that if the prices satisfy a certain quality consistency constraint, then their model upgrades customers with low fare class reservations to a relatively lower fare class, when compared with the customers with high fare class reservations.

A useful approach for solving optimization problems under the nested logit model is to construct a small collection of candidate solutions for each nest and to solve a linear program to combine the candidate solutions for the different nests. [20] and [16] follow this approach for assortment optimization problems, where the prices of the products are fixed and the goal is to find a set of products to offer to customers to maximize the expected revenue obtained from a customer. The development in this paper is based on this general approach as well, but there are two important challenges that need to be overcome when using this approach for pricing problems. First, constructing a small collection of candidate price vectors to charge in each nest carefully exploits the structure of the pricing problem. In particular, we make use of the fact that the attractiveness of a product is decreasing in its price and it is not clear how to construct a small collection of candidate price vectors when the attractiveness of a product is not necessarily decreasing in its price. Second, under price ladders between nests, the prices charged in different nests interact with each other since the price for any product in a nest corresponding to a higher quality level should be larger than the price for any

product in a nest corresponding to a lower quality level. Due to the interactions between the prices charged in different nests, finding the optimal combination of price vectors to choose in each nest becomes difficult. We address this difficulty by using the linear programming formulation of a dynamic program that finds the best combination of the candidate price vectors for the different nests.

### 4.3 Price Ladders Inside Nests

In this section, we consider the case with price ladders inside nests. In this setting, there is an intrinsic ordering between the qualities of the products in the same nest and the prices for the products of higher quality should also be larger. There is no intrinsic ordering between the qualities or the prices of the products in different nests.

#### 4.3.1 Problem Formulation

There are  $m$  nests and we index the nests by  $M = \{1, \dots, m\}$ . In each nest, there are  $n$  products and we index the products in each nest by  $N = \{1, \dots, n\}$ . For each product, there are  $q$  possible prices. The possible prices for a product are given by  $\Theta = \{\theta^1, \dots, \theta^q\}$ . Without loss of generality, we index the possible prices so that  $0 < \theta^1 < \theta^2 < \dots < \theta^q$ . We use  $p_{ij} \in \Theta$  to denote the price that we charge for product  $j$  in nest  $i$ . If we charge price  $p_{ij}$  for product  $j$  in nest  $i$ , then the preference weight of this product is given by  $v_{ij}(p_{ij})$ . If we charge a larger price for a product, then its preference weight becomes smaller, implying that  $v_{ij}(\theta^1) > v_{ij}(\theta^2) > \dots > v_{ij}(\theta^q) > 0$ . Our notation so far implies that the number of products in each nest is the same and the set of possible prices that we can charge for each product is the same. However, this assumption is only for notational brevity and our results

in the paper continue to hold with straightforward modifications when there are different numbers of products in different nests and the sets of possible prices for the different products are different.

We use  $p_i = (p_{i1}, \dots, p_{in}) \in \Theta^n$  to capture the price vector charged in nest  $i$ . As a function of the price vector  $p_i$  charged in nest  $i$ , we use  $V_i(p_i)$  to denote the total preference weight of the products in nest  $i$ , so that  $V_i(p_i) = \sum_{j \in N} v_{ij}(p_{ij})$ . Under the nested logit model, if we charge the price vector  $p_i$  in nest  $i$ , then a customer that has already decided to make a purchase in nest  $i$  chooses product  $j$  in this nest with probability  $v_{ij}(p_{ij})/V_i(p_i)$ . In this case, if we charge the price vector  $p_i$  in nest  $i$  and a customer has already decided to make a purchase in this nest, then the expected revenue obtained from this customer is given by

$$R_i(p_i) = \sum_{j \in N} p_{ij} \frac{v_{ij}(p_{ij})}{V_i(p_i)} = \frac{\sum_{j \in N} p_{ij} v_{ij}(p_{ij})}{V_i(p_i)}. \quad (4.1)$$

For each nest  $i$ , the nested logit model has a parameter  $\gamma_i \in [0, 1]$  characterizing the degree of dissimilarity between the products in this nest. We use  $v_0$  to denote the preference weight of the no purchase option. Under the nested logit model, if we charge the price vectors  $(p_1, \dots, p_m) \in \Theta^{m \times n}$  over all nests, then a customer decides to make a purchase in nest  $i$  with probability  $Q_i(p_1, \dots, p_m) = V_i(p_i)^{\gamma_i} / (v_0 + \sum_{l \in M} V_l(p_l)^{\gamma_l})$ . The last expression provides the probability that a customer chooses nest  $i$  as a function of the prices charged for all products in all nests. The parameter  $\gamma_i$  magnifies or dampens the preference weights of the products in nest  $i$ .

According to the nested logit model, if we charge the price vectors  $(p_1, \dots, p_m)$  over all nests, then a customer decides to make a purchase in nest  $i$  with probability  $Q_i(p_1, \dots, p_m) = V_i(p_i)^{\gamma_i} / (v_0 + \sum_{l \in M} V_l(p_l)^{\gamma_l})$ . If the customer decides to make a purchase in nest  $i$ , then the expected revenue obtained from this customer is  $R_i(p_i)$ . Thus, if we charge the price vectors  $(p_1, \dots, p_m)$  over all nests, then the expected

revenue from a customer is given by

$$\Pi(p_1, \dots, p_m) = \sum_{i \in M} Q_i(p_1, \dots, p_m) R_i(p_i) = \frac{\sum_{i \in M} V_i(p_i)^{\gamma_i} R_i(p_i)}{v_0 + \sum_{i \in M} V_i(p_i)^{\gamma_i}}. \quad (4.2)$$

Our goal is to find the price vectors  $(p_1, \dots, p_m)$  to charge over all nests to maximize the expected revenue above subject to the constraint that the price vector charged in each nest satisfies a price ladder constraint. To formulate the price ladder constraint, without loss of generality, we index the products in each nest such that products with larger indices are of higher quality. In other words, the products  $N = \{1, \dots, n\}$  in each nest are indexed in the order of increasing quality. The price ladder constraint ensures that the price for a product of higher quality is larger. That is, the price ladder constraint in nest  $i$  ensures that  $p_{i1} \leq p_{i2} \leq \dots \leq p_{in}$ . Thus, the set of feasible price vectors in nest  $i$  can be written as  $\mathcal{F}_i = \{p_i \in \Theta^n : p_{ij} \geq p_{i,j-1} \forall j \in N \setminus \{1\}\}$ . We want to find the price vectors to charge over all nests to maximize the expected revenue from a customer while satisfying the price ladder constraint, yielding the problem

$$z^* = \max_{\substack{(p_1, \dots, p_m) \in \Theta^{m \times n} : \\ p_i \in \mathcal{F}_i \forall i \in M}} \left\{ \Pi(p_1, \dots, p_m) \right\}. \quad (4.3)$$

In the problem above, the price of each product takes values in the discrete set  $\Theta$ . Furthermore, the objective function depends on the prices of the products in a nonlinear fashion. Thus, this problem is a nonlinear combinatorial optimization problem.

We emphasize two useful advantages of our formulation of problem (4.3). First, since the price for each product is chosen among a set of possible prices given by  $\Theta$  and we can design  $\Theta$  in any way we want, our formulation allows choosing the prices of the products among the prices that are commonly used in retail, such as prices that end in 99 cents or prices that are in increments of 10 dollars. Second,

the nested logit model commonly assumes a fixed functional relationship between the price of a product and its preference weight. For example, as a function of the price  $p_{ij}$  of product  $j$  in nest  $i$ , it is common to assume that the preference weight  $v_{ij}(p_{ij})$  of this product is given by  $v_{ij}(p_{ij}) = \exp(\alpha_{ij} - \beta_{ij} p_{ij})$ , where  $\alpha_{ij}$  and  $\beta_{ij}$  are fixed parameters. In contrast, our formulation of problem (4.3) does not rely on such a fixed functional relationship and we allow the dependence between  $v_{ij}(p_{ij})$  and  $p_{ij}$  to be arbitrary, as long as  $v_{ij}(p_{ij})$  is decreasing in  $p_{ij}$ .

### 4.3.2 Connection to a Fixed Point Representation

In this section, we answer a question that becomes critical when developing a tractable solution approach for problem (4.3). Assume that we have a collection of candidate price vectors  $\mathcal{P}_i = \{p_i^t : t \in \mathcal{T}_i\}$  to charge in nest  $i$  and all of the price vectors in the collection  $\mathcal{P}_i$  satisfy the price ladder constraint in the sense that  $p_i^t \in \mathcal{F}_i$  for all  $t \in \mathcal{T}_i$ . We know that we can stitch together an optimal solution to problem (4.3) by picking one price vector from each one of the candidate collections  $\mathcal{P}_1, \dots, \mathcal{P}_m$ . In other words, we know that there exists an optimal solution  $(p_1^*, \dots, p_m^*)$  to problem (4.3) that satisfies  $p_i^* \in \mathcal{P}_i$  for all  $i \in M$ . The question that we want to answer is how we can pick a price vector  $p_i^*$  from the collection  $\mathcal{P}_i$  for each nest  $i$  such that the solution  $(p_1^*, \dots, p_m^*)$  is indeed optimal to problem (4.3). It is difficult to answer this question through complete enumeration since complete enumeration requires checking the expected revenues from  $|\mathcal{P}_1| \times \dots \times |\mathcal{P}_m|$  possible solutions, which quickly gets intractable when the number of nests is large. To answer this question, we relate problem (4.3) to the problem of computing the fixed point of an appropriately defined function. In particular, for  $z \in \mathfrak{R}_+$ , we define

$f(z)$  as

$$f(z) = \sum_{i \in M} \max_{p_i \in \mathcal{P}_i} \left\{ V_i(p_i)^{\gamma_i} (R_i(p_i) - z) \right\}. \quad (4.4)$$

The value of  $\hat{z}$  satisfying  $v_0 \hat{z} = f(\hat{z})$  is the fixed point of the function  $f(\cdot)/v_0$ . Since  $v_0 z$  is increasing and  $f(z)$  is decreasing in  $z$  with  $f(0) \geq 0$ , there exists  $\hat{z}$  satisfying  $v_0 \hat{z} = f(\hat{z})$ . In the next theorem, we show that we can use this value of  $\hat{z}$  to construct an optimal solution to problem (4.3). In this theorem, we recall that  $z^*$  corresponds to the optimal objective value of problem (4.3).

**Theorem 4.3.1.** *Assume that we have a collection of candidate price vectors  $\mathcal{P}_i$  for each nest  $i$  such that we can stitch together an optimal solution to problem (4.3) by picking one price vector from each one of the collections  $\mathcal{P}_1, \dots, \mathcal{P}_m$ . Let the value of  $\hat{z}$  be such that  $v_0 \hat{z} = f(\hat{z})$  and  $\hat{p}_i$  be an optimal solution to the problem*

$$\max_{p_i \in \mathcal{P}_i} \left\{ V_i(p_i)^{\gamma_i} (R_i(p_i) - \hat{z}) \right\}. \quad (4.5)$$

*Then, we have  $\Pi(\hat{p}_1, \dots, \hat{p}_m) \geq z^*$ .*

*Proof.* We use  $(p_1^*, \dots, p_m^*)$  to denote an optimal solution to problem (4.3). By our assumption, we can stitch together an optimal solution to problem (4.3) by picking one price vector from each one of the collections  $\mathcal{P}_1, \dots, \mathcal{P}_m$ . Thus, we can assume that  $p_i^* \in \mathcal{P}_i$  for all  $i \in M$ , which implies that solution  $p_i^*$  is feasible to the problem on the right side of (4.4) and we get  $f(\hat{z}) \geq \sum_{i \in M} V_i(p_i^*)^{\gamma_i} (R_i(p_i^*) - \hat{z})$ . In this case, noting the fact that  $v_0 \hat{z} = f(\hat{z})$ , we have  $v_0 \hat{z} \geq \sum_{i \in M} V_i(p_i^*)^{\gamma_i} (R_i(p_i^*) - \hat{z})$ . Solving for  $\hat{z}$  in the last inequality, we obtain  $\hat{z} \geq \sum_{i \in M} V_i(p_i^*)^{\gamma_i} R_i(p_i^*) / (v_0 + \sum_{i \in M} V_i(p_i^*)^{\gamma_i})$ . Noting that  $z^* = \Pi(p_1^*, \dots, p_m^*) = \sum_{i \in M} V_i(p_i^*)^{\gamma_i} R_i(p_i^*) / (v_0 + \sum_{i \in M} V_i(p_i^*)^{\gamma_i})$  by the definition of  $\Pi(p_1, \dots, p_m)$  in (4.2), the last inequality implies that  $\hat{z} \geq z^*$ . Thus, to finish the proof, it is enough to show that  $\Pi(\hat{p}_1, \dots, \hat{p}_m) = \hat{z}$ . Since  $\hat{p}_i$  is an optimal solution to problem (4.5), by the definition of  $f(z)$  in (4.4) and the fact that

$v_0 \hat{z} = f(\hat{z})$ , we have  $v_0 \hat{z} = f(\hat{z}) = \sum_{i \in M} V_i(\hat{p}_i)^{\gamma_i} (R_i(\hat{p}_i) - \hat{z})$ . In this case, focusing on the first and last expressions in the last chain of equalities and solving for  $\hat{z}$ , we obtain  $\hat{z} = \sum_{i \in M} V_i(\hat{p}_i)^{\gamma_i} R_i(\hat{p}_i) / (v_0 + \sum_{i \in M} V_i(\hat{p}_i)^{\gamma_i})$  and the desired result follows by noting that  $\Pi(\hat{p}_1, \dots, \hat{p}_m) = \sum_{i \in M} V_i(\hat{p}_i)^{\gamma_i} R_i(\hat{p}_i) / (v_0 + \sum_{i \in M} V_i(\hat{p}_i)^{\gamma_i})$ .  $\square$

Theorem 4.3.1 suggests the following approach to obtain an optimal solution to problem (4.3). Assume that we have a collection of candidate price vectors  $\mathcal{P}_i = \{p_i^t : t \in \mathcal{T}_i\}$  for each nest  $i$  such that we can stitch together an optimal solution to problem (4.3) by picking one price vector from each one of the collections  $\mathcal{P}_1, \dots, \mathcal{P}_m$ . Furthermore, assume that each one of the price vectors in the candidate collection  $\mathcal{P}_i = \{p_i^t : t \in \mathcal{T}_i\}$  satisfies the price ladder constraint in the sense that  $p_i^t \in \mathcal{F}_i$  for all  $p_i^t \in \mathcal{P}_i$ . To obtain an optimal solution to problem (4.3), we find the value of  $\hat{z}$  that satisfies  $v_0 \hat{z} = f(\hat{z})$ . In this case, if we let  $\hat{p}_i$  be an optimal solution to problem (4.5), then it follows from Theorem 4.3.1 that  $\Pi(\hat{p}_1, \dots, \hat{p}_m) \geq z^*$ . Furthermore, since  $p_i^t \in \mathcal{F}_i$  for all  $p_i^t \in \mathcal{P}_i$ , the solution  $(\hat{p}_1, \dots, \hat{p}_m)$  is feasible to problem (4.3). Therefore,  $(\hat{p}_1, \dots, \hat{p}_m)$  is a feasible solution to problem (4.3) and provides an objective value to problem (4.3) that is at least as large as the optimal objective value of this problem, which implies that  $(\hat{p}_1, \dots, \hat{p}_m)$  is an optimal solution to problem (4.3), as desired. We observe that the discussion in this paragraph also provides an answer to the question that we ask at the beginning of this section. In particular, if we know that we can stitch together an optimal solution to problem (4.3) by picking one price vector from each one of the collections  $\mathcal{P}_1, \dots, \mathcal{P}_m$ , then we can use Theorem 4.3.1 to obtain an optimal solution to problem (4.3).

One remaining question is how we can find the value of  $\hat{z}$  that satisfies  $v_0 \hat{z} = f(\hat{z})$  in a tractable fashion. Noting that  $v_0 z$  is increasing and  $f(z)$  is decreasing in  $z$ , we can find the value of  $\hat{z}$  that satisfies  $v_0 \hat{z} = f(\hat{z})$  by solving the problem



$\min\{z : v_0 z \geq \sum_{i \in M} \max_{p_i \in \mathcal{P}_i} V_i(p_i)^{\gamma_i} (R_i(p_i) - z)\}$ , where the decision variable is  $z$ . The constraint in this problem is nonlinear in  $z$ , but we can linearize the constraint by using the additional decision variables  $y = (y_1, \dots, y_m)$  with the interpretation that  $y_i = \max_{p_i \in \mathcal{P}_i} V_i(p_i)^{\gamma_i} (R_i(p_i) - z)$ . In this case, we can find the value of  $\hat{z}$  that satisfies  $v_0 \hat{z} = f(\hat{z})$  by solving the problem

$$\min \left\{ z : v_0 z \geq \sum_{i \in M} y_i, y_i \geq V_i(p_i)^{\gamma_i} (R_i(p_i) - z) \quad \forall p_i \in \mathcal{P}_i, i \in M \right\}, \quad (4.6)$$

where the decision variables are  $(z, y)$ . The problem above is a linear program with  $O(m)$  decision variables and  $\sum_{i \in M} O(|\mathcal{P}_i|)$  constraints, which is tractable as long as the numbers of price vectors in the collections  $\mathcal{P}_1, \dots, \mathcal{P}_m$  are relatively small. In the rest of our discussion, we focus on how to construct a small collection of candidate price vectors  $\mathcal{P}_i$  for each nest  $i$  such that we can stitch together an optimal solution to problem (4.3) by picking one price vector from each one of the collections  $\mathcal{P}_1, \dots, \mathcal{P}_m$ . Once we have these collections, we can solve problem (4.6) to find  $\hat{z}$  satisfying  $v_0 \hat{z} = f(\hat{z})$  and we can use Theorem 4.3.1 to obtain an optimal solution to problem (4.3).

### 4.3.3 Characterizing Candidate Price Vectors

In this section, we give a characterization of the optimal price vector to charge in each nest. This characterization ultimately becomes useful to construct a collection of candidate price vectors  $\mathcal{P}_i$  for each nest  $i$  such that we can stitch together an optimal solution to problem (4.3) by picking one price vector from each one of the collections  $\mathcal{P}_1, \dots, \mathcal{P}_m$ . In the next lemma, we begin by giving a simple condition for a solution to provide an objective value for problem (4.3) that is at least as large as the optimal objective value. Subsequent to this lemma, we build on this condition to give a characterization of the optimal price vector to charge in each

nest.

**Lemma 4.3.2.** *Let  $(p_1^*, \dots, p_m^*)$  be an optimal solution to problem (4.3) providing the objective value  $z^*$ . If  $(\hat{p}_1, \dots, \hat{p}_m)$  satisfies*

$$V_i(\hat{p}_i)^{\gamma_i} (R_i(\hat{p}_i) - z^*) \geq V_i(p_i^*)^{\gamma_i} (R_i(p_i^*) - z^*)$$

for all  $i \in M$ , then we have  $\Pi(\hat{p}_1, \dots, \hat{p}_m) \geq z^*$ .

*Proof.* Adding the inequality in the lemma over all  $i \in M$  yields  $\sum_{i \in M} V_i(\hat{p}_i)^{\gamma_i} (R_i(\hat{p}_i) - z^*) \geq \sum_{i \in M} V_i(p_i^*)^{\gamma_i} (R_i(p_i^*) - z^*)$ . Since  $(p_1^*, \dots, p_m^*)$  is an optimal solution to problem (4.3), we have  $z^* = \sum_{i \in M} V_i(p_i^*)^{\gamma_i} R_i(p_i^*) / (v_0 + \sum_{i \in M} V_i(p_i^*)^{\gamma_i})$ . Arranging the terms in this equality, we obtain  $v_0 z^* = \sum_{i \in M} V_i(p_i^*)^{\gamma_i} (R_i(p_i^*) - z^*)$ , in which case, it follows that  $\sum_{i \in M} V_i(\hat{p}_i)^{\gamma_i} (R_i(\hat{p}_i) - z^*) \geq \sum_{i \in M} V_i(p_i^*)^{\gamma_i} (R_i(p_i^*) - z^*) = v_0 z^*$ . Focusing on the first and last terms in this chain of inequalities and solving for  $z^*$ , we get  $\sum_{i \in M} V_i(\hat{p}_i)^{\gamma_i} R_i(\hat{p}_i) / (v_0 + \sum_{i \in M} V_i(\hat{p}_i)^{\gamma_i}) \geq z^*$  and the desired result follows by noting that  $\Pi(\hat{p}_1, \dots, \hat{p}_m) = \sum_{i \in M} V_i(\hat{p}_i)^{\gamma_i} R_i(\hat{p}_i) / (v_0 + \sum_{i \in M} V_i(\hat{p}_i)^{\gamma_i})$ .  $\square$

In the next theorem, we build on the condition given in Lemma 4.3.2 to give a characterization of the optimal price vector to charge in each nest.

**Theorem 4.3.3.** *Let  $(p_1^*, \dots, p_m^*)$  be an optimal solution to problem (4.3) providing the objective value  $z^*$  and set  $u_i^* = \max\{\gamma_i z^* + (1 - \gamma_i) R_i(p_i^*), z^*\}$ . If  $\hat{p}_i$  is an optimal solution to the problem*

$$\max_{p_i \in \mathcal{F}_i} \left\{ V_i(p_i) (R_i(p_i) - u_i^*) \right\}, \quad (4.7)$$

then  $(\hat{p}_1, \dots, \hat{p}_m)$  is an optimal solution to problem (4.3).

*Proof.* For notational brevity, we let  $R_i^* = R_i(p_i^*)$ ,  $V_i^* = V_i(p_i^*)$ ,  $\hat{R}_i = R_i(\hat{p}_i)$  and  $\hat{V}_i = V_i(\hat{p}_i)$ . We claim that  $\hat{V}_i^{\gamma_i} (\hat{R}_i - z^*) \geq (V_i^*)^{\gamma_i} (R_i^* - z^*)$  for all  $i \in$

$M$ . To see this claim, we consider a nest  $i$  that satisfies  $R_i^* \geq z^*$ . Since  $p_i^*$  is a feasible but not necessarily an optimal solution to problem (4.7), we have  $\hat{V}_i(\hat{R}_i - u_i^*) \geq V_i^*(R_i^* - u_i^*)$ . Since  $R_i^* \geq z^*$ , we have  $u_i^* = \gamma_i z^* + (1 - \gamma_i) R_i^*$  by the definition of  $u_i^*$  and plugging this value of  $u_i^*$  into the last inequality yields  $\hat{V}_i(\hat{R}_i - z^*) - (1 - \gamma_i) \hat{V}_i(R_i^* - z^*) \geq \gamma_i V_i^*(R_i^* - z^*)$ . Arranging the terms in the last inequality, we obtain

$$\hat{R}_i - z^* \geq \left[ \gamma_i \frac{V_i^*}{\hat{V}_i} + (1 - \gamma_i) \right] (R_i^* - z^*). \quad (4.8)$$

Noting that the dissimilarity parameter for nest  $i$  satisfies  $\gamma_i \in [0, 1]$ , the function  $x^{\gamma_i}$  is concave in  $x$  and its derivative at point 1 is  $\gamma_i$ . Therefore, the subgradient inequality at point 1 yields  $x^{\gamma_i} \leq 1 + \gamma_i(x - 1) = \gamma_i x + (1 - \gamma_i)$  for all  $x \in \mathfrak{R}_+$ . Using the subgradient inequality with  $x = V_i^*/\hat{V}_i$ , it follows that  $(V_i^*/\hat{V}_i)^{\gamma_i} \leq \gamma_i (V_i^*/\hat{V}_i) + 1 - \gamma_i$ . In this case, since  $R_i^* \geq z^*$ , (4.8) implies that  $\hat{R}_i - z^* \geq (V_i^*/\hat{V}_i)^{\gamma_i} (R_i^* - z^*)$  and arranging the terms in this inequality yields  $\hat{V}_i^{\gamma_i} (\hat{R}_i - z^*) \geq (V_i^*)^{\gamma_i} (R_i^* - z^*)$ . Therefore, the claim holds for each nest  $i$  that satisfies  $R_i^* \geq z^*$ .

We consider a nest  $i$  that satisfies  $R_i^* < z^*$ . Since  $\theta^q$  is the largest possible price for a product, the optimal expected revenue in problem (4.3) does not exceed  $\theta^q$  and we obtain  $z^* \leq \theta^q$ . We define the solution  $\tilde{p}_i = (\tilde{p}_{i1}, \dots, \tilde{p}_{in})$  to problem (4.7) as  $\tilde{p}_{ij} = \theta^q$  for all  $j \in N$ , which is feasible to this problem. Furthermore, (4.1) implies that  $R_i(\tilde{p}_i) = \sum_{j \in N} \theta^q v_{ij}(\tilde{p}_{ij}) / \sum_{j \in N} v_{ij}(\tilde{p}_{ij}) = \theta^q$ . Since  $R_i^* < z^*$ , we have  $u_i^* = z^*$  by the definition of  $u_i^*$  and we obtain  $\hat{V}_i(\hat{R}_i - z^*) = \hat{V}_i(\hat{R}_i - u_i^*) \geq V_i(\tilde{p}_i)(R_i(\tilde{p}_i) - u_i^*) = V_i(\tilde{p}_i)(R_i(\tilde{p}_i) - z^*) \geq 0 > V_i^*(R_i^* - z^*)$ , where the first inequality uses the fact that  $\tilde{p}_i$  is a feasible but not necessarily an optimal solution to problem (4.7), the second inequality uses the fact that  $R_i(\tilde{p}_i) = \theta^q \geq z^*$  and the third inequality uses the fact that  $R_i^* < z^*$ . Thus, we have  $\hat{V}_i(\hat{R}_i - z^*) \geq 0 > V_i^*(R_i^* - z^*)$ , which implies that  $\hat{V}_i^{\gamma_i} (\hat{R}_i - z^*) \geq 0 > (V_i^*)^{\gamma_i} (R_i^* - z^*)$ , establishing the claim for each nest  $i$  that satisfies  $R_i^* < z^*$ .

The discussion in the previous two paragraphs shows that our claim holds and we obtain  $\hat{V}_i^{\gamma_i}(\hat{R}_i - z^*) \geq (V_i^*)^{\gamma_i}(R_i^* - z^*)$  for all  $i \in M$ . In this case, the solution  $(\hat{p}_1, \dots, \hat{p}_m)$  satisfies the assumption of Lemma 4.3.2 and it follows from this lemma that  $\Pi(\hat{p}_1, \dots, \hat{p}_m) \geq z^*$ . On the other hand, noting that  $\hat{p}_i$  is a feasible solution to problem (4.7), we have  $\hat{p}_i \in \mathcal{F}_i$  for all  $i \in M$ , which indicates that  $(\hat{p}_1, \dots, \hat{p}_m)$  is a feasible solution to problem (4.3). The solution  $(\hat{p}_1, \dots, \hat{p}_m)$  is feasible to problem (4.3) and it provides an objective value for problem (4.3) that is at least as large as the optimal objective value of this problem. Therefore, we conclude that  $(\hat{p}_1, \dots, \hat{p}_m)$  is an optimal solution to problem (4.3), as desired.  $\square$

By Theorem 4.3.3, we can recover an optimal solution to problem (4.3) by solving problem (4.7) for all  $i \in M$ . Thus, if we let  $\hat{p}_i$  be an optimal solution to problem (4.7) and use the singleton  $\mathcal{P}_i = \{\hat{p}_i\}$  as the collection of candidate price vectors to charge in nest  $i$ , then we can stitch together an optimal solution to problem (4.3) by picking one price vector from each one of the collections  $\mathcal{P}_1, \dots, \mathcal{P}_m$ . However, this approach is not immediately useful for constructing a collection of candidate price vectors, since solving problem (4.7) requires the knowledge of  $u_i^*$ , which, in turn, requires the knowledge of an optimal solution to problem (4.3). To get around this difficulty, as a function of  $u_i \in \mathfrak{R}_+$ , we use  $\hat{p}_i(u_i)$  to denote an optimal solution to the problem

$$\max_{p_i \in \mathcal{F}_i} \left\{ V_i(p_i) (R_i(p_i) - u_i) \right\}. \quad (4.9)$$

In this case, we observe that if we use the collection of price vectors  $\mathcal{P}_i = \{\hat{p}_i(u_i) : u_i \in \mathfrak{R}_+\}$  as the collection of candidate price vectors for nest  $i$ , then we can stitch together an optimal solution to problem (4.3) by picking one price vector from each one of the collections  $\mathcal{P}_1, \dots, \mathcal{P}_m$ . To see this result, letting  $u_i^*$  be as defined in Theorem 4.3.3, we note that  $\hat{p}_i(u_i^*) \in \{\hat{p}_i(u_i) : u_i \in \mathfrak{R}_+\}$  for all  $i \in M$ . Furthermore, since problem (4.9) with  $u_i = u_i^*$  is identical to problem (4.7), by Theorem 4.3.3,

the solution  $(\hat{p}_1(u_1^*), \dots, \hat{p}_m(u_m^*))$  is optimal to problem (4.3). Therefore, for each nest  $i$ , the solution  $(\hat{p}_1(u_1^*), \dots, \hat{p}_m(u_m^*))$  uses one price vector from the collection of candidate price vectors  $\mathcal{P}_i = \{\hat{p}_i(u_i) : u_i \in \mathfrak{R}_+\}$  and this solution is optimal to problem (4.3).

We propose using  $\{\hat{p}_i(u_i) : u_i \in \mathfrak{R}_+\}$  as the collection of candidate price vectors for nest  $i$ , which is the collection of optimal solutions to problem (4.9) for any value of  $u_i \in \mathfrak{R}_+$ . In the subsequent sections, we show that the collection  $\{\hat{p}_i(u_i) : u_i \in \mathfrak{R}_+\}$  includes a reasonably small number of price vectors and we can find these price vectors in a tractable fashion.

#### 4.3.4 Counting Candidate Price Vectors

In this section, we show that there exists a collection of price vectors  $\mathcal{P}_i = \{p_i^t : t \in \mathcal{T}_i\}$  such that this collection includes an optimal solution to problem (4.9) for any value of  $u_i \in \mathfrak{R}_+$  and there are at most  $nq$  price vectors in this collection, where  $n$  is the number of products in a nest and  $q$  is the number of possible price levels. The fact that the objective function of problem (4.9) has a simple form plays an especially important role in this result. In particular, using the definitions of  $V_i(p_i)$  and  $R_i(p_i)$ , we can write problem (4.9) equivalently as

$$\max_{p_i \in \mathcal{F}_i} \left\{ \sum_{j \in N} v_{ij}(p_{ij}) \left[ \frac{\sum_{j \in N} p_{ij} v_{ij}(p_{ij})}{\sum_{j \in N} v_{ij}(p_{ij})} - u_i \right] \right\} = \max_{p_i \in \mathcal{F}_i} \left\{ \sum_{j \in N} (p_{ij} - u_i) v_{ij}(p_{ij}) \right\}. \quad (4.10)$$

In the next lemma, we begin by showing that as the value of  $u_i$  in problem (4.10) gets larger, then the optimal price for each product also gets larger.

**Lemma 4.3.4.** *Using  $\hat{p}_i(u_i) = (\hat{p}_{i1}(u_i), \dots, \hat{p}_{in}(u_i))$  to denote an optimal solution to problem (4.10) as a function of  $u_i$ , if we have  $u_i^- < u_i^+$ , then it holds that  $\hat{p}_{ij}(u_i^-) \leq \hat{p}_{ij}(u_i^+)$  for all  $j \in N$ .*

*Proof.* To get a contradiction, assume that  $u_i^- < u_i^+$  but we have  $\hat{p}_{ij}(u_i^-) > \hat{p}_{ij}(u_i^+)$  for some  $j \in N$ . For notational brevity, we let  $\hat{p}_i^- = \hat{p}_i(u_i^-)$  and  $\hat{p}_i^+ = \hat{p}_i(u_i^+)$ . Since the solutions  $\hat{p}_i^-$  and  $\hat{p}_i^+$  are optimal to problem (4.10) when this problem is solved at particular values of  $u_i$ , we have  $\hat{p}_i^- \in \mathcal{F}_i$  and  $\hat{p}_i^+ \in \mathcal{F}_i$ , which is to say that  $\hat{p}_{i1}^- \leq \hat{p}_{i2}^- \leq \dots \leq \hat{p}_{in}^-$  and  $\hat{p}_{i1}^+ \leq \hat{p}_{i2}^+ \leq \dots \leq \hat{p}_{in}^+$ . We let  $J = \{j \in N : \hat{p}_{ij}^- > \hat{p}_{ij}^+\}$ , which is nonempty by the assumption that  $\hat{p}_{ij}^- > \hat{p}_{ij}^+$  for some  $j \in N$ .

We define the solution  $\tilde{p}_i = (\tilde{p}_{i1}, \dots, \tilde{p}_{in})$  to problem (4.10) as  $\tilde{p}_{ij} = \hat{p}_{ij}^- \vee \hat{p}_{ij}^+$  for all  $j \in N$ , where we use  $a \vee b = \max\{a, b\}$ . If  $f(j)$  and  $g(j)$  are both increasing functions of  $j \in N$ , then  $f(j) \vee g(j)$  is also an increasing function of  $j \in N$ . By the discussion at the end of the previous paragraph,  $\hat{p}_{ij}^-$  and  $\hat{p}_{ij}^+$  are both increasing functions of  $j \in N$ . Thus,  $\tilde{p}_{ij} = \hat{p}_{ij}^- \vee \hat{p}_{ij}^+$  is also an increasing function of  $j \in N$ , which implies that  $\tilde{p}_{i1} \leq \tilde{p}_{i2} \leq \dots \leq \tilde{p}_{in}$ . Therefore, we have  $\tilde{p}_i \in \mathcal{F}_i$ , indicating that  $\tilde{p}_i$  is a feasible solution to problem (4.10). In this case, since  $\hat{p}_i^+$  is an optimal solution to problem (4.10) when we solve this problem with  $u_i = u_i^+$ , we have  $\sum_{j \in N} (\hat{p}_{ij}^+ - u_i^+) v_{ij}(\hat{p}_{ij}^+) \geq \sum_{j \in N} (\tilde{p}_{ij} - u_i^+) v_{ij}(\tilde{p}_{ij})$ . By the definitions of  $J$  and  $\tilde{p}_i$ , we have  $\tilde{p}_{ij} = \hat{p}_{ij}^-$  for all  $j \in J$  and  $\tilde{p}_{ij} = \hat{p}_{ij}^+$  for all  $j \notin J$ . Thus, the last inequality can be written as  $\sum_{j \in N} (\hat{p}_{ij}^+ - u_i^+) v_{ij}(\hat{p}_{ij}^+) \geq \sum_{j \in J} (\hat{p}_{ij}^- - u_i^+) v_{ij}(\hat{p}_{ij}^-) + \sum_{j \notin J} (\hat{p}_{ij}^+ - u_i^+) v_{ij}(\hat{p}_{ij}^+)$ , in which case, canceling the common terms on the two sides of the inequality, we have  $\sum_{j \in J} (\hat{p}_{ij}^+ - u_i^+) v_{ij}(\hat{p}_{ij}^+) \geq \sum_{j \in J} (\hat{p}_{ij}^- - u_i^+) v_{ij}(\hat{p}_{ij}^-)$ .

We define the solution  $\bar{p}_i = (\bar{p}_{i1}, \dots, \bar{p}_{in})$  to problem (4.10) as  $\bar{p}_{ij} = \hat{p}_{ij}^+ \wedge \hat{p}_{ij}^-$  for all  $j \in N$ , where we use  $a \wedge b = \min\{a, b\}$ . We note that if  $f(j)$  and  $g(j)$  are both increasing functions of  $j \in N$ , then  $f(j) \wedge g(j)$  is also an increasing function of  $j \in N$ . In this case, using the same approach in the previous paragraph, we can show that  $\bar{p}_i \in \mathcal{F}_i$ . Thus, since  $\hat{p}_i^-$  is an optimal solution to problem (4.10) when we solve this problem with  $u_i = u_i^-$ , we have  $\sum_{j \in N} (\hat{p}_{ij}^- - u_i^-) v_{ij}(\hat{p}_{ij}^-) \geq \sum_{j \in N} (\bar{p}_{ij} - u_i^-) v_{ij}(\bar{p}_{ij})$ . Noting the definitions of  $J$  and  $\bar{p}_i$ , the last inequality

can equivalently be written as  $\sum_{j \in N} (\hat{p}_{ij}^- - u_i^-) v_{ij}(\hat{p}_{ij}^-) \geq \sum_{j \in J} (\hat{p}_{ij}^+ - u_i^-) v_{ij}(\hat{p}_{ij}^+) + \sum_{j \notin J} (\hat{p}_{ij}^- - u_i^-) v_{ij}(\hat{p}_{ij}^-)$ , in which case, canceling the common terms on the two sides of the inequality yields  $\sum_{j \in J} (\hat{p}_{ij}^- - u_i^-) v_{ij}(\hat{p}_{ij}^-) \geq \sum_{j \in J} (\hat{p}_{ij}^+ - u_i^-) v_{ij}(\hat{p}_{ij}^+)$ . From the previous paragraph, we also have  $\sum_{j \in J} (\hat{p}_{ij}^+ - u_i^+) v_{ij}(\hat{p}_{ij}^+) \geq \sum_{j \in J} (\hat{p}_{ij}^- - u_i^+) v_{ij}(\hat{p}_{ij}^-)$ . Adding the last two inequalities and canceling the common terms yield  $u_i^- \sum_{j \in J} (v_{ij}(\hat{p}_{ij}^+) - v_{ij}(\hat{p}_{ij}^-)) \geq u_i^+ \sum_{j \in J} (v_{ij}(\hat{p}_{ij}^+) - v_{ij}(\hat{p}_{ij}^-))$ .

By the definition of  $J$ , we have  $\hat{p}_{ij}^- > \hat{p}_{ij}^+$  for all  $j \in J$ . Noting that the preference weight of a product gets larger as we charge a smaller price for the product, we have  $v_{ij}(\hat{p}_{ij}^-) < v_{ij}(\hat{p}_{ij}^+)$  for all  $j \in J$ . Thus, we have  $\sum_{j \in J} (v_{ij}(\hat{p}_{ij}^+) - v_{ij}(\hat{p}_{ij}^-)) > 0$ , in which case, by the inequality at the end of the previous paragraph, we obtain  $u_i^- \geq u_i^+$ , which is a contradiction.  $\square$

We observe that the last step in the proof of Lemma 4.3.4 critically depends on the assumption that  $v_{ij}(p_{ij})$  is a decreasing function of  $p_{ij}$ . Also, it is worthwhile to note that Lemma 4.3.4 holds even when there are multiple optimal solutions to problem (4.10) and we choose  $\hat{p}_i(u_i)$  as any one of these solutions. In the next theorem, we use Lemma 4.3.4 to show that there exists a collection of at most  $nq$  price vectors such that this collection includes an optimal solution to problem (4.10) for any value of  $u_i \in \mathfrak{R}_+$ . The intuition behind this result is that if we increase the value of  $u_i$  in problem (4.10), then Lemma 4.3.4 implies that the price of a product in an optimal solution either does not change or becomes larger. Since there are  $q$  possible prices for a product, the price of a product will no longer change after a small number of price changes.

**Theorem 4.3.5.** *There exists a collection of at most  $nq$  price vectors such that this collection includes an optimal solution to problem (4.10) for any value of  $u_i \in \mathfrak{R}_+$ .*

*Proof.* Assume that there are  $K$  distinct values of  $u_i \in \mathfrak{R}_+$  such that if we solve problem (4.10) with each one of these values, then we obtain a distinct optimal

solution. We use  $\{\hat{u}_i^k : k = 1, \dots, K\}$  to denote these values of  $u_i \in \mathfrak{R}_+$  and use  $\hat{p}_i^k$  to denote an optimal solution to problem (4.10) when we solve this problem with  $u_i = \hat{u}_i^k$ . By our assumption, none of price vectors in  $\{\hat{p}_i^k : k = 1, \dots, K\}$  are equal to each other. To get a contradiction, assume that  $K > nq$ . Without loss of generality, we index the values  $\{\hat{u}_i^k : k = 1, \dots, K\}$  such that  $\hat{u}_i^1 < \hat{u}_i^2 < \dots < \hat{u}_i^K$ , in which case, Lemma 4.3.4 implies that  $\hat{p}_{ij}^1 \leq \hat{p}_{ij}^2 \leq \dots \leq \hat{p}_{ij}^K$  for all  $j \in N$ . Since the price vectors  $\{\hat{p}_i^k : k = 1, \dots, K\}$  are distinct, using  $\mathbf{1}(\cdot)$  to denote the indicator function, we have  $\sum_{j \in N} \mathbf{1}(\hat{p}_{ij}^k < \hat{p}_{ij}^{k+1}) > 1$ , indicating that there is at least one different price in the price vectors  $\hat{p}_i^k$  and  $\hat{p}_i^{k+1}$ . Adding the last inequality over all  $k = 1, \dots, K - 1$  and noting that  $K > nq$ , we obtain  $\sum_{j \in N} \sum_{k=1}^{K-1} \mathbf{1}(\hat{p}_{ij}^k < \hat{p}_{ij}^{k+1}) > K - 1 \geq nq$ . Focusing on the first and last terms in the last chain of inequalities, since  $|N| = n$ , it must be the case that  $\sum_{k=1}^{K-1} \mathbf{1}(\hat{p}_{ij}^k < \hat{p}_{ij}^{k+1}) > q$  for some  $j \in N$ , which implies that more than  $q$  of the inequalities  $\hat{p}_{ij}^1 \leq \hat{p}_{ij}^2 \leq \dots \leq \hat{p}_{ij}^K$  are strict, but since there are  $q$  possible values for the price of a product, more than  $q$  of these inequalities cannot be strict and we obtain a contradiction.  $\square$

Thus, there exists a reasonably small collection of price vectors that includes an optimal solution to problem (4.10) for any  $u_i \in \mathfrak{R}_+$ . In the next section, we show how to construct this collection.

### 4.3.5 Constructing Candidate Price Vectors

In the previous section, we show that there exists a collection of price vectors with at most  $nq$  price vectors in it such that this collection includes an optimal solution to problem (4.10) for any value of  $u_i \in \mathfrak{R}_+$ . In this section, we show how to come up with this collection in a tractable fashion. In problem (4.10), if we charge the price  $p_{ij}$  for product  $j$  in nest  $i$ , then we obtain a contribution of  $(p_{ij} - u_i) v_{ij}(p_{ij})$ . By the constraint  $p_i \in \mathcal{F}_i$ , the price charged for product  $j$  should be at least as



large as the price charged for product  $j - 1$ . Problem (4.10) finds the prices to charge for the products in nest  $i$  to maximize the total contribution. So, we can solve problem (4.10) by using a dynamic program. The decision epochs are the products in nest  $i$ . When making the decision for product  $j$  in nest  $i$ , the state variable is the price for product  $j - 1$ . Thus, for a fixed value of  $u_i \in \mathfrak{R}_+$ , we can solve problem (4.10) by using the dynamic program

$$\Phi_{ij}(p_{i,j-1} | u_i) = \max_{\substack{p_{ij} \in \Theta : \\ p_{ij} \geq p_{i,j-1}}} \left\{ (p_{ij} - u_i) v_{ij}(p_{ij}) + \Phi_{i,j+1}(p_{ij} | u_i) \right\}, \quad (4.11)$$

with the boundary condition that  $\Phi_{i,n+1}(\cdot | u_i) = 0$ . The optimal objective value of problem (4.10) is given by  $\Phi_{i1}(\theta^1 | u_i)$ , where the value functions  $\{\Phi_{ij}(p_{i,j-1} | u_i) : p_{i,j-1} \in \Theta, j \in N\}$  are obtained through the dynamic program in (4.11). By Theorem 4.3.5, there are most  $nq$  solutions from the dynamic program in (4.11) such that the solution from this dynamic program for any value of  $u_i \in \mathfrak{R}_+$  is one of these  $nq$  solutions. The question is how to come up with these solutions.

To answer this question, we use the linear programming formulation of the dynamic program in (4.11). Dynamic programs with finite state and action spaces have equivalent linear programming formulations; see [46]. In these linear programs, there is one decision variable for each state and decision epoch and there is one constraint for each state, action and decision epoch. The linear program corresponding to the dynamic program in (4.11) is given by

$$\begin{aligned} \min \quad & \phi_{i1}(\theta^1) \\ \text{s.t.} \quad & \phi_{ij}(p_{i,j-1}) \geq (p_{ij} - u_i) v_{ij}(p_{ij}) + \phi_{i,j+1}(p_{ij}) \quad \forall p_{i,j-1} \in \Theta, p_{ij} \in \mathcal{L}(p_{i,j-1}), \end{aligned} \quad (4.12)$$

where the decision variables are  $\{\phi_{ij}(p_{i,j-1}) : p_{i,j-1} \in \Theta, j \in N\}$  and we follow the convention that  $\phi_{i,n+1}(p_{in}) = 0$  for all  $p_{in} \in \Theta$ . The set  $\mathcal{L}(p_{i,j-1})$  is the set of feasible prices for product  $j$  given that the price for product  $j - 1$  is  $p_{i,j-1}$ , which

is given by  $\mathcal{L}(p_{i,j-1}) = \{p_{ij} \in \Theta : p_{ij} \geq p_{i,j-1}\}$ . If we solve the linear program in (4.12), then the optimal value of the decision variable  $\phi_{i1}(\theta^1)$  is equal to  $\Phi_{i1}(\theta^1 | u_i)$  obtained through the dynamic program in (4.11), which is, in turn, equal to the optimal objective value of problem (4.10). The critical observation is that the value of  $u_i \in \mathfrak{R}_+$  only affects the right hand side coefficients of the constraints in problem (4.12). Therefore, we can vary  $u_i \in \mathfrak{R}_+$  parametrically and solve problem (4.12) by using the parametric simplex method to generate the possible optimal solutions to this problem for all values of  $u_i \in \mathfrak{R}_+$ . These solutions provide the solutions to the dynamic program in (4.11) for all values of  $u_i \in \mathfrak{R}_+$ .

Since there are  $q$  possible prices for a product and there are  $n$  products in a nest, the linear program in (4.12) has  $O(nq)$  decision variables and  $O(nq^2)$  constraints. Putting all of the discussion so far together, we solve problem (4.12) by using the parametric simplex method to generate the optimal solutions to this problem for all values of  $u_i \in \mathfrak{R}_+$ . These solutions correspond to the optimal solutions to problem (4.10) for all values of  $u_i \in \mathfrak{R}_+$ . By the discussion that follows Theorem 4.3.3, we can use the optimal solutions to problem (4.10) for all values of  $u_i \in \mathfrak{R}_+$  as the collection of candidate price vectors  $\mathcal{P}_i$  for nest  $i$ . Once we have the collection of candidate price vectors for each nest, we can solve the linear program in (4.6) to obtain the value of  $\hat{z}$  that satisfies  $v_0 \hat{z} = f(\hat{z})$ . Since there are at most  $nq$  price vectors in each one of the collections  $\mathcal{P}_1, \dots, \mathcal{P}_m$ , there are  $O(m)$  decision variables and  $O(mnq)$  constraints in the linear program in (4.6). In this case, by Theorem 4.3.1, we can solve problem (4.5) for all  $i \in M$  to obtain an optimal solution to problem (4.3).

In Section 4, we provide numerical experiments and demonstrate that the approach described above can obtain an optimal solution to problem (4.3) quite fast. For problem instances with six nests, 30 products in each nest and 30 possible

prices for each product, we can obtain an optimal solution to problem (4.3) in no more than two seconds.

## 4.4 Price Ladders Between Nests

In this section, we consider the case with price ladders between nests. In this setting, there is an intrinsic ordering between the qualities of the nests and the prices charged in a nest corresponding to a higher quality level should also be larger. There is no intrinsic ordering between the qualities or the prices of the products in the same nest.

### 4.4.1 Problem Formulation

Our problem formulation is similar to the one in Section 4.3.1. There are  $m$  nests indexed by  $M = \{1, \dots, m\}$ . In each nest, there are  $n$  products indexed by  $N = \{1, \dots, n\}$ . For each product, there are  $q$  possible prices given by  $\Theta = \{\theta^1, \dots, \theta^q\}$ . The possible prices for a product are indexed such that  $0 < \theta^1 < \theta^2 < \dots < \theta^q$ . We use  $p_{ij} \in \Theta$  to denote the price that we charge for product  $j$  in nest  $i$ . If we charge the price  $p_{ij}$  for product  $j$  in nest  $i$ , then the preference weight of this product is given by  $v_{ij}(p_{ij})$ . If we charge a larger price for a product, then its preference weight becomes smaller, implying that  $v_{ij}(\theta^1) > v_{ij}(\theta^2) > \dots > v_{ij}(\theta^q) > 0$ . Customers follow the same choice process described in Section 4.3.1. Thus, if we use  $p_i = (p_{i1}, \dots, p_{in}) \in \Theta^n$  to denote the price vector charged in nest  $i$ , then the expected revenue obtained from a customer that has already decided to make a purchase in nest  $i$  is given by  $R_i(p_i)$ , where  $R_i(p_i)$  is as in (4.1). If we charge the price vectors  $(p_1, \dots, p_m) \in \Theta^{m \times n}$  over all nests, then the expected revenue obtained from a customer is given by  $\Pi(p_1, \dots, p_m)$ , where  $\Pi(p_1, \dots, p_m)$  is as in

(4.2).

Our goal is to find the price vectors  $(p_1, \dots, p_m)$  to maximize the expected revenue  $\Pi(p_1, \dots, p_m)$  subject to the constraint that the price vectors charged in the different nests are consistent with the quality level that each nest represents. In other words, if nest  $i$  corresponds to a higher quality level than nest  $l$ , then the prices of the products in nest  $i$  should be larger than the prices of the products in nest  $l$ . This constraint can be interpreted as a price ladder constraint between the nests. To formulate the price ladder constraint, without loss of generality, we index the nests such that a nests with a larger index represents a higher quality level. In other words, the nests  $M = \{1, \dots, m\}$  are indexed in the order of increasing quality levels. Thus, the price ladder constraint ensures that the price vectors  $(p_1, \dots, p_m)$  charged over all nests satisfy  $\max_{j \in N} p_{1j} \leq \min_{j \in N} p_{2j}$ ,  $\max_{j \in N} p_{2j} \leq \min_{j \in N} p_{3j}$ ,  $\dots$ ,  $\max_{j \in N} p_{m-1,j} \leq \min_{j \in N} p_{mj}$ . As a function of the price vector  $p_{i-1}$  charged in nest  $i - 1$ , the set of feasible price vectors in nest  $i$  is  $\mathcal{G}_i(p_{i-1}) = \{p_i \in \Theta^n : \min_{j \in N} p_{ij} \geq \max_{j \in N} p_{i-1,j}\}$ . We want to find the price vectors to charge over all nests to maximize the expected revenue from a customer, yielding the problem

$$z^* = \max_{\substack{(p_1, \dots, p_m) \in \Theta^{m \times n} : \\ p_i \in \mathcal{G}_i(p_{i-1}) \forall i \in M \setminus \{1\}}} \left\{ \Pi(p_1, \dots, p_m) \right\}. \quad (4.13)$$

Problem (4.13) is significantly more difficult than problem (4.3) since the constraints link the price vectors charged in different nests. The broad outline of our approach for problem (4.13) is similar to the one for problem (4.3). We relate problem (4.13) to the problem of computing the fixed point of a function. Assuming that we have a collection of candidate price vectors for each nest such that we can stitch together an optimal solution to problem (4.13) by picking one price vector from each one of the collections, we show how to obtain an optimal solution

to problem (4.13). Finally, we show how to come up with the collections of candidate price vectors. Although the broad outline of our approach for problem (4.13) is similar to the one for problem (4.3), the details are quite different as problem (4.13) is significantly more difficult than problem (4.3).

#### 4.4.2 Connection to a Fixed Point Representation

Assume that we have a collection of candidate price vectors  $\mathcal{P}_i = \{p_i^t : t \in \mathcal{T}_i\}$  for each nest  $i$  such that we can stitch together an optimal solution to problem (4.13) by picking one price vector from each one of the collections  $\mathcal{P}_1, \dots, \mathcal{P}_m$ . In other words, there exists an optimal solution  $(p_1^*, \dots, p_m^*)$  to problem (4.13) such that  $p_i^* \in \mathcal{P}_i$  for all  $i \in M$ . The question is how we can pick a price vector  $p_i^*$  from the collection  $\mathcal{P}_i$  for each nest  $i$  such that the solution  $(p_1^*, \dots, p_m^*)$  is indeed optimal to problem (4.13). To answer this question, for any  $z \in \mathfrak{R}_+$ , we define  $g(z)$  as

$$g(z) = \max_{\substack{(p_1, \dots, p_m) \in \mathcal{P}_1 \times \dots \times \mathcal{P}_m : \\ p_i \in \mathcal{G}_i(p_{i-1}) \forall i \in M \setminus \{1\}}} \left\{ \sum_{i \in M} V_i(p_i)^{\gamma_i} (R_i(p_i) - z) \right\}. \quad (4.14)$$

Since  $v_0 z$  is increasing and  $g(z)$  is decreasing in  $z$  with  $g(0) \geq 0$ , there exists a value of  $\hat{z}$  that satisfies  $v_0 \hat{z} = g(\hat{z})$ , which corresponds to the fixed point of the function  $g(\cdot)/v_0$ . In the next theorem, we show that the value of  $\hat{z}$  that satisfies  $v_0 \hat{z} = g(\hat{z})$  can be used to construct an optimal solution to problem (4.13). The proof of this theorem follows from an outline that is similar to the proof of Theorem 4.3.1 and we omit the proof. In the theorem, we recall that  $z^*$  corresponds to the optimal objective value of problem (4.13).

**Theorem 4.4.1.** *Assume that we have a collection of candidate price vectors  $\mathcal{P}_i$  for each nest  $i$  such that we can stitch together an optimal solution to problem (4.13) by picking one price vector from each one of the collections  $\mathcal{P}_1, \dots, \mathcal{P}_m$ . Let*

$\hat{z}$  be such that  $v_0 \hat{z} = g(\hat{z})$  and  $(\hat{p}_1, \dots, \hat{p}_m)$  be an optimal solution to problem (4.14) when we solve this problem with  $z = \hat{z}$ . Then, we have  $\Pi(\hat{p}_1, \dots, \hat{p}_m) \geq z^*$ .

Building on Theorem 4.4.1, we can obtain an optimal solution to problem (4.13) as follows. Assume that we have a collection of candidate price vectors  $\mathcal{P}_i = \{p_i^t : t \in \mathcal{T}_i\}$  for each nest  $i$  such that we can stitch together an optimal solution to problem (4.13) by picking one price vector from each one of the collections  $\mathcal{P}_1, \dots, \mathcal{P}_m$ . We find the value of  $\hat{z}$  that satisfies  $v_0 \hat{z} = g(\hat{z})$ . If we let  $(\hat{p}_1, \dots, \hat{p}_m)$  be an optimal solution to problem (4.14) when we solve this problem with  $z = \hat{z}$ , then by Theorem 4.4.1, we have  $\Pi(\hat{p}_1, \dots, \hat{p}_m) \geq z^*$ . Since  $(\hat{p}_1, \dots, \hat{p}_m)$  is a feasible solution to problem (4.14), we also have  $\hat{p}_i \in \mathcal{G}_i(p_{i-1})$  for all  $i \in M \setminus \{1\}$ . Thus, the solution  $(\hat{p}_1, \dots, \hat{p}_m)$  is feasible to problem (4.13) and provides an objective value for problem (4.13) that is at least as large as the optimal objective value of this problem, indicating that  $(\hat{p}_1, \dots, \hat{p}_m)$  is an optimal solution to problem (4.13).

One important question is how we can find the value of  $\hat{z}$  that satisfies  $v_0 \hat{z} = g(\hat{z})$ . In problem (4.14), we observe that if we charge the price vector  $p_i$  in nest  $i$ , then we obtain a contribution of  $V_i(p_i)^{\gamma_i} (R_i(p_i) - z)$ . By the constraints  $\hat{p}_i \in \mathcal{G}_i(p_{i-1})$  for all  $i \in M \setminus \{1\}$ , the smallest price charged in nest  $i$  should be at least as large as the largest price charged in nest  $i - 1$ . Problem (4.14) finds the price vectors to charge for the nests to maximize the total contribution. So, for a fixed value of  $z \in \mathfrak{R}_+$ , we can solve problem (4.14) by using a dynamic program. The decision epochs are the nests. When making the decision for nest  $i$ , the state variable is the largest price charged for the products in nest  $i - 1$ . Thus, for a fixed value of  $z \in \mathfrak{R}_+$ , we can obtain an optimal solution to problem (4.14) by solving

the dynamic program

$$\Psi_i(w_{i-1} | z) = \max_{\substack{p_i \in \mathcal{P}_i : \\ p_{ij} \geq w_{i-1} \forall j \in N}} \left\{ V_i(p_i)^{\gamma_i} (R_i(p_i) - z) + \Psi_{i+1}(\max_{j \in N} p_{ij} | z) \right\}, \quad (4.15)$$

with the boundary condition that  $\Psi_{m+1}(\cdot | z) = 0$ . The optimal objective value of problem (4.14) is given by  $\Psi_1(\theta^1 | z)$ , where the value functions  $\{\Psi_i(w_{i-1} | z) : w_{i-1} \in \Theta, i \in M\}$  are obtained through the dynamic program in (4.15). Since there are  $q$  possible prices for a product, we can solve the dynamic program above in  $O(q \sum_{i \in M} |\mathcal{P}_i|)$  operations, which is reasonable as long as the numbers of price vectors in the collections  $\mathcal{P}_1, \dots, \mathcal{P}_m$  are not too large. In Section 4.3.2, we recall that we find the value of  $\hat{z}$  satisfying  $v_0 \hat{z} = f(\hat{z})$  by using the linear program in (4.6), but this linear program is not useful to find the value of  $\hat{z}$  satisfying  $v_0 \hat{z} = g(\hat{z})$  since problem (4.14) does not decompose by the nests due to the constraints  $p_i \in \mathcal{G}_i(p_{i-1})$  for all  $i \in M \setminus \{1\}$ . Instead, we show how to use the linear programming formulation of the dynamic program in (4.15) to find the value of  $\hat{z}$  that satisfies  $v_0 \hat{z} = g(\hat{z})$ .

As mentioned in Section 4.3.5, dynamic programs with finite state and action spaces have equivalent linear programming formulations. In these linear programs, there is one decision variable for each state and decision epoch and there is one constraint for each state, action and decision epoch. Therefore, building on the linear programming formulation corresponding to the dynamic program in (4.15), we propose solving the linear program

$$\begin{aligned} \min \quad & \psi_1(\theta^1) & (4.16) \\ \text{s.t.} \quad & \psi_i(w_{i-1}) \geq V_i(p_i)^{\gamma_i} (R_i(p_i) - z) + \psi_{i+1}(\max_{j \in N} p_{ij}) \quad \forall w_{i-1} \in \Theta, p_i \in \mathcal{M}_i(w_{i-1}) \\ & v_0 z = \psi_1(\theta^1) \end{aligned}$$

to find the value of  $\hat{z}$  that satisfies  $v_0 \hat{z} = g(\hat{z})$ . In the linear program above, the decision variables are  $z$  and  $\psi = \{\psi_i(w_{i-1}) : w_{i-1} \in \Theta, i \in M\}$ . We follow the convention that  $\psi_{m+1}(w_m) = 0$  for all  $w_m \in \Theta$ . The set  $\mathcal{M}_i(w_{i-1})$  is the set of feasible price vectors in nest  $i$  given that the largest price charged in nest  $i - 1$  is  $w_{i-1}$ , which is given by  $\mathcal{M}_i(w_{i-1}) = \{p_i \in \mathcal{P}_i : p_{ij} \geq w_{i-1} \forall j \in N\}$ . If we drop the second constraint in problem (4.16) and solve this problem for a fixed value of  $z \in \mathfrak{R}_+$ , then this problem corresponds to the linear programming formulation for the dynamic program in (4.15). Therefore, the optimal value of the decision variable  $\psi_1(\theta^1)$  would correspond to  $\Psi_1(\theta^1 | z)$  obtained through the dynamic program in (4.15), which is equal to the optimal objective value of problem (4.14) for a fixed value of  $z$ . On the other hand, it turns out that if we solve problem (4.16) as formulated, then the optimal value of the decision variable  $z$  corresponds to the value of  $\hat{z}$  that satisfies  $v_0 \hat{z} = g(\hat{z})$ . We show this result in the next theorem.

**Theorem 4.4.2.** *Using  $(\hat{z}, \hat{\psi})$  to denote an optimal solution to problem (4.16), we have  $v_0 \hat{z} = g(\hat{z})$ .*

*Proof.* Let  $\tilde{z}$  satisfy  $v_0 \tilde{z} = g(\tilde{z})$ . We want to show that  $\tilde{z} = \hat{z}$ . We solve the dynamic program in (4.15) with  $z = \tilde{z}$  to obtain the value functions  $\Psi(\tilde{z}) = \{\Psi_i(w_{i-1} | \tilde{z}) : w_{i-1} \in \Theta, i \in M\}$ . Due to the way these value functions are computed in the dynamic program in (4.15), we have  $\Psi_i(w_{i-1} | \tilde{z}) \geq V_i(p_i)^{\gamma_i} (R_i(p_i) - \tilde{z}) + \Psi_{i+1}(\max_{j \in N} p_{ij} | \tilde{z})$  for all  $w_{i-1} \in \Theta, p_i \in \mathcal{M}(w_{i-1})$  and  $i \in M$ . Thus, the solution  $(\hat{z}, \Psi(\hat{z}))$  satisfies the first set of constraints in problem (4.16). By the discussion that follows the dynamic program in (4.15),  $\Psi_1(\theta^1 | \tilde{z})$  provides the optimal objective value of problem (4.14) when we solve this problem with  $z = \tilde{z}$ , yielding  $\Psi_1(\theta^1 | \tilde{z}) = g(\tilde{z}) = v_0 \tilde{z}$ . Thus, the solution  $(\hat{z}, \Psi(\hat{z}))$  satisfies the second constraint in problem (4.16) as well. Since the solution  $(\hat{z}, \Psi(\hat{z}))$  is feasible to problem (4.16), the objective value provided by this solution is at least as large



as the optimal objective value, yielding  $\Psi_1(\theta^1 | \tilde{z}) \geq \hat{\psi}_1(\theta^1)$ . Thus, we obtain  $v_0 \tilde{z} = g(\tilde{z}) = \Psi_1(\theta^1 | \tilde{z}) \geq \hat{\psi}_1(\theta^1) = v_0 \hat{z}$ , where the last equality holds since  $(\hat{z}, \hat{\psi})$  is a feasible solution to problem (4.16).

The last chain of inequalities in the previous paragraph shows that  $\tilde{z} \geq \hat{z}$ . To show that  $\tilde{z} = \hat{z}$ , we solve problem (4.14) with  $z = \hat{z}$  to obtain an optimal solution  $(\hat{p}_1, \dots, \hat{p}_m)$ . Therefore, we have  $g(\hat{z}) = \sum_{i \in M} V_i(\hat{p}_i)^{\gamma_i} (R_i(\hat{p}_i) - \hat{z})$ . For all  $i \in M$ , we let  $\hat{w}_i = \max_{j \in N} \hat{p}_{ij}$  with the convention that  $\hat{w}_0 = \theta^1$ . Since the solution  $(\hat{p}_1, \dots, \hat{p}_m)$  is feasible to problem (4.14), we have  $\hat{p}_i \in \mathcal{G}_i(\hat{p}_{i-1})$  for all  $i \in M \setminus \{1\}$  and  $\hat{p}_i \in \mathcal{P}_i$  for all  $i \in M$ , which is equivalent to having  $\hat{p}_i \in \mathcal{M}(\hat{w}_{i-1})$  for all  $i \in M$ . In this case, using the fact that the solution  $(\hat{z}, \hat{\psi})$  is feasible to problem (4.16), we have  $\hat{\psi}_i(\hat{w}_{i-1}) \geq V_i(\hat{p}_i)^{\gamma_i} (R_i(\hat{p}_i) - \hat{z}) + \hat{\psi}_{i+1}(\hat{w}_i)$  for all  $i \in M$ . Adding these inequalities over all  $i \in M$  and noting that  $\hat{w}_0 = \theta^1$ , we obtain  $\hat{\psi}_1(\theta^1) \geq \sum_{i \in M} V_i(\hat{p}_i)^{\gamma_i} (R_i(\hat{p}_i) - \hat{z}) = g(\hat{z})$ , where the last equality uses the definition of  $(\hat{p}_1, \dots, \hat{p}_m)$ . This chain of inequalities shows that  $\hat{\psi}_1(\theta^1) \geq g(\hat{z})$ . As mentioned at the beginning of this paragraph, we have  $\tilde{z} \geq \hat{z}$ . Noting that  $g(z)$  is decreasing in  $z$ , we obtain  $g(\hat{z}) \geq g(\tilde{z})$ . In this case, we have  $\hat{\psi}_1(\theta^1) \geq g(\hat{z}) \geq g(\tilde{z}) = v_0 \tilde{z} \geq v_0 \hat{z} = \hat{\psi}_1(\theta^1)$ , where the first equality uses the definition of  $\tilde{z}$ , the third inequality is by the fact that  $\tilde{z} \geq \hat{z}$  and the second equality uses the fact that the solution  $(\hat{z}, \hat{\psi})$  is feasible to problem (4.16) so that it satisfies the second constraint in this problem. Thus, all of the inequalities in the last chain of inequalities hold as equality and we obtain  $g(\hat{z}) = g(\tilde{z}) = v_0 \tilde{z} = v_0 \hat{z}$ , establishing that  $\tilde{z} = \hat{z}$ .  $\square$

By Theorem 4.4.2, we can solve problem (4.16) to find the value of  $\hat{z}$  that satisfies  $v_0 \hat{z} = g(\hat{z})$ . Problem (4.16) is a linear program with  $O(mq)$  decision variables and  $\sum_{i \in M} O(q|\mathcal{P}_i|)$  constraints, which is tractable as long as the numbers of price vectors in the collections  $\mathcal{P}_1, \dots, \mathcal{P}_m$  are reasonably small. In the rest of our discussion, we focus on how to construct a reasonably small collection of candidate

price vectors  $\mathcal{P}_i$  for each nest  $i$  such that we can stitch together an optimal solution to problem (4.13) by picking one price vector from each one of the collections  $\mathcal{P}_1, \dots, \mathcal{P}_m$ . Once we have these collections, we can solve problem (4.16) to find  $\hat{z}$  satisfying  $v_0 \hat{z} = g(\hat{z})$  and we can use Theorem 4.4.1 to obtain an optimal solution to problem (4.13).

### 4.4.3 Characterizing Candidate Price Vectors

In this section, we provide an alternative characterization of the optimal price vector to charge in each nest. This characterization ultimately becomes useful when we construct a collection of candidate price vectors  $\mathcal{P}_i$  for each nest  $i$  such that we can stitch together an optimal solution to problem (4.13) by picking one price vector from each one of the collections  $\mathcal{P}_1, \dots, \mathcal{P}_m$ . In the next theorem, we provide our alternative characterization of the optimal price vector to charge in each nest. This theorem is analogous to Theorem 4.3.3, but its proof is substantially more involved due to the constraints in problem (4.13) that link the price vectors charged in different nests. We defer the proof to the appendix.

**Theorem 4.4.3.** *Let  $(p_1^*, \dots, p_m^*)$  be an optimal solution to problem (4.13) providing the objective value  $z^*$  and set  $u_i^* = \max\{\gamma_i z^* + (1 - \gamma_i) R_i(p_i^*), z^*\}$ ,  $\ell_i^* = \min_{j \in N} p_{ij}^*$  and  $w_i^* = \max_{j \in N} p_{ij}^*$ . If  $\hat{p}_i$  is an optimal solution to the problem*

$$\begin{aligned} \max_{p_i \in \Theta:} \quad & \left\{ V_i(p_i) (R_i(p_i) - u_i^*) \right\}, \quad (4.17) \\ & \ell_i^* \leq p_{ij} \leq w_i^* \quad \forall j \in N \end{aligned}$$

*then  $(\hat{p}_1, \dots, \hat{p}_m)$  is an optimal solution to problem (4.13).*

Theorem 4.4.3 implies that we can recover an optimal solution to problem (4.13) by solving problem (4.17) for all  $i \in M$ . Therefore, if we let  $\hat{p}_i$  be an optimal solution to problem (4.17) and use  $\mathcal{P}_i = \{\hat{p}_i\}$  as the collection of candidate

price vectors to charge in nest  $i$ , then we can stitch together an optimal solution to problem (4.13) by picking one price vector from each one of the collections  $\mathcal{P}_1, \dots, \mathcal{P}_m$ . However, this approach is not immediately useful for constructing a collection of candidate price vectors, since solving problem (4.17) requires the knowledge of  $u_i^*$ ,  $\ell_i^*$  and  $w_i^*$ , all of which, in turn, require the knowledge of an optimal solution to problem (4.13). To deal with this difficulty, as a function of  $u_i \in \mathfrak{R}_+$ ,  $\ell_i \in \Theta$  and  $w_i \in \Theta$ , we use  $\hat{p}_i(u_i, \ell_i, w_i)$  to denote an optimal solution to the problem

$$\begin{aligned} & \max_{p_i \in \Theta :} \left\{ V_i(p_i) (R_i(p_i) - u_i) \right\}. & (4.18) \\ & \ell_i \leq p_{ij} \leq w_i \quad \forall j \in N \end{aligned}$$

In this case, if we use the collection  $\mathcal{P}_i = \{\hat{p}_i(u_i, \ell_i, w_i) : u_i \in \mathfrak{R}_+, \ell_i \in \Theta, w_i \in \Theta\}$  as the collection of candidate price vectors for nest  $i$ , then we can stitch together an optimal solution to problem (4.13) by picking one price vector from each one of the collections  $\mathcal{P}_1, \dots, \mathcal{P}_m$ . To see this result, letting  $u_i^*$ ,  $\ell_i^*$  and  $w_i^*$  be as defined in Theorem 4.4.3, we have  $\hat{p}_i(u_i^*, \ell_i^*, w_i^*) \in \{\hat{p}_i(u_i, \ell_i, w_i) : u_i \in \mathfrak{R}_+, \ell_i \in \Theta, w_i \in \Theta\}$  for all  $i \in M$ . Furthermore, since problem (4.18) with  $u_i = u_i^*$ ,  $\ell_i = \ell_i^*$  and  $w_i = w_i^*$  is identical to problem (4.17), Theorem 4.4.3 implies that  $(\hat{p}_1(u_1^*, \ell_1^*, w_1^*), \dots, \hat{p}_m(u_m^*, \ell_m^*, w_m^*))$  is an optimal solution to problem (4.13). Therefore, if we use the collection  $\mathcal{P}_i = \{\hat{p}_i(u_i, \ell_i, w_i) : u_i \in \mathfrak{R}_+, \ell_i \in \Theta, w_i \in \Theta\}$  as the collection of candidate price vectors for nest  $i$ , then we can stitch together an optimal solution to problem (4.13) by picking one price vector from each one of the collections  $\mathcal{P}_1, \dots, \mathcal{P}_m$ .

Noting the discussion above, we can use  $\{\hat{p}_i(u_i, \ell_i, w_i) : u_i \in \mathfrak{R}_+, \ell_i \in \Theta, w_i \in \Theta\}$  as the collection of candidate price vectors to charge in nest  $i$ . In the subsequent sections, we show that for a given  $\ell_i \in \Theta$  and  $w_i \in \Theta$ , the collection  $\{\hat{p}_i(u_i, \ell_i, w_i) : u_i \in \mathfrak{R}_+\}$  includes at most  $nq$  price vectors and we can find these

price vectors in a tractable fashion. Therefore, since there are  $q$  possible values for each of  $\ell_i$  and  $w_i$ , the collection  $\{\hat{p}_i(u_i, \ell_i, w_i) : u_i \in \mathfrak{R}_+, \ell_i \in \Theta, w_i \in \Theta\}$  includes at most  $nq^3$  price vectors.

#### 4.4.4 Counting Candidate Price Vectors

In this section, we consider problem (4.18) for fixed values of  $\ell_i \in \Theta$  and  $w_i \in \Theta$ . We show that there exists a collection of price vectors  $\mathcal{P}_i = \{p_i^t : t \in \mathcal{T}_i\}$  such that this collection includes an optimal solution to problem (4.18) for any value of  $u_i \in \mathfrak{R}_+$  and there are at most  $nq$  price vectors in this collection. To show this result, we write problem (4.18) as

$$\begin{aligned} \max_{p_i \in \Theta :} \quad & \left\{ \sum_{j \in N} v_{ij}(p_{ij}) \left[ \frac{\sum_{j \in N} p_{ij} v_{ij}(p_{ij})}{\sum_{j \in N} v_{ij}(p_{ij})} - u_i \right] \right\} \\ & \ell_i \leq p_{ij} \leq w_i \quad \forall j \in N \\ & = \max_{p_i \in \Theta :} \left\{ \sum_{j \in N} (p_{ij} - u_i) v_{ij}(p_{ij}) \right\}. \quad (4.19) \\ & \ell_i \leq p_{ij} \leq w_i \quad \forall j \in N \end{aligned}$$

In the next lemma, we begin by showing that as the value of  $u_i$  in problem (4.19) gets larger, then the optimal price for each product also gets larger. This lemma is similar to Lemma 4.3.4 but its proof is significantly simpler than that of Lemma 4.3.4 since the prices in problem (4.19) has only upper and lower bound constraints, rather than a price ladder constraint.

**Lemma 4.4.4.** *Using  $\hat{p}_i(u_i) = (\hat{p}_{i1}(u_i), \dots, \hat{p}_{in}(u_i))$  to denote an optimal solution to problem (4.19) as a function of  $u_i$ , if we have  $u_i^- < u_i^+$ , then it holds that  $\hat{p}_{ij}(u_i^-) \leq \hat{p}_{ij}(u_i^+)$  for all  $j \in N$ .*

*Proof.* To get a contradiction, assume that  $u_i^- < u_i^+$ , but we have  $\hat{p}_{ij}(u_i^-) > \hat{p}_{ij}(u_i^+)$  for some  $j \in N$ . For notational brevity, we let  $\hat{p}_i^- = \hat{p}_i(u_i^-)$  and  $\hat{p}_i^+ = \hat{p}_i(u_i^+)$ . Noting

that  $\hat{p}_{ij}^- > \hat{p}_{ij}^+$  and using the fact that the preference weight of a product gets larger as we charge a smaller price for the product, we obtain  $v_{ij}(\hat{p}_{ij}^-) < v_{ij}(\hat{p}_{ij}^+)$ . In problem (4.19), if we charge the price  $p_{ij}$  for product  $j$ , then this product makes a contribution of  $(p_{ij} - u_i) v_{ij}(p_{ij})$  to the objective function. We note that  $\hat{p}_i^+$  is an optimal solution to problem (4.19) when we solve this problem with  $u_i = u_i^+$ . Therefore, if we solve problem (4.19) with  $u_i = u_i^+$ , then the contribution of product  $j$  when we charge the price  $p_{ij}^+$  for this product should be at least as large as the contribution when we charge the price  $p_{ij}^-$ . Otherwise, it would not be optimal to charge the price  $p_{ij}^+$  for product  $j$  when we solve problem (4.19) with  $u_i = u_i^+$ . Thus, we obtain  $(p_{ij}^+ - u_i^+) v_{ij}(p_{ij}^+) \geq (p_{ij}^- - u_i^+) v_{ij}(p_{ij}^-)$ . Similarly,  $\hat{p}_i^-$  is an optimal solution to problem (4.19) when we solve this problem with  $u_i = u_i^-$ . Therefore, following an argument similar to the preceding one, it holds that  $(p_{ij}^- - u_i^-) v_{ij}(p_{ij}^-) \geq (p_{ij}^+ - u_i^-) v_{ij}(p_{ij}^+)$ . Adding the last two inequalities and canceling the common terms, we obtain  $u_i^- (v_{ij}(\hat{p}_{ij}^+) - v_{ij}(\hat{p}_{ij}^-)) \geq u_i^+ (v_{ij}(\hat{p}_{ij}^+) - v_{ij}(\hat{p}_{ij}^-))$ . Noting that  $v_{ij}(\hat{p}_{ij}^-) < v_{ij}(\hat{p}_{ij}^+)$  by the discussion at the beginning of the proof, the last inequality implies that  $u_i^- \geq u_i^+$ , which is a contradiction.  $\square$

In the next theorem, we build on the lemma above show that there exists a collection of at most  $nq$  price vectors such that this collection includes an optimal solution to problem (4.19) for any value of  $u_i \in \mathfrak{R}_+$ . The proof of this theorem uses Lemma 4.4.4 and it follows from an outline that is identical to that of Theorem 4.3.5. Thus, we omit the proof.

**Theorem 4.4.5.** *There exists a collection of at most  $nq$  price vectors such that this collection includes an optimal solution to problem (4.19) for any value of  $u_i \in \mathfrak{R}_+$ .*

By Theorem 4.4.5, for fixed values of  $\ell_i \in \Theta$  and  $w_i \in \Theta$ , there exists a collection of at most  $nq$  price vectors such that this collection includes an optimal solution to problem (4.19) for any value of  $u_i \in \mathfrak{R}_+$ . In the next section, we show how to

construct this collection. Since there are  $q$  possible values for each of  $\ell_i$  and  $w_i$ , repeating our approach for all possible values of  $\ell_i$  and  $w_i$ , it follows that there exists a collection of at most  $nq^3$  price vectors such that this collection includes an optimal solution to problem (4.19) for any value of  $u_i \in \mathfrak{R}_+$ ,  $\ell_i \in \Theta$  and  $w_i \in \Theta$ .

#### 4.4.5 Constructing Candidate Price Vectors

In the previous section, we consider problem (4.19) for fixed values of  $\ell_i \in \Theta$  and  $w_i \in \Theta$ . We show that there exists a collection of at most  $nq$  price vectors such that this collection includes an optimal solution to problem (4.19) for any value of  $u_i \in \mathfrak{R}_+$ . In this section, we show how to come up with this collection in a tractable fashion. Our approach builds on a linear programming formulation of problem (4.19). To give this linear programming formulation, we use the decision variables  $\{x_{ij}(p_{ij}) : p_{ij} \in \Theta, j \in N\}$ , where  $x_{ij}(p_{ij}) = 1$  if we charge price  $p_{ij}$  for product  $j$  in nest  $i$ , otherwise  $x_{ij}(p_{ij}) = 0$ . In this case, problem (4.19) can be written as

$$\begin{aligned}
\max \quad & \sum_{j \in N} \sum_{p_{ij} \in \Theta} (p_{ij} - u_i) v_{ij}(p_{ij}) x_{ij}(p_{ij}) & (4.20) \\
\text{s.t.} \quad & \sum_{p_{ij} \in \Theta} x_{ij}(p_{ij}) = 1 & \forall j \in N \\
& x_{ij}(p_{ij}) = 0 & \forall p_{ij} \notin \{\ell_i, \dots, w_i\}, j \in N \\
& x_{ij}(p_{ij}) \in \{0, 1\} & \forall p_{ij} \in \Theta, j \in N.
\end{aligned}$$

In the problem above, the first set of constraints ensures that we choose one price for each product, whereas the second set of constraints ensures that the price of each product is between  $\ell_i$  and  $w_i$ . Using the second set of constraints, we can set the values of the decision variables  $\{x_{ij}(p_{ij}) : p_{ij} \notin \{\ell_i, \dots, w_i\}, j \in N\}$  to zero and drop these decision variables from problem (4.20). On the other hand, each

row of the constraint matrix corresponding to the first set of constraints includes consecutive ones. Such a matrix is called an interval matrix and interval matrices are totally unimodular; see [44]. Therefore, we can obtain an optimal solution to problem (4.20) by solving its linear programming relaxation. Also, we observe that the value of  $u_i \in \mathfrak{R}_+$  only affects the objective function coefficients in problem (4.20). Thus, we can vary  $u_i \in \mathfrak{R}_+$  parametrically and solve problem (4.20) by using the parametric simplex method to generate the optimal solutions to this problem for all values of  $u_i \in \mathfrak{R}_+$ . These solutions correspond to the optimal solutions to problem (4.19) for all values of  $u_i \in \mathfrak{R}_+$ .

Therefore, for fixed values of  $\ell_i \in \Theta$  and  $w_i \in \Theta$ , we solve problem (4.20) by using the parametric simplex method to generate the optimal solutions to this problem for all values of  $u_i \in \mathfrak{R}_+$ . Repeating this approach for all possible values of  $\ell_i$  and  $w_i$ , we obtain the optimal solutions to problem (4.20) for all values of  $u_i \in \mathfrak{R}_+$ ,  $\ell_i \in \Theta$  and  $w_i \in \Theta$ . By the discussion that follows Theorem 4.4.3, we can use the optimal solutions to problem (4.20) for all values of  $u_i \in \mathfrak{R}_+$ ,  $\ell_i \in \Theta$  and  $w_i \in \Theta$  as the collection of candidate price vectors  $\mathcal{P}_i$  in nest  $i$ . Once we have the collection of candidate price vectors in each nest, we can solve the linear program in (4.16) to find the value of  $\hat{z}$  that satisfies  $v_0 \hat{z} = g(\hat{z})$ . Since there are at most  $nq^3$  price vectors in each one of the collections  $\mathcal{P}_1, \dots, \mathcal{P}_m$ , we have  $|\mathcal{M}(w_{i-1})| = O(nq^3)$ , which implies that the linear program in (4.16) has  $O(mq)$  decision variables and  $O(mnq^4)$  constraints. In this case, by Theorem 4.4.1, we can solve problem (4.14) with  $z = \hat{z}$  to obtain an optimal solution to problem (4.13). To solve problem (4.14) with  $z = \hat{z}$ , we can simply solve the dynamic program in (4.15) with  $z = \hat{z}$ .

## 4.5 Numerical Experiments

In this section, we provide numerical experiments to show that the approaches in Sections 4.3 and 4.4 can obtain the optimal solutions to problems (4.3) and (4.13) reasonably fast. We also investigate the number of candidate price vectors that we construct to obtain the optimal solutions.

### 4.5.1 Price Ladders Inside Nests

In this section, we consider problem instances with price ladders inside nests. In our numerical experiments, we vary the number of nests over  $m \in \{2, 4, 6\}$ , the number of products in each nest over  $n \in \{10, 20, 30\}$  and the number of possible prices for each product over  $q \in \{10, 20, 30\}$ . This setup provides 27 parameter combinations for  $(m, n, q)$ . In each parameter combination, we generate 10 individual problem instances by using the following approach. The possible prices for each product take values over the interval  $[1, 10]$  and we obtain the prices  $\{\theta^1, \dots, \theta^q\}$  by dividing the interval  $[1, 10]$  into  $q$  equal pieces. To come up with the preference weights, we sample  $\alpha_{ij}$  and  $\beta_{ij}$  from the uniform distribution over the interval  $[0, 2]$  for all  $i \in M, j \in N$ . The preference weight of product  $j$  in nest  $i$  corresponding to the price  $p_{ij}$  is given by  $\exp(\alpha_{ij} - \beta_{ij} p_{ij})$ . The nested logit model has a random utility maximization interpretation, where a customer associates random utilities with the products and the no purchase option, choosing the option with the largest utility. In the random utility maximization setup,  $\alpha_{ij}$  captures the nominal mean utility of product  $j$  in nest  $i$  and  $\beta_{ij}$  captures how the mean utility of product  $j$  in nest  $i$  changes as a function of its price; see [38]. We sample the dissimilarity parameter  $\gamma_i$  for each nest  $i$  from the uniform distribution over the interval  $[0.25, 1]$ . For each problem instance, we use the approach described at the end of Section 4.3.5 to obtain an optimal solution to problem (4.3).



We summarize our numerical results in Table 4.1. The first column in this table shows the parameter configurations for our test problems. We recall that we generate 10 individual problem instances in each parameter configuration. The second column shows the average CPU seconds to obtain an optimal solution to problem (4.3), where the average is computed over 10 problem instances that we generate for a particular parameter combination. The third and fourth columns respectively show the maximum and minimum CPU seconds over 10 problem instances. Similar to the average, the maximum and minimum are computed over 10 problem instances that we generate for a particular parameter combination. There are two main steps in obtaining an optimal solution to problem (4.3). First, we construct the collections of candidate price vectors for each nest, which requires solving problem (4.12) by using the parametric simplex method to generate the possible optimal solutions to this problem for all values of  $u_i \in \mathfrak{R}_+$ . Second, we solve problem (4.6) to stitch together an optimal solution by using the collection of candidate price vectors for each nest. The fifth column in Table 4.1 shows what percent of the CPU seconds is spent on generating the collections of candidate price vectors. The remaining portion of the CPU seconds is spent on stitching together an optimal solution. The sixth column shows the average number of price vectors in the collection that we generate for each nest, where the average is computed over all nests in a problem instance and over 10 problem instances that we generate for a particular parameter combination. The seventh and eighth columns respectively show the maximum and minimum number of price vectors in the collection that we generate for each nest.

To demonstrate the potential importance of generating the collections of candidate price vectors carefully, we also find the best solution to problem (4.3) that charges a constant price in each nest. Letting  $e \in \mathfrak{R}^n$  be the vector of all ones,

this solution can be obtained by using the collection  $\mathcal{P}_i = \{\theta^1 e, \dots, \theta^q e\}$  as the collection of candidate price vectors for each nest  $i$  and solving problem (4.6) by using this collection for each nest  $i$ . ([34] show that if  $\beta_{ij} = \beta_{ik}$  for all  $j, k \in N$  and  $i \in M$ , then it is optimal to charge a constant price in each nest.) The ninth column in Table 4.1 shows the average percent gap between the optimal objective value of problem (4.3) and the best expected revenue obtained by charging a constant price in each nest, where the average is computed over 10 problem instances that we generate for a particular parameter combination. In other words, using  $\text{Opt}^k$  to denote the optimal expected revenue for problem instance  $k$  that we generate for a particular parameter combination and  $\text{ConP}^k$  to denote the best expected revenue obtained by charging a constant price in each nest, the ninth column shows the average of the data  $\{100 \times (\text{Opt}^k - \text{ConP}^k) / \text{Opt}^k : k = 1, \dots, 10\}$ . The tenth and eleventh columns show the maximum and minimum percent gaps between the optimal objective value of problem (4.3) and the best expected revenue obtained by charging a constant price in each nest.

Our computational results indicate that we can obtain an optimal solution to problem (4.3) rather fast. For the largest problem instances with  $m = 6$ ,  $n = 30$  and  $q = 30$ , we can obtain an optimal solution in less than two seconds and the average CPU seconds for these problem instances is about 1.4. Naturally, the CPU seconds have an increasing trend as the number of nests, products or possible prices increases. We observe that almost all of the CPU seconds are spent on constructing the collections of candidate price vectors. In Section 4.3.4, we show that we need to construct at most  $nq$  candidate price vectors in each nest, but our numerical results demonstrate that the number of candidate price vectors that we actually end up constructing can be substantially smaller than the upper bound of  $nq$ . For example, for the problem instances with  $n = 30$  and  $q = 30$ , we have  $nq = 900$ , but

Param. Comb. ( $m, n, q$ )	Total CPU Secs.			Perc. Time Const. Cand.	No. of Cand. Price Vectors per Nest			Perc. Opt. Gap of Const. Price in Nests		
	Avg.	Max.	Min.		Avg.	Max.	Min.	Avg.	Max.	Min.
(2, 10, 10)	0.01	0.01	0.00	91.89	23	34	10	9.05	22.98	0.01
(2, 10, 20)	0.05	0.06	0.03	96.92	52	64	34	9.24	19.45	0.01
(2, 10, 30)	0.12	0.18	0.06	98.00	73	97	35	11.40	18.90	0.00
(2, 20, 10)	0.02	0.02	0.01	95.35	29	40	12	10.62	20.04	0.01
(2, 20, 20)	0.10	0.13	0.03	98.64	55	74	14	12.38	20.84	0.01
(2, 20, 30)	0.26	0.38	0.14	99.17	78	117	41	11.17	20.65	0.01
(2, 30, 10)	0.03	0.04	0.01	96.50	30	41	13	8.72	18.37	0.02
(2, 30, 20)	0.14	0.24	0.06	98.89	52	98	22	6.33	16.28	0.01
(2, 30, 30)	0.49	0.78	0.26	99.47	98	163	48	8.71	17.45	0.03
(4, 10, 10)	0.01	0.02	0.01	92.48	22	30	11	6.86	12.87	0.00
(4, 10, 20)	0.08	0.09	0.06	97.10	43	55	34	7.01	11.57	0.04
(4, 10, 30)	0.23	0.32	0.17	98.58	66	95	48	7.20	13.43	0.02
(4, 20, 10)	0.03	0.04	0.02	96.31	25	31	15	5.24	11.19	0.01
(4, 20, 20)	0.17	0.23	0.10	98.36	50	69	29	4.18	10.51	0.02
(4, 20, 30)	0.53	0.77	0.30	99.24	77	115	43	5.65	10.85	0.01
(4, 30, 10)	0.05	0.07	0.04	97.12	30	41	22	6.88	10.46	0.01
(4, 30, 20)	0.26	0.31	0.20	98.90	51	62	38	5.97	10.19	0.02
(4, 30, 30)	0.85	1.03	0.50	99.48	84	103	47	6.99	11.00	0.02
(6, 10, 10)	0.02	0.03	0.01	91.71	21	31	10	5.17	10.93	0.00
(6, 10, 20)	0.12	0.15	0.05	97.39	42	59	20	6.27	9.88	0.00
(6, 10, 30)	0.34	0.43	0.23	98.32	65	84	45	5.90	10.79	0.02
(6, 20, 10)	0.05	0.07	0.04	95.51	28	40	19	5.96	10.11	0.01
(6, 20, 20)	0.28	0.38	0.15	98.58	53	74	29	5.55	9.40	0.00
(6, 20, 30)	0.79	1.12	0.49	99.29	78	113	45	5.11	9.64	0.01
(6, 30, 10)	0.07	0.10	0.05	97.41	27	35	21	3.98	10.83	0.01
(6, 30, 20)	0.47	0.61	0.38	99.04	59	80	47	5.81	10.74	0.00
(6, 30, 30)	1.40	1.89	1.12	99.51	89	118	72	7.26	10.93	0.02

Table 4.1: Computational results for test problems with price ladders inside nests.

the average number of candidate price vectors that we construct for each nest is about 90 and the number of candidate price vectors that we construct for each nest does not exceed 163. Charging a constant price in each nest can provide a good solution for some problem instances, but it is not too surprising to see that this approach is generally not reliable. There are numerous problem instances in Table 4.1 where the optimal objective value exceeds the best expected revenue obtained by charging a constant price in each nest by more than 20%.

## 4.5.2 Price Ladders Between Nests

In this section, we consider problem instances with price ladders between nests. We generate our problem instances by using the same approach that we use for generating the problem instances with price ladders inside nests. For each problem instance, we use the approach described at the end of Section 4.4.5 to obtain an optimal solution to problem (4.13). In particular, to construct the collection of candidate price vectors for each nest, we solve problem (4.20) through the parametric simplex method to generate the possible optimal solutions to this problem for all values of  $u_i \in \mathfrak{R}_+$ . In this case, we solve problem (4.16) to find the value of  $\hat{z}$  satisfying  $v_0 \hat{z} = g(\hat{z})$  and to stitch together an optimal solution by using the collection of candidate price vectors for each nest.

We summarize our numerical results in Table 4.2. The layout of this table is identical to that of Table 4.1. For the problem instances with 10 possible prices for each product, we can obtain an optimal solution in 1.34 seconds. For the largest problem instances with  $m = 6$ ,  $n = 30$  and  $q = 30$ , the CPU seconds are below two minutes. On average, about half of the CPU seconds is spent on constructing the collections of candidate price vectors. In Section 4.4.4, we show that we need to construct at most  $nq^3$  price vectors in each nest, but we actually end up generating substantially fewer candidate price vectors. For example, for the problem instances with  $n = 30$  and  $q = 30$ , we have  $nq^3 = 810,000$ , but the average number of candidate price vectors that we construct for each nest is about 25,000. Finally, the best expected revenue obtained by charging a constant price in each nest can deviate significantly from the optimal expected revenue and there are problem instances where this approach suffers optimality gaps that exceed 20%.

Param. Comb. ( $m, n, q$ )	Total CPU Secs.			Perc. Time Const. Cand.	No. of Cand. Price Vectors per Nest			Perc. Opt. Gap of Const. Price in Nests		
	Avg.	Max.	Min.		Avg.	Max.	Min.	Avg.	Max.	Min.
(2, 10, 10)	0.06	0.08	0.05	42.90	357	463	262	9.01	22.79	0.01
(2, 10, 20)	0.77	1.00	0.59	58.00	1,798	2,447	882	9.35	19.59	0.02
(2, 10, 30)	3.60	4.43	3.03	69.65	5,845	8,191	3,914	11.37	19.19	0.01
(2, 20, 10)	0.19	0.20	0.18	47.24	924	1,030	674	9.97	18.84	0.01
(2, 20, 20)	2.75	2.99	2.54	58.24	4,833	6,276	3,481	11.54	18.93	0.01
(2, 20, 30)	14.05	15.76	11.56	69.99	15,797	21,829	12,189	10.54	19.87	0.01
(2, 30, 10)	0.37	0.40	0.33	53.34	1,332	1,476	1,061	10.52	22.18	0.02
(2, 30, 20)	5.55	5.90	5.09	65.54	8,432	9,921	6,873	7.65	19.69	0.01
(2, 30, 30)	32.01	35.95	29.40	72.53	21,821	26,983	14,980	10.53	21.42	0.02
(4, 10, 10)	0.15	0.16	0.13	40.00	284	374	206	7.50	13.97	0.00
(4, 10, 20)	1.87	2.57	1.37	49.60	1,981	2,980	899	7.69	13.22	0.02
(4, 10, 30)	9.52	12.54	6.71	56.84	5,246	7,580	3,373	7.95	15.14	0.02
(4, 20, 10)	0.46	0.54	0.40	41.59	853	954	688	5.89	12.60	0.01
(4, 20, 20)	6.28	7.45	5.25	52.35	5,233	6,390	4,451	4.61	11.82	0.02
(4, 20, 30)	30.65	33.15	28.63	60.86	15,433	19,105	13,145	6.33	12.12	0.02
(4, 30, 10)	0.81	0.83	0.75	46.37	1,368	1,450	1,219	7.97	12.12	0.01
(4, 30, 20)	11.62	12.34	10.38	60.00	7,753	9,109	5,289	6.92	11.82	0.02
(4, 30, 30)	68.79	71.38	64.05	63.61	24,193	27,144	21,519	8.18	12.89	0.02
(6, 10, 10)	0.20	0.25	0.18	36.71	301	365	223	4.85	10.23	0.00
(6, 10, 20)	2.54	2.90	2.08	49.89	1,733	2,202	1,070	5.93	9.39	0.00
(6, 10, 30)	13.86	16.93	11.86	52.79	5,928	8,913	4,201	5.50	10.05	0.01
(6, 20, 10)	0.66	0.74	0.59	39.98	863	984	737	5.39	9.33	0.01
(6, 20, 20)	8.92	9.48	7.81	52.88	4,899	5,513	3,992	5.02	8.52	0.00
(6, 20, 30)	48.62	51.89	44.69	57.91	15,677	18,043	12,110	4.58	8.64	0.00
(6, 30, 10)	1.25	1.34	1.23	44.98	1,335	1,446	1,259	3.72	9.91	0.01
(6, 30, 20)	18.31	19.17	17.29	56.94	8,049	9,222	7,078	5.54	10.78	0.00
(6, 30, 30)	108.72	116.77	102.09	60.95	24,930	27,364	21,331	6.92	10.93	0.01

Table 4.2: Computational results for test problems with price ladders between nests.

### Appendix: Proof of Theorem 4.4.3

In this section, we show Theorem 4.4.3. We need the two intermediate lemmas to show Theorem 4.4.3. In the next lemma, we show an ordering between the optimal expected revenues from a customer that has already decided to make a purchase in different nests.

**Lemma 4.5.1.** *If  $(p_1^*, \dots, p_m^*)$  is an optimal solution to problem (4.13), then we have  $R_1(p_1^*) \leq R_2(p_2^*) \leq \dots \leq R_m(p_m^*)$ .*

*Proof.* Since  $(p_1^*, \dots, p_m^*)$  is a feasible solution to problem (4.13), we have  $p_i \in \mathcal{G}_i(p_{i-1})$ , which implies that  $\max_{j \in N} p_{i-1,j}^* \leq \min_{j \in N} p_{ij}^*$ . Therefore, the largest price in the price vector  $p_{i-1}^*$  is smaller than the smallest price in the price vector  $p_i^*$ . By (4.1), we observe that  $R_{i-1}(p_{i-1}^*)$  is a convex combination of the prices in the price vector  $p_{i-1}^*$ , whereas  $R_i(p_i^*)$  is a convex combination of the prices in the price vector  $p_i^*$ . Since the largest price in the price vector  $p_{i-1}^*$  is smaller than the smallest price in the price vector  $p_i^*$ , we obtain  $R_{i-1}(p_{i-1}^*) \leq R_i(p_i^*)$ .  $\square$

In the next lemma, we show that if the optimal expected revenue from a customer that has already decided to make a purchase in a particular nest does not exceed the optimal expected revenue, then the smallest price in the next nest does not exceed the optimal expected revenue.

**Lemma 4.5.2.** *If  $(p_1^*, \dots, p_m^*)$  is an optimal solution to problem (4.13) providing the objective value  $z^*$  and  $R_i(p_i^*) < z^*$  for some  $i \in M$ , then we have  $i \in M \setminus \{m\}$  and  $\min_{j \in N} p_{i+1,j}^* < z^*$ .*

*Proof.* First, we show that if  $R_i(p_i^*) < z^*$  for some  $i \in M$ , then we have  $i \in M \setminus \{m\}$ . To get a contradiction, assume that  $R_m(p_m^*) < z^*$ . By Lemma 4.5.1, we have  $R_1(p_1^*) \leq R_2(p_2^*) \leq \dots \leq R_m(p_m^*) < z^*$ . Thus, we obtain  $\Pi(p_1^*, \dots, p_m^*) = \sum_{i \in M} V_i(p_i^*)^{\gamma_i} R_i(p_i^*) / (v_0 + \sum_{i \in M} V_i(p_i^*)^{\gamma_i}) < \sum_{i \in M} V_i(p_i^*)^{\gamma_i} z^* / (v_0 + \sum_{i \in M} V_i(p_i^*)^{\gamma_i}) < z^*$ , which contradicts the fact that  $(p_1^*, \dots, p_m^*)$  is an optimal solution to problem (4.13).

Second, we show that if  $R_i(p_i^*) < z^*$  for some  $i \in M$ , then  $\min_{j \in N} p_{i+1,j}^* < z^*$ . To get a contradiction assume that there exists a nest  $k$  such that  $R_k(p_k^*) < z^*$  and  $\min_{j \in N} p_{k+1,j}^* \geq z^*$ . For notational brevity, we let  $\ell_{k+1}^* = \min_{j \in N} p_{k+1,j}^*$ . By our assumption, we have  $\ell_{k+1}^* \geq z^*$ . We define a solution  $(\hat{p}_1, \dots, \hat{p}_m)$  to problem (4.13) as  $\hat{p}_i = p_i^*$  for all  $i \in M \setminus \{k\}$  and  $\hat{p}_{kj} = \ell_{k+1}^*$  for all  $j \in N$ . Since the solutions  $(p_1^*, \dots, p_m^*)$  and  $(\hat{p}_1, \dots, \hat{p}_m)$  charge the same prices in

all nests other than nest  $k$ , we have  $V_i(p_i^*)^{\gamma_i} (R_i(p_i^*) - z^*) = V_i(\hat{p}_i)^{\gamma_i} (R_i(\hat{p}_i) - z^*)$  for all  $i \in M \setminus \{k\}$ . For nest  $k$ , we have  $R_k(p_k^*) < z^*$ , but  $R_k(\hat{p}_k) = \sum_{j \in N} \hat{p}_{kj} v_{kj}(\hat{p}_{kj}) / \sum_{j \in N} v_{kj}(\hat{p}_{kj}) = \sum_{j \in N} \ell_{k+1}^* v_{kj}(\hat{p}_{kj}) / \sum_{j \in N} v_{kj}(\hat{p}_{kj}) = \ell_{k+1}^* \geq z^*$ . Thus, we obtain  $V_k(p_k^*)^{\gamma_k} (R_k(p_k^*) - z^*) < 0 \leq V_k(\hat{p}_k)^{\gamma_k} (R_k(\hat{p}_k) - z^*)$ . The discussion so far in this paragraph shows that  $V_i(p_i^*)^{\gamma_i} (R_i(p_i^*) - z^*) \leq V_i(\hat{p}_i)^{\gamma_i} (R_i(\hat{p}_i) - z^*)$  for all  $i \in M$  and the inequality holds as strict inequality for nest  $k$ . Adding this inequality over all  $i \in M$ , we have  $\sum_{i \in M} V_i(p_i^*)^{\gamma_i} (R_i(p_i^*) - z^*) < \sum_{i \in M} V_i(\hat{p}_i)^{\gamma_i} (R_i(\hat{p}_i) - z^*)$ . On the other hand, since  $(p_1^*, \dots, p_m^*)$  is an optimal solution to problem (4.13), we have  $z^* = \sum_{i \in M} V_i(p_i^*) R_i(p_i^*) / (v_0 + \sum_{i \in M} V_i(p_i^*)^{\gamma_i})$  and arranging the terms in this equality yields  $v_0 z^* = \sum_{i \in M} V_i(p_i^*)^{\gamma_i} (R_i(p_i^*) - z^*)$ . In this case, having  $\sum_{i \in M} V_i(p_i^*)^{\gamma_i} (R_i(p_i^*) - z^*) < \sum_{i \in M} V_i(\hat{p}_i)^{\gamma_i} (R_i(\hat{p}_i) - z^*)$  and  $v_0 z^* = \sum_{i \in M} V_i(p_i^*)^{\gamma_i} (R_i(p_i^*) - z^*)$  yields  $v_0 z^* < \sum_{i \in M} V_i(\hat{p}_i)^{\gamma_i} (R_i(\hat{p}_i) - z^*)$ . Solving for  $z^*$  in this inequality, we obtain  $z^* < \sum_{i \in M} V_i(\hat{p}_i) R_i(\hat{p}_i) / (v_0 + \sum_{i \in M} V_i(\hat{p}_i)^{\gamma_i}) = \Pi(\hat{p}_1, \dots, \hat{p}_m)$ . Thus, the solution  $(\hat{p}_1, \dots, \hat{p}_m)$  provides an objective value for problem (4.13) that is strictly larger than the optimal objective value. In the rest of the proof, we show that  $(\hat{p}_1, \dots, \hat{p}_m)$  is a feasible solution to problem (4.13), which yields a contradiction and the desired result follows.

We have  $\min_{j \in N} \hat{p}_{kj} = \ell_{k+1}^* = \min_{j \in N} p_{k+1,j}^* \geq \max_{j \in N} p_{kj}^* \geq \min_{j \in N} p_{kj}^* \geq \max_{j \in N} p_{k-1,j}^* = \max_{j \in N} \hat{p}_{k-1,j}$ , where the first equality uses the definition of  $\hat{p}_k$ , the second equality uses the definition of  $\ell_{k+1}^*$ , the first and third inequalities use the fact that  $(p_1^*, \dots, p_m^*)$  is a feasible solution to problem (4.13) so that  $p_{k+1}^* \in \mathcal{G}_{k+1}(p_k)$  and  $p_k^* \in \mathcal{G}_k(p_{k-1}^*)$  and the last equality is by the definition of  $\hat{p}_{k-1}$ . Thus, this chain of inequalities shows that  $\hat{p}_k \in \mathcal{G}_k(\hat{p}_{k-1})$ . Similarly, we have  $\min_{j \in N} \hat{p}_{k+1,j} = \min_{j \in N} p_{k+1,j}^* = \ell_{k+1}^* = \max_{j \in N} \hat{p}_{kj}$ , where the first and third equalities use the definitions of  $\hat{p}_{k+1}$  and  $\hat{p}_k$ , whereas the second equality uses the definition of  $\ell_{k+1}^*$ . Thus, this chain of equalities shows that  $\hat{p}_{k+1} \in \mathcal{G}_{k+1}(\hat{p}_k)$ . Since the solutions

$(p_1^*, \dots, p_m^*)$  and  $(\hat{p}_1, \dots, \hat{p}_m)$  charge the same prices in all nests other than nest  $k$  and  $(p_1^*, \dots, p_m^*)$  is a feasible solution to problem (4.13), we have  $\hat{p}_i \in \mathcal{G}_i(\hat{p}_{i-1})$  for all  $i \in M \setminus \{1, k, k+1\}$  as well. Therefore, we have  $\hat{p}_i \in \mathcal{G}_i(\hat{p}_{i-1})$  for all  $i \in M \setminus \{1\}$ , which indicates that  $(\hat{p}_1, \dots, \hat{p}_m)$  is a feasible solution to problem (4.13).  $\square$

In the rest of this section, we show Theorem 4.4.3.

For notational brevity, we let  $R_i^* = R_i(p_i^*)$ ,  $V_i^* = V_i(p_i^*)$ ,  $\hat{R}_i = R_i(\hat{p}_i)$  and  $\hat{V}_i = V_i(\hat{p}_i)$ . First, we consider a nest  $i$  that satisfies  $R_i^* < z^*$ . By Lemma 4.5.2, we observe that  $i \in M \setminus \{m\}$ . Since  $\hat{p}_i$  is a feasible solution to problem (4.17), we have  $\hat{p}_{ij} \leq w_i^*$  for all  $j \in N$ , where  $w_i^*$  is as defined in Theorem 4.4.3. We claim that  $\hat{p}_{ij} = w_i^*$  for all  $j \in N$ . To get a contradiction, assume that  $\hat{p}_{ij} < w_i^*$  for some  $j \in N$ . Since  $(p_1^*, \dots, p_m^*)$  is a feasible solution to problem (4.13), we have  $p_{i+1}^* \in \mathcal{G}_{i+1}(p_i^*)$ , which implies that  $\min_{j \in N} p_{i+1,j}^* \geq \max_{j \in N} p_{ij}^* = w_i^*$ , where the equality is by the definition of  $w_i^*$  given in Theorem 4.4.3. On the other hand, since  $R_i^* < z^*$ , Lemma 4.5.2 implies that  $\min_{j \in N} p_{i+1,j}^* < z^*$ . Therefore, we obtain  $w_i^* = \max_{j \in N} p_{ij}^* \leq \min_{j \in N} p_{i+1,j}^* < z^*$ . We define a solution  $\tilde{p}_i = (\tilde{p}_{i1}, \dots, \tilde{p}_{in})$  to problem (4.17) as  $\tilde{p}_{ij} = w_i^*$  for all  $j \in N$ . This solution is clearly feasible to problem (4.17) and satisfies  $R_i(\tilde{p}_i) = \sum_{j \in N} w_i^* v_{ij}(p_{ij}^*) / \sum_{j \in N} v_{ij}(p_{ij}^*) = w_i^* < z^*$ . Furthermore, we have  $R_i(\hat{p}_i) = \sum_{j \in N} \hat{p}_{ij} v_{ij}(\hat{p}_{ij}) / \sum_{j \in N} v_{ij}(\hat{p}_{ij}) \leq \sum_{j \in N} w_i^* v_{ij}(\hat{p}_{ij}) / \sum_{j \in N} v_{ij}(\hat{p}_{ij}) = w_i^* = R_i(\tilde{p}_i)$ . By the last two chains of inequalities, we get  $z^* - R_i(\hat{p}_i) \geq z^* - R_i(\tilde{p}_i) = z^* - w_i^* > 0$ . Noting that the preference weight of a product becomes smaller as we charge a larger price, since  $\hat{p}_{ij} \leq w_i^* = \tilde{p}_{ij}$  for all  $j \in N$  and the inequality is strict for some  $j \in N$ , it holds that  $v_{ij}(\hat{p}_{ij}) \geq v_{ij}(\tilde{p}_{ij})$  for all  $j \in N$  and the inequality is strict for some  $j \in N$ . Thus, adding the last inequality over all  $j \in N$ , we obtain  $V_i(\hat{p}_i) > V_i(\tilde{p}_i)$ . In this case, having  $z^* - R_i(\hat{p}_i) \geq z^* - R_i(\tilde{p}_i) > 0$  and  $V_i(\hat{p}_i) > V_i(\tilde{p}_i)$  implies that  $V_i(\hat{p}_i)(z^* - R_i(\hat{p}_i)) > V_i(\tilde{p}_i)(z^* - R_i(\tilde{p}_i))$ . Since  $R_i^* < z^*$ , we have  $u_i^* = z^*$  by the definition of  $u_i^*$ , in which case, the last inequality



can equivalently be written as  $V_i(\hat{p}_i)(R_i(\hat{p}_i) - u_i^*) < V_i(\tilde{p}_i)(R_i(\tilde{p}_i) - u_i^*)$ , which contradicts the fact that  $\hat{p}_i$  is an optimal solution to problem (4.17). Thus, our claim holds and we have  $\hat{p}_{ij} = w_i^*$  for all  $j \in N$ .

By the claim established in the previous paragraph, we have  $\hat{p}_{ij} = w_i^*$  for all  $j \in N$ . Noting that  $w_i^* = \max_{j \in N} p_{ij}^*$  by the definition of  $w_i^*$ , we have  $\hat{p}_{ij} = w_i^* \geq p_{ij}^*$  for all  $j \in N$ . Since the preference weight of a product becomes smaller as we charge a larger price, the last inequality implies that  $v_{ij}(\hat{p}_{ij}) \leq v_{ij}(p_{ij}^*)$  for all  $j \in N$ . Adding this inequality over all  $j \in N$ , we obtain  $V_i(\hat{p}_i) \leq V_i(p_i^*)$ . Furthermore, we have  $R_i(\hat{p}_i) = \sum_{j \in N} w_i^* v_{ij}(\hat{p}_{ij}) / \sum_{j \in N} v_{ij}(\hat{p}_{ij}) = w_i^* = \max_{j \in N} p_{ij}^* \leq \min_{j \in N} p_{i+1,j}^* < z^*$ , where the first inequality is by the fact that  $(p_1^*, \dots, p_m^*)$  is a feasible solution to problem (4.13) so that  $p_{i+1}^* \in \mathcal{G}_{i+1}(p_i^*)$  and the second inequality follows by noting that  $R_i^* < z^*$  and using Lemma 4.5.2. Since  $w_i^* \geq p_{ij}^*$  for all  $j \in N$ , we have  $R_i(p_i^*) \leq \sum_{j \in N} w_i^* v_{ij}(\hat{p}_{ij}) / \sum_{j \in N} v_{ij}(\hat{p}_{ij}) = w_i^* = R_i(\hat{p}_i)$  as well. The last two chains of inequalities show that  $z^* - R_i(p_i^*) \geq z^* - R_i(\hat{p}_i) = z^* - w_i^* > 0$ . In this case, having  $V_i(\hat{p}_i) \leq V_i(p_i^*)$  and  $z^* - R_i(p_i^*) > z^* - R_i(\hat{p}_i) > 0$  yields  $V_i(p_i^*)^{\gamma_i} (z^* - R_i(p_i^*)) > V_i(\hat{p}_i)^{\gamma_i} (z^* - R_i(\hat{p}_i))$ . The last inequality shows that  $(V_i^*)^{\gamma_i} (R_i^* - z^*) < \hat{V}_i^{\gamma_i} (\hat{R}_i - z^*)$  for each nest  $i$  that satisfies  $R_i^* < z^*$ .

Second, we consider a nest  $i$  that satisfies  $R_i^* \geq z^*$ . In this case, we can follow the same argument at the beginning of the proof of Theorem 4.3.3 to show that  $(V_i^*)^{\gamma_i} (R_i^* - z^*) \leq \hat{V}_i^{\gamma_i} (\hat{R}_i - z^*)$  for each nest  $i$  that satisfies  $R_i^* \geq z^*$ . Therefore, we obtain  $(V_i^*)^{\gamma_i} (R_i^* - z^*) \leq \hat{V}_i^{\gamma_i} (\hat{R}_i - z^*)$  for all  $i \in M$ . Adding this inequality over all  $i \in M$ , we have  $\sum_{i \in M} (V_i^*)^{\gamma_i} (R_i^* - z^*) \leq \sum_{i \in M} \hat{V}_i^{\gamma_i} (\hat{R}_i - z^*)$ . Since  $(p_1^*, \dots, p_m^*)$  is an optimal solution to problem (4.13), we have  $z^* = \sum_{i \in M} (V_i^*)^{\gamma_i} R_i^* / (v_0 + \sum_{i \in M} (V_i^*)^{\gamma_i})$ . Arranging the terms in this equality, it follows that  $v_0 z^* = \sum_{i \in M} (V_i^*)^{\gamma_i} (R_i^* - z^*)$ , in which case, we have  $v_0 z^* = \sum_{i \in M} (V_i^*)^{\gamma_i} (R_i^* - z^*) \leq \sum_{i \in M} \hat{V}_i^{\gamma_i} (\hat{R}_i - z^*)$ . Focusing on the first and last terms in this chain of inequalities and solving for  $z^*$ ,

we get  $z^* \leq \sum_{i \in M} \hat{V}_i^{\gamma_i} \hat{R}_i / (v_0 + \sum_{i \in M} \hat{V}_i^{\gamma_i}) = \Pi(\hat{p}_1, \dots, \hat{p}_m)$ . Thus, the solution  $(\hat{p}_1, \dots, \hat{p}_m)$  provides an expected revenue that is at least as large as the optimal objective value of problem (4.13). In the rest of the proof, we show that  $(\hat{p}_1, \dots, \hat{p}_m)$  is a feasible solution to problem (4.13), which establishes that  $(\hat{p}_1, \dots, \hat{p}_m)$  is an optimal solution to problem (4.13).

Consider nest  $i \in M \setminus \{1\}$ . Noting that  $\hat{p}_i$  is a feasible solution to problem (4.17), we obtain  $\hat{p}_{ij} \geq \ell_i^*$  and  $\hat{p}_{i-1,j} \leq w_{i-1}^*$  for all  $j \in N$ , which imply that  $\min_{j \in N} \hat{p}_{ij} \geq \ell_i^*$  and  $\max_{j \in N} \hat{p}_{i-1,j} \leq w_{i-1}^*$ . Since  $(p_1^*, \dots, p_m^*)$  is a feasible solution to problem (4.13), we also have  $p_i^* \in \mathcal{G}_i(p_{i-1}^*)$ , which implies that  $w_{i-1}^* = \max_{j \in N} p_{i-1,j}^* \leq \min_{j \in N} p_{ij}^* = \ell_i^*$ . Therefore, we obtain  $\max_{j \in N} \hat{p}_{i-1,j} \leq w_{i-1}^* \leq \ell_i^* \leq \min_{j \in N} \hat{p}_{ij}$ . The last inequality shows that  $\hat{p}_i \in \mathcal{G}_i(\hat{p}_{i-1})$ . Since our choice of nest  $i$  is arbitrary, we have  $\hat{p}_i \in \mathcal{G}_i(\hat{p}_{i-1})$  for all  $i \in M \setminus \{1\}$ .

## ASSORTMENT OPTIMIZATION OVER TIME

**5.1 Introduction**

In this chapter we introduce the problem of assortment planning over time. In this problem, we have a sequence of time periods and can only introduce one new product per time step, and we are not allowed to remove products from our assortment that have already been introduced. The goal is to determine which products to introduce, and in what order, so as to maximize the total revenue realized over all the time steps under some choice model.

We show how to give a  $1/2$ -approximation algorithm for this problem under a general choice model, for which the multinomial logit choice model is a special case. We further show a  $(1 - 1/e)$ -approximation algorithm if the revenue function is monotone and submodular. Finally, we show that the problem is NP-hard to compute for a natural special case of the general choice model whose revenue function is monotone and submodular.

**5.2 Literature Review**

In dynamic assortment problems, the offered assortment is adjusted over time, possibly due to depleted product inventories, better understanding of customer choice processes or changes in customer tastes. [28] and [37] study the problem of finding an assortment to offer and the corresponding stocking quantities with the understanding that customers choose only among the products that are still in stock. [3] and [24] consider the problem of dynamically customizing the assortment offerings based on the preferences of each customer and remaining product inventories. [6]

and [9] study assortment problems where the attractiveness of the products diminishes over time and they seek optimal policies to replace such products. [7] and [62] develop models where the assortment offering needs to be adjusted over time in response to a better understanding of the customer choice process.

The rest of the note is organized as follows. §5.3 describes our assortment problem, where the firm needs to gradually build its product portfolio. §5.4 analyzes approximation algorithms for the problem. §5.5 discusses the complexity of the problem.

### 5.3 Preliminaries

Let  $I$  be the set of items that can be offered for sale, and let  $n = |I|$ . We let  $r_j$  be the revenue earned when item  $j$  is sold. Let  $P_j(S)$  be the probability that item  $j$  is purchased if  $S \subseteq I$  is offered for sale; then  $P_j(S) = 0$  if  $j \notin S$ . We consider choice models where the following two properties hold. First,

$$P_j(S) \geq P_j(T) \quad \forall T \forall j \in S \subset T;$$

that is, the probability of purchasing item  $j$  cannot increase if we offer a larger set of items. This holds for utility maximization models, for example. Second,  $\sum_{j \in S} P_j(S) \leq 1$  for any non-empty set of items  $S$ . If an assortment  $S$  is offered for sale, then with probability  $1 - \sum_{j \in S} P_j(S)$ , no item is purchased. The expected revenue for a set  $S$  of items is  $R(S) = \sum_{j \in S} r_j P_j(S)$ . For lack of a better term, let us call such choice models *monotone* choice models. The multinomial logit choice model is one example of a monotone choice model.

Additionally, we will also consider the case in which we only know that the revenue function  $R(S)$  is *monotone* (that is,  $R(S) \leq R(T)$  for any  $S \subseteq T \subseteq I$ ) and *submodular* (that is, for any  $j \notin S \subseteq T \subseteq I$ , then  $R(S \cup \{j\}) - R(S) \geq$

$R(T \cup \{j\}) - R(T)$ ), without knowing anything about the underlying choice model.

We will consider variants of the assortment planning problem in which there is a capacity  $c$  on the number of items that can be offered for sale; that is, we wish to find a set  $S_c^*$  with  $|S_c^*| \leq c$  that maximizes  $R(S_c^*)$ . We call this the *capacitated assortment planning problem*. For an optimal solution  $S_c^*$ , let  $\text{OPT}_c$  be the expected revenue obtained; that is,  $\text{OPT}_c = R(S_c^*)$ . For our first result, we will suppose that we have a polynomial-time algorithm for the capacitated assortment planning problem under the given monotone choice model. For example, there is an optimal algorithm for the problem for the multinomial logit and nested logit choice models (see [16] and [51]). For our second result, we will use a well-known greedy  $(1 - 1/e)$ -approximation algorithm for finding a maximum valued set  $S$  with  $|S| \leq c$  for any monotone, submodular set function due to [45]. The greedy algorithm repeatedly chooses an element to add to the set  $S$  until  $|S| = c$ , and each time adds the element in  $I - S$  that maximizes the marginal gain; that is, it selects  $j \in I - S$  that maximizes  $R(S \cup \{j\}) - R(S)$  and adds it to  $S$ .

We wish to study the *assortment planning problem over time*. Intuitively, we would like to find a sequence of items in which we can offer at most one new item for sale at each of  $T$  time steps that maximizes the overall expected revenue achieved over the given time horizon. Once an item is offered for sale, it remains available to purchase for the remainder of the time horizon. More precisely, we would like to find sets  $S_1, S_2, \dots, S_T$  such that  $|S_t| \leq t$  for all  $1 \leq t \leq T$  and  $S_1 \subseteq S_2 \subseteq \dots \subseteq S_T$  that maximizes  $\sum_{t=1}^T R(S_t)$ .

## 5.4 Algorithm

In this section, we first give an  $1/2$ -approximation algorithm for the assortment planning problem over time for monotone choice models, given a polynomial-time

algorithm for the capacitated assortment planning problem under the choice model. We then give a  $(1 - 1/e)$ -approximation algorithm for the assortment planning problem over time for monotone submodular revenue functions.

The first algorithm works as follows. We initially use the polynomial-time algorithm to find an optimal assortment of at most capacity  $t$  for all  $t$ ,  $1 \leq t \leq T$ . Let  $\tau$  be the capacity for which we get the largest revenue assortment and let  $S_\tau$  be the assortment. We then order the items of  $S_\tau$  by nonincreasing value of  $r_j P_j(S_\tau)$ ; we will offer items for sale in this order. Without loss of generality, assume that items of  $S_\tau$  are indexed by  $1, 2, \dots, \tau$  so that  $r_1 P_1(S_\tau) \geq r_2 P_2(S_\tau) \geq \dots \geq r_\tau P_\tau(S_\tau)$ . Then we set  $S_1 = \{1\}$ ,  $S_2 = \{1, 2\}$ ,  $\dots$ ,  $S_{\tau-1} = \{1, \dots, \tau - 1\}$ , and  $S_t = S_\tau$  for all  $\tau \leq t \leq T$ .

We now analyze the algorithm. Let  $\text{OPT}$  be the overall expected revenue of an optimal assortment over time. We observe the following.

**Observation 5.4.1.**

$$\text{OPT} \leq \sum_{t=1}^T \text{OPT}_t \leq T \cdot \text{OPT}_\tau = T \cdot R(S_\tau^*),$$

where  $\text{OPT}_t = R(S_t^*)$  is the optimal expected revenue for the capacitated assortment planning problem with capacity  $t$ ,  $S_t^*$  is the optimal solution to this capacitated problem, and  $\tau$  is such that  $\text{OPT}_\tau \geq \text{OPT}_t$  for  $1 \leq t \leq T$ .

We then get the following easy lemma bounding the value of the algorithm's solution in terms of the overall optimum.

**Lemma 5.4.2.** *Given a monotone choice model, for any of the sets  $S_t$  constructed above,  $1 \leq t \leq \tau$ , we have that*

$$R(S_t) \geq \frac{t}{\tau} \text{OPT}_\tau.$$

*Proof.* Consider the following inequalities:

$$\begin{aligned}
OPT_\tau &= \sum_{j \in S_\tau} r_j P_j(S_\tau) \\
&= \sum_{j \in S_t} r_j P_j(S_\tau) + \sum_{j \in S_\tau/S_t} r_j P_j(S_\tau) \\
&\leq \sum_{j \in S_t} r_j P_j(S_t) + \sum_{j \in S_\tau/S_t} r_j P_j(S_\tau) \\
&\leq \sum_{j \in S_t} r_j P_j(S_t) + \frac{\tau - t}{\tau} \sum_{j \in S_\tau} r_j P_j(S_\tau) \\
&= R(S_t) + \frac{\tau - t}{\tau} OPT_\tau,
\end{aligned}$$

where the final inequality follows by our choice of the highest revenue items in  $S_\tau$ . Rearranging terms gives the claimed inequality.  $\square$

**Theorem 5.4.3.** *The algorithm is a 1/2-approximation algorithm for the assortment planning problem over time for any monotone choice model such that there is a polynomial-time algorithm for the capacitated assortment planning problem.*

*Proof.* The expected value of our solution is  $\sum_{1 \leq t \leq \tau} R(S_t) + \sum_{\tau < t \leq T} R(S_\tau^*)$ . Then

$$\begin{aligned}
\sum_{1 \leq t \leq \tau} R(S_t) + \sum_{\tau < t \leq T} R(S_\tau^*) &\geq \sum_{1 \leq t \leq \tau} \frac{t}{\tau} R(S_\tau^*) + \sum_{\tau < t \leq T} R(S_\tau^*) \\
&= R(S_\tau^*) \left( T - \tau + 1 + \sum_{1 \leq t \leq \tau} \frac{t}{\tau} \right) \\
&= R(S_\tau^*) \left( T - \tau + \frac{\tau(\tau + 1)}{2\tau} \right) \\
&= R(S_\tau^*) \left( T - \tau + \frac{\tau + 1}{2} \right) \\
&= R(S_\tau^*) \left( T - \frac{\tau}{2} + \frac{1}{2} \right) \\
&\geq \frac{T}{2} \cdot R(S_\tau^*) \geq \frac{1}{2} OPT,
\end{aligned}$$

where the final inequality uses Observation 5.4.1.  $\square$

We now show that we can obtain a  $(1 - 1/e)$ -approximation algorithm for assortment planning over time given that the revenue function  $R$  is monotone and submodular. We simply run the greedy algorithm, and let  $S_1$  be the first item selected by the greedy algorithm,  $S_2$  be the first two elements selected by the greedy algorithm, and so on. If  $T \geq |I|$ , then for time steps  $t \geq T$ , we let  $S_t = I$ . We can now show the following.

**Theorem 5.4.4.** *This algorithm gives a  $(1 - 1/e)$ -approximation algorithm for assortment planning over time when the revenue function is monotone and submodular.*

*Proof.* Let  $S_t^*$  be the optimal assortment of  $t$  items for revenue function, where  $S_t^* = I$  for  $t \geq T$ . Then because the greedy algorithm is a  $(1 - 1/e)$ -approximation algorithm for the capacitated assortment planning problem when the revenue function is monotone and submodular, we know that for any  $t \geq 1$ ,

$$R(S_t) \geq \left(1 - \frac{1}{e}\right) R(S_t^*).$$

Therefore, we have a revenue of

$$\sum_{t=1}^T R(S_t) \geq \left(1 - \frac{1}{e}\right) \sum_{t=1}^T R(S_t^*) \geq \left(1 - \frac{1}{e}\right) \text{OPT},$$

since by Observation 5.4.1  $\sum_{t=1}^T \text{OPT}_t = \sum_{t=1}^T R(S_t^*)$  must be an upper bound on the optimal revenue.  $\square$

## 5.5 Hardness

We now show that assortment planning over time is NP-hard under a particular monotone choice model whose revenue function is monotone and submodular. Our reduction is from the min-sum set cover problem. In this problem, we are given a



hypergraph  $H = (V, E)$  as input, and the output is a sequence of the elements of  $V$ ; we can think of the output as a bijective function  $f : V \rightarrow \{1, \dots, n\}$ , where  $n = |V|$ . We extend the function  $f$  to the hyperedges so that  $f(e) = \min_{v \in e} f(v)$ . Then the goal of the min-sum set cover problem is to find a bijection  $f$  so as to minimize  $\sum_{e \in E} f(e)$ . [15] show that this is an NP-hard problem; in particular, they show that there is no  $(2 - \epsilon)$ -approximation algorithm for the problem, even for  $r$ -uniform  $d$ -regular hypergraphs, unless  $P = NP$ . A hypergraph is  $r$ -uniform if  $|e| = r$  for all  $e \in E$ , and is  $d$ -regular if each vertex  $v \in V$  is in exactly  $d$  of the hyperedges. We will need the  $d$ -regularity for our reduction.

**Theorem 5.5.1.** *It is NP-complete to decide whether the expected revenue of an assortment planning over time instance is at least  $C$  for a choice model that is monotone, or a revenue function that is monotone and submodular.*

*Proof.* Given an instance of the min-sum set cover problem in which we have an  $d$ -regular hypergraph, we create an instance of assortment planning over time as follows. We create an item  $j$  of revenue  $r_j = |E|$  for each vertex  $j \in V$ . We set  $P_j(S) = 0$  if  $j \notin S$ , and otherwise

$$P_j(S) = \frac{1}{|E|} \sum_{e \in E} \frac{|e \cap \{j\}|}{|e \cap S|},$$

where we assume  $0/0 = 0$  in the case that  $e \cap S = \emptyset$ .

We now show that this choice model is monotone. Since there are exactly  $d$  hyperedges that contain  $j$  and  $|E| \geq d$ , then  $0 \leq P_j(S) \leq 1$  for all  $j$  and all  $S$ , and furthermore  $P_j(S) \geq P_j(T)$  when  $j \in S \subset T$ , since for any edge  $e \in E$ , if  $j \notin e$ , then  $|e \cap \{j\}|/|e \cap S| = |e \cap \{j\}|/|e \cap T| = 0$ , while if  $j \in e$ ,  $1/|e \cap S| \geq 1/|e \cap T|$ .

Furthermore, for any nonempty set  $S$ ,

$$\begin{aligned}
\sum_{j \in S} P_j(S) &= \frac{1}{|E|} \sum_{j \in S} \sum_{e \in E} \frac{|e \cap \{j\}|}{|e \cap S|} \\
&= \frac{1}{|E|} \sum_{e \in E} \sum_{j \in S} \frac{|e \cap \{j\}|}{|e \cap S|} \\
&= \frac{1}{|E|} \sum_{e \in E} \frac{|e \cap S|}{|e \cap S|} \\
&= \frac{1}{|E|} \sum_{e \in E} \mathbb{I}(e \cap S),
\end{aligned}$$

where  $\mathbb{I}(e \cap S)$  is the indicator function that is 1 if  $e \cap S \neq \emptyset$  and is 0 otherwise.

Thus  $\sum_{j \in S} P_j(S) \leq 1$ .

We now show that the revenue function  $R(S)$  is monotone and submodular.

From the above, we have that

$$R(S) = |E| \sum_{j \in S} P_j(S) = \sum_{e \in E} \mathbb{I}(e \cap S)$$

so the revenue for a set  $S$  is simply the number of hyperedges with which  $S$  has a nonempty intersection. Clearly this function is monotone. It is also submodular since the number of additional hyperedges intersected by adding a new element  $j$  to  $S$  is at least as large as the number of additional hyperedges intersected by adding  $j$  to  $T \supseteq S$ ; that is, for  $T \supseteq S$ ,

$$\begin{aligned}
R(S \cup \{j\}) - R(S) &= \sum_{e \in E} \mathbb{I}(e \cap (S \cup \{j\})) - \sum_{e \in E} \mathbb{I}(e \cap S) \\
&\geq \sum_{e \in E} \mathbb{I}(e \cap (T \cup \{j\})) - \sum_{e \in E} \mathbb{I}(e \cap T) = R(T \cup \{j\}) - R(T).
\end{aligned}$$

Now suppose we are given some ordering of the vertices given by a function  $f$ . Consider the sets  $S_1 = \{f^{-1}(1)\}$ ,  $S_2 = \{f^{-1}(1), f^{-1}(2)\}$ ,  $\dots$ ,  $S_n =$

$\{f^{-1}(1), \dots, f^{-1}(n)\}$ . Then the same ordering of the items gives a revenue of

$$\begin{aligned}
|E| \sum_{i=1}^n \sum_{j \in S_i} P_j(S_i) &= \sum_{i=1}^n \sum_{j \in S_i} \sum_{e \in E} \frac{|e \cap \{j\}|}{|e \cap S_i|} \\
&= \sum_{e \in E} \sum_{i=1}^n \sum_{j \in S_i} \frac{|e \cap \{j\}|}{|e \cap S_i|} \\
&= \sum_{e \in E} \sum_{i=1}^n \frac{|e \cap S_i|}{|e \cap S_i|} \\
&= \sum_{e \in E} \sum_{i=1}^n \mathbb{I}(e \cap S_i).
\end{aligned}$$

We now observe that  $\sum_{i=1}^n \mathbb{I}(e \cap S_i) + f(e) = n + 1$ , since  $f(e)$  is the smallest  $j$  for which  $e \cap S_j \neq \emptyset$ , while  $\mathbb{I}(e \cap S_i)$  is 1 for  $j \leq i \leq n$  and 0 for  $i < j$ .

Thus for any ordering of the vertices, the sum of the expected revenue plus the sum  $\sum_{e \in E} f(e)$  is

$$|E| \sum_{j \in V} \sum_{i=1}^n P_j(S_i) + \sum_{e \in E} f(e) = \sum_{e \in E} \sum_{i=1}^n (\mathbb{I}(e \cap S_i) + f(e)) = (n + 1)|E|.$$

Hence maximizing revenue for this instance is equivalent to minimizing the min-sum set cover objective function. Therefore, given an instance of the min-sum set cover problem in which we must check if the objective is at most  $B$ , we can reduce it to this instance of assortment planning over time and check if the expected revenue is at least  $(n + 1)|E| - B$ . Therefore, the decision version of our incremental assortment planning problem is also NP-complete.  $\square$

We now say a few words about the particular choice model implied by the probabilities  $P_j(S)$  given above. We can view each edge  $e \in E$  as representing a particular customer type who is solely interested in the items in  $e$  but is indifferent between them. Given a set  $S$  of products, a uniformly random customer type  $e$  arrives, and selects uniformly at random amongst any of the  $e \cap S$  items offered (if there are any). Consider now the capacitated version of this problem: suppose we

want to choose  $S \subseteq V$ ,  $|S| \leq k$ , to maximize  $|E| \sum_{j \in S} P_j(S)$ . From the above, we have that

$$R(S) = |E| \sum_{j \in S} P_j(S) = \sum_{e \in E} \mathbb{I}(e \cap S),$$

so we need to pick  $S$ ,  $|S| \leq k$ , so as to maximize the number of hyperedges  $e$  with which  $S$  has some intersection (we say that  $S$  *covers* a hyperedge  $e$  if  $S \cap e \neq \emptyset$ ). This problem is known as the *maximum coverage problem*, and it is well-known to be an NP-hard problem (see [14]).

## BIBLIOGRAPHY

- [1] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, Inc., New York, second edition, 1993.
- [2] M. Ben-Akiva and S. Lerman. *Discrete Choice Analysis: Theory and Application to Travel Demand*. MIT Press in Transportation Studies, Cambridge, MA, 1994.
- [3] F. Bernstein, A. G. Kok, and L. Xie. Dynamic assortment customization with limited inventories. Technical report, Duke University, Fuqua School of Business, 2011.
- [4] A. Börsch-Supan. On the compatibility of nested logit models with utility maximization. *Journal of Econometrics*, 43:373–388, 1990.
- [5] 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.
- [6] F. Caro, V. Martinez de Albeniz, and P. Rusmevichientong. The assortment packing problem: Multiperiod assortment planning for short-lived products. *Management Science*, (to appear), 2014.
- [7] F. Caro and J. Gallien. Dynamic assortment with demand learning for seasonal consumer goods. *Management Science*, 53(2):276–292, 2007.
- [8] Kyle D. Chen and Warren H. Hausman. Technical note: Mathematical properties of the optimal product line selection problem using choice-based conjoint analysis. *Management Science*, 46(2):327–332, 2000.

- [9] E. Cinar and V. Martinez-de-Albeniz. A closed-loop approach to dynamic assortment planning. Technical report, University of Navara, IESE Business School, 2014.
- [10] J. Davis, G. Gallego, and H. Topaloglu. Assortment planning under the multinomial logit model with totally unimodular constraint structures. Technical report, Cornell University, School of Operations Research and Information Engineering, 2013.  
Available at <http://legacy.orie.cornell.edu/~huseyin/publications/publications...>
- [11] J.M. Davis, G. Gallego, and H. Topaloglu. Assortment optimization under variants of the nested logit model. *Operations Research*, 62(2):250–273, 2014.
- [12] L. Dong, P. Kouvelis, and Z. Tian. Dynamic pricing and inventory control of substitute products. *Manufacturing & Service Operations Management*, 11(2):317–339, 2009.
- [13] Vivek F. Farias, Srikanth Jagabathula, and Devavrat Shah. A non-parametric approach to modeling choice with limited data. *Management Science*, (to appear), 2011.
- [14] Uriel Feige. A threshold of  $\ln n$  for approximating set cover. *Journal of the ACM*, 45:634–652, 1998.
- [15] Uriel Feige, László Lovász, and Prasad Tetali. Approximating min sum set cover. *Algorithmica*, 40:219–234, 2004.
- [16] J. B. Feldman and H. Topaloglu. Capacity constraints across nests in assortment optimization under the nested logit model. Technical report, Cornell University, School of Operations Research and Information Engineering, 2014.

- [17] A. M. Frieze and M. R. B. Clarke. Approximation algorithms for the  $m$ -dimensional 0-1 knapsack problem: Worst-case and probabilistic analyses. *European Journal of Operational Research*, 15:100–109, 1984.
- [18] 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.
- [19] G. Gallego and C. Stefanescu. Upgrades, upsells and pricing in revenue management. Technical report, Columbia University, 2009.
- [20] G. Gallego and H. Topaloglu. Constrained assortment optimization for the nested logit model. Technical report, Cornell University, School of Operations Research and Information Engineering, 2012.  
Available at <http://legacy.orie.cornell.edu/~huseyin/publications/publications...>
- [21] G. Gallego and R. Wang. Multi-product price optimization and competition under the nested attraction model. Technical report, Columbia University, 2011.
- [22] Guillermo Gallego, Richard Ratliff, and Sergey Shebalov. A general attraction model and an efficient formulation for the network revenue management problem. Technical report, Columbia University, New York, NY, 2011.
- [23] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA, 1979.
- [24] N. Golrezaei, H. Nazerzadeh, and P. Rusmevichientong. Real-time optimization of personalized assortments. *Management Science*, (to appear), 2014.
- [25] W. Hanson and K. Martin. Optimizing multinomial logit profit functions. *Management Science*, 42(7):992–1003, 1996.

- [26] I. Heller. On linear systems with integral valued solutions. *Pacific Journal of Mathematics*, 7(3):1351–1364, 1957.
- [27] I. Heller and A. J. Hoffman. On unimodular matrices. *Pacific Journal of Mathematics*, 12(4):1321–1327, 1962.
- [28] Dorothee Honhon, Vishal Gaur, and Sridhar Seshadri. Assortment planning and inventory decisions under stock-out based substitution. *Operations Research*, 58(5):1364–1379, 2010.
- [29] P. W. Keller, R. Levi, and G. Perakis. Efficient formulations for pricing under attraction demand models. *Math. Programming A.*, 145:223–261, 2014.
- [30] A. G. Kok and Y. Xu. Optimal and competitive assortments with endogenous pricing under hierarchical consumer choice models. *Management Science*, 57(9):1546–1563, 2011.
- [31] S. Kunnumkal and H. Topaloglu. A refined deterministic linear program for the network revenue management problem with customer choice behavior. *Naval Research Logistics Quarterly*, 55(6):563–580, 2008.
- [32] Bosang Lee. Calling patterns and usage of residential toll service under self selecting tariffs. *Journal of Regulatory Economics*, 16:45–81, 1999.
- [33] G. Li, P. Rusmevichientong, and H. Topaloglu. The  $d$ -level nested logit model: Assortment and price optimization problems. *Operations Research*, (to appear), 2014.
- [34] 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.



- [35] H. Li and W. T. Huh. Pricing under the nested attraction model with a multi-stage choice structure. In *INFORMS Annual Meeting, Minneapolis, MN*, 2013.
- [36] Qian Liu and Garrett van Ryzin. On the choice-based linear programming model for network revenue management. *Manufacturing & Service Operations Management*, 10(2):288–310, 2008.
- [37] S. Mahajan and G. van Ryzin. Stocking retail assortments under dynamic customer substitution. *Operations Research*, 49(3):334–351, 2001.
- [38] Daniel McFadden. Conditional logit analysis of qualitative choice behavior. In P. Zarembka, editor, *Frontiers in Economics*, pages 105–142. Academic Press, 1974.
- [39] Daniel McFadden. Econometric models for probabilistic choice among products. *The Journal of Business*, 53(3):S13–29, 1980.
- [40] Daniel McFadden. Econometric models of probabilistic choice. In Charles F. Manski and Daniel McFadden, editors, *Structural Analysis of Discrete Data with Economic Applications*, pages 198–272. MIT Press, Cambridge, MA, 1981.
- [41] J. Meissner and A. K. Strauss. Network revenue management with inventory-sensitive bid prices and customer choice. *European Journal of Operational Research*, 216:459–468, 2012.
- [42] Joern Meissner, Arne Strauss, and Kalyan Talluri. An enhanced concave program relaxation for choice network revenue management. *Production and Operations Management*, (to appear), 2012.

- [43] 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.
- [44] G. Nemhauser and L. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, Inc., Chichester, 1988.
- [45] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions — I. *Mathematical Programming*, 14:265–294, 1978.
- [46] M. L. Puterman. *Markov Decision Processes*. John Wiley and Sons, Inc., New York, 1994.
- [47] Z. Rayfield, P. Rusmevichientong, and H. Topaloglu. Approximation methods for pricing problems under the nested logit model with price bounds. *INFORMS Journal on Computing*, (to appear), 2013.
- [48] P. Rusmevichientong, B. Van Roy, and P. Glynn. A nonparametric approach to multiproduct pricing. *Operations Research*, 54(1):82–98, 2006.
- [49] P. Rusmevichientong, Z.-J. M. Shen, and D. B. Shmoys. A PTAS for capacitated sum-of-ratios optimization. *Operations Research Letters*, 37(4):230–238, 2009.
- [50] 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.
- [51] 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.

- [52] Paat Rusmevichientong, David B. Shmoys, and Huseyin Topaloglu. Assortment optimization with mixtures of logits. Technical report, Cornell University, School of Operations Research and Information Engineering, 2010. Available at <http://legacy.orie.cornell.edu/~huseyin/publications/publications>.
- [53] Paat Rusmevichientong and Huseyin Topaloglu. Robust assortment optimization in revenue management under the multinomial logit choice model. Technical report, Cornell University, School of Operations Research and Information Engineering, 2011. Available at <http://legacy.orie.cornell.edu/~huseyin/publications/publications>.
- [54] A. Ruszczyński. *Nonlinear Optimization*. Princeton University Press, Princeton, New Jersey, 2006.
- [55] J.-S. Song and Z. Xue. Demand management and inventory control for substitutable products. Technical report, Duke University, Durham, NC, 2007.
- [56] K. Talluri. A randomized concave programming method for choice network revenue management. Technical report, Universitat Pompeu Fabra, Barcelona, Spain, 2011.
- [57] K. Talluri and G. van Ryzin. Revenue management under a general discrete choice model of consumer behavior. *Management Science*, 50(1):15–33, 2004.
- [58] Piyush Tiwari and Hiroshi Hasegawa. Demand for housing in Tokyo: A discrete choice analysis. *Regional Studies*, 38(1):27–42, 2004.
- [59] Kenneth Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, Cambridge, UK, 2003.
- [60] Kenneth E. Train, Moshe Ben-Akiva, and Terry Atherton. Patterns and self-selecting tariffs. *The Review of Economics and Statistics*, 71(1):62–73, 1989.

- [61] Kenneth E. Train, Daniel L. McFadden, and Moshe Ben-Akiva. The demand for local telephone service: A fully discrete model of residential calling patterns and service choices. *The RAND Journal of Economics*, 18(1):109–123, 1987.
- [62] C. Ulu, D. Honhon, and A. Alptekinoglu. Learning consumer tastes through dynamic assortments. *Operations Research*, 60(4):833–849, 2012.
- [63] R. Wang. Capacitated assortment and price optimization under the multinomial logit model. *Operations Research Letters*, 40:492–497, 2012.
- [64] H. C. W. L. Williams. On the formation of travel demand models and economic evaluation measures of user benefit. *Environment and Planning*, 9(3):285–344, 1977.
- [65] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, New York, 2011.
- [66] Judith Yates and Daniel F. Mackay. Discrete choice modeling of urban housing markets: A critical review and an application. *Urban Studies*, 43(3):559–581, 2006.
- [67] D. Zhang and D. Adelman. An approximate dynamic programming approach to network revenue management with customer choice. *Transportation Science*, 42(3):381–394, 2009.
- [68] Dan Zhang and Zhaosong Lu. Assessing the value of dynamic pricing in network revenue management. *INFORMS Journal on Computing*, 25(1):102–115, 2013.