NEW PERSPECTIVES ON INCORPORATING CUSTOMER CHOICE INTO REVENUE MANAGEMENT

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

Jacob Belinkoff Feldman

August 2015

NEW PERSPECTIVES ON INCORPORATING CUSTOMER CHOICE INTO REVENUE MANAGEMENT

Jacob Belinkoff Feldman, Ph.D.

Cornell University 2015

In this thesis, we consider assortment optimization problems under a variety of customer choice models. In these assortment problems, a retailer has access to a collection of products and she must decide which set of products to make available for purchase with the goal of maximizing the expected revenue derived from each arriving customer. We consider assortment problems when customers choose according to the Markov chain, nested logit and mixtures of multinomial logit choice models.

Biographical Sketch

Jake grew up in Bethesda, Maryland. He loves sports, food and comedy.

To never losing your dinosaur.

Acknowledgements

A big thanks to my advisor Huseyin Topaloglu who was a great advisor. I'd also like to thank my dad, Sol Feldman, for teaching me to always bring a note of levity to everything I do and also to my annoying, yet loving Jewish mother.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The term Big Data has become commonplace, a colloquialism used by mathematicians, businessmen, and even lawyers alike. Its use is ubiquitous, but nonetheless, it is still difficult to find consistency with regards to how its central principles and techniques are applied. Ask fifty different people and you would most likely get fifty different answers regarding its definition and practices. While at its core Big Data is difficult to define, it is far easier to understand its motivation and to see the benefits of its end goals. More data means more information, and hence the opportunity to better understand how people think and make decisions. More importantly, once models are in place and fitted to the underlying data, the goal shifts towards figuring out how to use these models to make profitable managerial decisions.

This thesis focuses on this latter goal as it relates to retailers such as Walmart or Amazon, both whose client lists contain millions of names and whose warehouses contain thousands or even millions of products. More generally, we consider the

essential question that arises in every retail setting, namely, which subset of the available products should the retailer offer to her customers so as to maximize revenue? This problem of finding the optimal subset or assortment of products that yields the largest expected revenue is commonly referred to as the assortment optimization problem. The central difficulty in this problem is that the retailer must carefully balance the appeal of her assortment as a whole, and the relative appeal of the individual products which are the most profitable: adding a product to an assortment diversifies a retailer's display and thus increases her market share, but this additional product can also cannibalize the sales of other more profitable items.

The collection of products available to the retailer can be thought of as physical items such as electronic equipment or textbooks, or virtual products such as reservations or appointments. Consequently, assortment problems are relevant in a variety of retail and service industries. Most hotels and airlines, for example, have developed online channels through which customers can book hotels or flights. As a result, these businesses face assortment decisions regarding which sets of rooms or travel plans to make available for bookings. The assortment problem faced by classic brick and mortar retailers like Walmart is fairly easy to imagine: limited by space, these retailers must choose the most profitable set of products to display on their shelves. When the context is shifted to online retailers such as Amazon, a variety of new assortment problems arise and the problem scale becomes much larger. The notion of shelf space is replaced by web page displays, and the assortment decisions involve a greater variety of products.

Inherent to any assortment optimization problem is an assumption on how customers make purchasing decisions. Within the revenue management literature, the

mechanisms put in place to govern these purchasing decisions are known as customer choice models. A variety of customer choice models exist, each nuanced to capture different aspects of customer purchasing behavior. As input, the customer choice models take any assortment of products that the retailer could potentially offer, and as output they give the probability that each of the products in the particular assortment is purchased. Particularly noteworthy is the ability of customer choice models to capture substitution behavior. This phenomenon occurs when a customer's preferred product is unavailable due to stock-outs or deliberate inventory control. In these situations, the customer has to make a choice between either settling on a suitable alternative or leaving the store without making a purchase. This thesis studies assortment optimization and revenue management problems under the Markov chain, the nested logit and a mixture of multinomial Logit (MMNL) choice models.

In the next section, we detail the dynamics of the choice models that we study and provide a brief background on the estimation procedures used to estimate the various parameters of the presumed choice model from historical sales data. The estimation problem is an important prerequisite to the assortment problem, as the retailer cannot identify profitable assortments without an accurate characterization of the buying patterns of the underlying customer population. As a result, it is difficult to justify the use of any choice model for which the assortment and estimation problems are not both well understood.

1.1 Customer Choice Models

To formally describe the dynamics of each customer choice model, we assume that the retailer has access to a collection of n products indexed by the set $N = \{1, \ldots, n\}$. In addition to these n products, there is a dummy product which we refer to as the no purchase option and which represents the option for the customer to leave the store without purchasing anything. Naturally, we assume that the no purchase option is included in every assortment, since the customer should always have the option to leave the store without buying anything. We let $P_i(S)$ be the probability that a customer purchases product i given assortment decision $S \subseteq N$. The structure of the purchase probabilities $P_i(S)$ depends on the customer choice model used to model customer choice behavior. Further, it is the structure of these underlying purchase probabilities that directly determines the difficulty of the assortment optimization problem at hand. In what follows, we describe the multinomial logit (MNL), the nested logit (NL) and the Markov chain choice models.

1.1.1 Multinomial Logit Choice Model

The MNL choice model is both simple and powerful, and as a result, it is often used as a benchmark for other choice models (See [6], [1], [15]). It was first introduced by [37] and later shown by [25] to be consistent with the random utility maximization (RUM) principle. Under the RUM principle, a customer associates a random utility with each of the available products and then purchases the offered product with the largest utility. The RUM principle lays the foundation for each of the choice

models we discuss and also gives a practically sound framework for their origins. After all, the MNL choice model is just that, a mathematical model, and while it is hard to imagine a choice model that perfectly captures the psychology of a customer, it helps to build such models on top of a notion that is very intuitive: we choose to buy the product that brings us the largest utility.

Under the MNL choice model, the utilities of each product $j \in N$ are given by the expression $U_i = w_i + \epsilon_i$; where w_i is a fixed component known to the retailer and ϵ_j is the random component that is Gumbel distributed with zero mean. Stemming from the RUM principle, we have that

$$P_i(S) = P(U_i = \max_{j \in S} U_j) = \frac{e^{w_i}}{\sum_{j \in S} e^{w_j}},$$

where the second inequality results from the utilities coming from independent Gumbel distributions. It is common to set $v_i = e^{w_i}$ and to normalize the mean utility of the no purchase option to zero. The term v_i is often referred to as the preference weight for product i. Under this transformation, we get the most recognizable expression for the purchase probabilities under the MNL choice model:

$$P_i(S) = \frac{v_i}{1 + \sum_{j \in S} v_j}.$$

There are many advantages to using the MNL choice model. First, it is possible to express the mean utilities w_j as a linear combination of product features such as price, model and make. As a result, when variations among consumer tastes are largely described by observable features of the products, the MNL choice model can accurately capture consumer purchasing patterns. However, when tastes vary due to unobservable features intrinsic to the individual customers, i.e. style preferences, the MNL can fall short in its modeling ability. The MNL model assumes that the average valuation of each product is the same for every arriving customer and so it

cannot capture inherent segmentations in the customer population. For example, under the MNL choice model, the price sensitivity of both students and working professionals is captured through a single parameter for each product. In reality, students are far more price sensitive and the expressions for the respective utilities should reflect this. The MMNL choice model, which is described in detail in Chapter 4 is a simple extension of MNL model and can be used to overcome this deficiency. Under this choice model, customers of different segments choose according to different multinomial logit models whose preference weights depend on the proclivities of the given customer type.

A second key feature of the MNL choice model is that it possible to estimate the preference weights from historical sales data. The log likelihood terms out to be concave, see [25], and thus any standard optimization software package can be used to derive estimates for the preference weights using maximum likelihood estimation. The estimation problem for the MMNL problem is bit more complex because the specific type of each arriving customer is unobservable to the retailer. As a result, Expectation-Maximization algorithms have been applied to uncensor the type of each arriving customer and find their respective vector of preference weights, see [15].

The major criticism of the MNL choice model is that it exhibits the so-called independence of irrelevant alternatives (IIA), which states that the ratio of purchase probabilities of two products does not depend on what assortment is offered. Interpreted through the lens of a manager, this mathematical property results in proportional substitution across all products as the set of offered products changes. For example, consider the scenario where Gap adds a blue short sleeve t-shirt to its assortment of shirts that it offers at one of its stores. Under the MNL choice

model, the addition of this short sleeve blue t-shirt will affect the purchase probabilities for light blue short sleeve t-shirts and white long sleeve t-shirts in the same proportion. In reality, the purchase probabilities for the light blue short sleeve t-shirts should be affected more dramatically by this change since customers are more likely to substitute within product categories than across product categories. The IIA property is a consequence of the fact that the utilities of each product are independent and identically distributed random variables under the MNL choice. Next, we describe the NL choice model which avoids the IIA property by allowing for correlations between the utilities of each product.

1.1.2 Nested Logit Choice Model

Under the NL choice model, products are grouped into categories or nests based on product attributes. The choice process for each customer can be described as a two stage process where the customer first chooses a nest, and then chooses a product from within this nest. The NL choice model is a generalization of the MNL choice model which avoids the IIA property by introducing correlations between the utilities of products grouped in the same nest. Since the derivation of the NL choice model from RUM principle is not as straightforward as that of the MNL choice model, we leave it out for brevity and point the reader to [26] for the full derivation.

In order to formally give expression for the purchase probabilities under the NL choice model, we assume there are m nests indexed by the set $M = \{1, \ldots, m\}$ and within each nest there are n products indexed by the set $N = \{1, \ldots, n\}$. The preference weight associated with product j from nest i is given by v_{ij}. Furthermore, we assume that the customer associates a preference weight of v_0 with

the option of not purchasing anything. Given offer set $S_i \subseteq N$ for nest i, we let $V_i(S_i) = \sum_{j \in S_i} v_{ij}$ denote the sum of the of the preference weights of all available products within this nest. Each nest has an associated parameter $\gamma_i \in [0, 1]$ which captures the degree of the dissimilarity of the products in the nest. Larger values of γ_i represent smaller amounts of correlation between the utilities of the products in nest i. If we offer the assortment $(S_1, \ldots, S_m)$ over all nests, then a customer chooses nest i with probability

$$Q_i(S_1, \ldots, S_m) = \frac{V_i(S_i)^{\gamma_i}}{v_0 + \sum_{j \in M} V_j(S_j)^{\gamma_j}}.$$

Once the customer chooses a nest, he then chooses among the nest's offerings according to the MNL choice model. Conditioned on selecting nest i, the probability that a customer purchases product j is given by

$$P_j(S_i | \text{Chooses nest } i) = \frac{v_{ij}}{V_i(S_i)}.$$

Therefore, for fixed assortments $(S_1, \ldots, S_m)$ across all nests, we can compute the probability that a customer purchases product j from nest i as

$$P_{ij}(S_1, \ldots, S_m) = \frac{V_i(S_i)^{\gamma_i}}{v_0 + \sum_{j \in M} V_j(S_j)^{\gamma_j}} \frac{v_{ij}}{V_i(S_i)}.$$

We point the reader to [26] for the derivation of the NL model through the RUM principle.

The main advantage of the NL choice model is that it allows us to paint a more accurate picture of customer substitution pattern since it does not suffer from the IIA property. This increase in modeling flexibility comes at the cost of a more complex expression for the purchase probabilities that makes both the assortment optimization problem and the estimation problem more difficult. The assortment problem is addressed in Chapter 3, while [26] details an estimation procedure for the NL choice model. The difficulty in estimating the parameters of the NL

choice model comes about because the likelihood is not jointly concave in both the nest dissimilarity parameters γ_i and the preference weights v_{ij}. As a result, [26] recommend an iterative approach, which alternates between optimizing over the preference weights and dissimilarity parameters with the other set of parameters held fixed. These two optimization problems turn out to have concave objectives and thus this approach is guaranteed to converge to a local maximum.

1.1.3 Markov Chain Choice model

The Markov chain choice model was only recently introduced in [1]. Under this choice model, the states of the Markov chain correspond to products, and the transition probabilities capture substitution behavior. Each arriving customer begins in the state corresponding to his or her most preferred product. There is a distribution over the first choice products of the customer population. If this preferred products is not available, the customer transitions to another product according to the transition probabilities of the underlying Markov chain. The customer continues to transition until he or she either arrives at the state of an offered product, at which point the customer purchases this products, or transitions to the state corresponding to the no purchase option, at which point the customer leaves the store without purchasing anything. The purchase probabilities are therefore exactly equal to the absorption probabilities of each state. We delay a complete formulation of the purchase probabilities until Chapter 2, where all of our work surrounding this choice model is detailed. We derive the expressions for the purchase probabilities in a slightly different manner than [1] in an effort to provide a bit more intuition regarding these key terms.

In [1], the authors introduce the Markov chain choice model and emphasize its

importance by showing that it subsumes the MNL choice models in addition to accurately approximating a variety of other choice models. The authors do not interpret this choice model within the RUM framework and they provide little guidance with regards to estimating the parameters of the choice model. In Chapter 2, in addition to considering a variety of assortment optimization problems, we also show that the Markov chain choice model has a RUM interpretation and give an estimation procedure that hints at the potential power of this new model.

1.2 Assortment Optimization Problems

There are two varieties of assortment optimization problems: static and dynamic. In the static problem, a retailer offers a single assortment with the goal of maximizing expected revenue extracted from each customer. In the dynamic setting, a retailer varies her assortments over time to account for changing demands or depleting inventories, with the intention of maximizing expected revenue over the selling horizon. For the remainder of this summary, we refer to the static problem simply as an assortment problem and the dynamic problem as a revenue management problem. Whether the retailer operates a brick and mortar store or an online store plays a central role in determining which problem setting is more applicable, as it is often more difficult for a retailer selling physical products to vary her assortment over time. On the one hand, Best Buy generally fixes its floor display of televisions since switching between assortments requires physical moving televisions from the warehouse to the showroom. On the other hand, an airline or online retailer can vary the set of itineraries or products they offer with the click of a button. This makes revenue management problems more amenable to an online

retail setting.

1.2.1 Assortment Problem

In the assortment problem, the retailer must choose a single assortment to offer with the intention of maximizing expected revenue extracted from each customer. More formally, we assume that a retailer has access to n products indexed by the set $N = \{1, \ldots, n\}$, where the revenue for product i is given by r_i. There is also the no purchase option which is assumed to be included in every assortment the retailer offers. The assortment problem can be formulated as follows:

$$\max_{S \subseteq N} P_i(S) r_i.$$

Since the number of possible assortments that the retailer could offer grows exponentially in n, enumeration tactics quickly become intractable, and as a result, more efficient algorithms must be developed.

In Chapters 2 and 4 we consider the assortment problem when customer choices are governed by the Markov chain and MMNL models respectively. In Chapter 3, we consider two variants of the assortment problem under the NL choice model. First, we impose a cardinality constraint on the offered assortment which states that the total number of products that can be offered across all nests must be fewer than c. Formally, if S_i is the assortment of products offered in nest i, we must choose assortments $(S_1, \ldots, S_m)$ over all nests that maximizes expected revenue and satisfies $\sum_{i \in M} |S_i| \leq c$. In the second type of constraint, we let w_{ij} be the space requirement of product j in nest i and limit the total space requirement of the products offered over all nests to c. Again, our goal is to choose an assortment to offer in each nest that maximizes expected revenue, but in this scenario, we

must abide by the constraint $\sum_{i \in M} \sum_{j \in S_i} w_{ij} \leq c$.

1.2.2 Revenue Management Problems

In Chapter 2, in addition to considering the assortment problem, we also consider two related revenue management problems when customers choose according to the Markov chain choice model. In the first problem, a retailer sells multiple products which all consume a single resource. We refer to this problem as the single resource revenue management problem. A popular example of this scenario is an airline selling seats on a single flight leg at multiple fare classes. There is a fixed selling horizon consisting of T time periods where we assume, without loss of generality, that there is a exactly one customer arrival in every time period. Let C be the initial capacity of the resource available to the retailer, and let the set $N = \{1, \ldots, n\}$ index the set of products available to the retailer. Again, there is an ever-present no purchase option. The sale of a product $i \in N$ brings a revenue of r_i and reduces the inventory of the resource by one. For each customer that arrives, the retailer must choose a set of products $S \subseteq N$ to make available for purchase so as to maximize expected revenue over the selling horizon. The single resource revenue management problem can be solved optimally via a dynamic program. Define the value functions $V_t(x)$ to be the optimal expected revenue generated from having x units of inventory remaining at time period t. The Bellman recursion for the single leg problem can be formulated as follows:

$$V_t(x) = \max_{S \subseteq N} \Big\{ \sum_{i \in S} P_i(S)[r_i + V_{t+1}(x-1)] + (1 - \sum_{i \in S} P_i(S)) V_{t+1}(x) \Big\}$$

The boundary conditions are:

$$V_{T+1}(x) = 0 \quad \forall\, x \in \{1, \ldots, C\} \quad \text{and} \quad V_t(0) = 0 \quad \forall\, t \in \{1, \ldots, T\}.$$

Even though the above dynamic program only requires keeping track of a single state variable, it still can be quite computationally burdensome to solve optimally since at each time period t and inventory level x we are required to solve an optimization problem over all possible assortments of products. In Chapter 2, we are able to exploit the special structure of the Markov chain choice model to give efficient methods to solve the dynamic programing formulation of the single resource revenue management problem. Specifically, for each time period t, we are able to compute protection levels that represent the smallest inventory level for which product $j \in N$ is included in the optimal assortment. The protection levels can be computed a priori and provide an easily implementable policy that scales linearly in the number of products.

The second problem, which we refer to as the network revenue management problem, generalizes the first. In this second revenue management problem, a retailer has access to a collection of products, each which consume different amounts of a set of resources. During each time period of the selling horizon, the retailer must decide which products to make available for purchase subject to the remaining capacities of each resource. The retailer makes its offer decision with the intention of maximizing its expected revenue over the selling horizon. This problem setting captures the scenario where an airline must make offer decisions regarding a collection of itineraries, each comprised of a set of flight legs, over a selling horizon.

Let the set $M = \{1, \ldots, m\}$ index the set of resources and the set $N = \{1, \ldots, n\}$ index the set of products. Additionally, there is a no purchase option that is included in every assortment. As mentioned previously, each product $j \in N$ is made up pre-specified number of resources. The relationship between each product and its resources is captured in the incidence matrix $A \in \{0, 1\}^{m \times n}$,

where entry $a_{q,j} \in A$ is 1 if product j incorporates resource q and 0 otherwise. We can characterize each product $j \in N$ by the set of resources $\{q : a_{q,j} = 1\}$ that make it up and the revenue r_j that it generates through a single sale. Let A^j be the j^{th} column of A. If the vector $x \in \mathbb{R}^m$ gives the remaining capacity on each flight leg, then the sale of a single product j reduces the remaining inventory of each resource to $x - A^j$. Let c_q be the initial capacities of each resource. Similar to the single leg problem, we discretize the selling horizon into T time periods indexed by $\{1, \ldots, T\}$.

The network revenue management problem can also be solved optimally using dynamic programming . Define the value functions $V_t(x)$ to be the optimal expected revenue generated from having x units of inventory of each resource remaining at time period t. The Bellman recursion for the network revenue management can be formulated as follows:

$$V_t(x) = \max_{S \subseteq \bar{N}} \Big\{ \sum_{j \in S} P_j(S)[r_j + V_{t+1}(x - A^j)] + (1 - \sum_{j \in S} P_i(S)) V_{t+1}(x) \Big\},$$

where $\bar{N} = \{j \in N : A^j \leq x\}$ and the $\leq$ is component wise comparison. The boundary conditions are:

$$V_{T+1}(x) = 0 \quad \forall\, x \in \{1, \ldots, C\} \quad \text{and} \quad V_t(0) = 0 \quad \forall\, t \in \{1, \ldots, T\}.$$

For large networks, computing the exact value function proves to be intractable since the size of the state space grows exponentially in the number of resources. As a result, in Chapter 2, we resort to a well known deterministic approximation where the choices of the customers take on their expected values. This deterministic approximation comes in the form of a linear program where the decision variables represent the fraction of time during which each subset of products is offered over the selling horizon. Consequently, the number of decision variables increases exponentially with the number of products, and thus the linear program

remains difficult to solve for large flight networks. We show that if customers choose according to the Markov chain choice model, then the linear program with an exponential number of decision variables can be formulated equivalently as a linear program where the number of decision variables grows linearly in the number of products.

1.3 Summaries of Main Results

In this section, we summarize the main results that make up the individual chapters of this thesis. Provided below are brief descriptions of each problem and our main findings.

1.3.1 Revenue Management under the Markov Chain Choice Model

In Chapter 2, we consider both assortment and revenue management problems when customers choose among the offered products according to the Markov chain choice model. Under the Markov chain choice model, a customer arriving into the system considers purchasing a product with a certain probability. If this product is available for purchase, then the customer purchases it. Otherwise, the customer transitions to another product according to the transition matrix of the underlying Markov chain. The customer continues to transition until she reaches either an offered product, or the no purchase option. We give novel solution approaches, when customers choose under the Markov chain choice model, for three fundamental

classes of problems. Specifically, we consider assortment optimization problems, revenue management problems with a single resource, and revenue management problems over a network of resources.

First, we consider the assortment problems. In the assortment setting, there is a revenue for each product. Customers choose among the offered products according to the Markov chain choice model. We want to offer a set of products to maximize the expected revenue from each customer. We relate the probability of purchasing each product under the Markov chain choice model to the extreme points of a polyhedron. Using this result, we show that the optimal set of products to offer can be obtained by a linear program. Also, we prove that as the revenues of the products increase by the same amount, the optimal subset to offer becomes larger. This property becomes critical when we study the optimal policy for the single resource revenue management problem.

Second, we consider revenue management problems with a single resource. In this setting, we need to decide which set of products to make available to customers dynamically over a selling horizon. At each time period, an arriving customer chooses among the set of available products. There is a limited inventory of the resource and the sale of a product consumes one unit of the resource. The goal is to find a policy to decide which set of products to make available at each time period in the selling horizon so as to maximize the total expected revenue. Assuming that customers choose under the Markov chain choice model, we formulate the problem as a dynamic program and show that the optimal policy offers a larger set of products as we have more remaining capacity at a certain time period or as we get closer to the end of the selling horizon, all else being equal. In other words, as we have more capacity or as we get closer to the end of the selling

horizon, the urgency to liquidate the resource inventory takes over and we offer a larger set of products. Using these properties, we show that the optimal policy can be implemented by associating a protection level with each product so that a product is made available to the customers whenever the remaining inventory of the resource exceeds the protection level of the product.

Third, we consider revenue management problems over a network of resources. In the network revenue management setting, we have a number of resources with limited inventories and each product consumes a certain combination of resources. We need to decide which set of products to make available dynamically over a selling horizon. At each time period, an arriving customer chooses among the set of available products according to the Markov chain choice model. The goal is to find a policy to decide which set of products to make available at each time period in the selling horizon so as to maximize the total expected revenue. We can formulate the network revenue management problem as a dynamic program, but this dynamic program requires keeping track of the remaining inventory for all resources, so it can be difficult to solve. Instead, we focus on a deterministic linear programming approximation formulated under the assumption that the customer choices take on their expected values. In this linear program, there is one decision variable for each subset of products, which corresponds to the frequency with which we offer a subset of products to customers. Consequently, the number of decision variables increases exponentially with the number of products and the deterministic linear program is solved by using column generation.

Focusing on the deterministic linear program, we show that if the customers choose according to the Markov chain choice model, then the deterministic linear program can immediately be reduced to an equivalent linear program whose num-

bers of decision variables and constraints increase only linearly with the numbers of products and resources. We develop an algorithm to recover the optimal solution to the original deterministic linear program by using the optimal solution to the reduced linear program. This algorithm allows us to recover the frequency with which we should offer each subset of products to customers by using the optimal solution to the reduced linear program. Our computational experiments show that using the reduced linear program can provide substantial computational savings over solving the original deterministic linear program through column generation.

In addition to considering these three assortment optimization problems, we also give evidence to support the Markov chain choice model as a valuable addition to the choice modeling literature. First, we show that the Markov chain choice model can be derived from the RUM principle by relating it to the nonparametric choice model of [6]. We also give a maximum likelihood estimation procedure for estimating the parameters of the Markov chain choice model from historical sales data. We compare the maximum likelihood estimates for the parameters of both the MNL and Markov chain choice models on two metrics: out of sample of likelihoods and ability to identify profitable assortments. With regards to both metrics, the Markov chain choice model proves more effective at capturing customer purchasing patterns.

1.3.2 Capacity Constraints across Nests in Assortment Optimization under the Nested Logit Model

In Chapter 3, we consider the assortment problem when customers choose according to the NL model and there is limited capacity for the products in the offered

assortment. Under the NL model, the products are organized in nests. Each customer, after viewing the offered assortment, decides either to make a purchase within one of the nests or to leave the system without purchasing anything. If a nest is chosen, then the customer purchases one of the products within the chosen nest. In our problem setup, there is a capacity constraint limiting the total capacity consumption of the products in the offered assortment. The goal is to choose an assortment of products to offer so as to maximize the expected revenue obtained from each customer. We consider two types of capacity constraints. In the first type, each product occupies one unit of space, in which case the capacity constraint limits the total number of products in the offered assortment. We refer to this type of capacity constraint as a cardinality constraint. In the second type of constraint, the capacity consumption of a product is arbitrary, possibly reflecting the space or capital requirement of a product. We refer to this type of capacity constraint as a space constraint.

Under a cardinality constraint, we show that we can obtain an optimal assortment by solving a linear program with $O(m^2n)$ decision variables and $O(m^2n^4)$ constraints, where m is the number of nests and n is the number of products in each nest. As far as we are aware, the assortment problem was not known to be tractable when customers choose according to the NL model and there is a cardinality constraint limiting the total number of products in the offered assortment. This result gives the first exact solution method for this problem. We also show, under a space constraint, that we can obtain a 4-approximate solution by solving a linear program with $O(m)$ decision variables and $O(mn^4)$ constraints. The running time of this algorithm scales polynomially with the number of products and the number of nests. To our knowledge, this result gives the first algorithm for the assortment problem that scales polynomially with the number of nests, when there is a ca-

pacity constraint on the space consumption of all offered products and customers choose according to the NL model.

We also show that we can compute an upper bound on the optimal expected revenue for an individual problem instance by solving a linear program. In our numerical experiments, we consider problem instances involving products with arbitrary capacity consumptions. Comparing the expected revenues from the assortments obtained by our 4-approximation algorithm with the upper bounds on the optimal expected revenues, our numerical results indicate that the 4-approximation algorithm performs quite well, yielding less than 2% optimality gap on average.

1.3.3 Bounding Optimal Expected Revenues for Assortment Optimization under a Mixture of Multinomial Logit Models

In Chapter 4, we study the assortment problem when customer purchasing patterns are governed by a MMNL choice model. In our problem setting, a firm wants to find a set of products to offer to its customers. There is a fixed revenue associated with each product. An arriving customer may be one of multiple customer types. The firm does not know the type of an arriving customer, but it has access to the probability that an arriving customer is of a particular type. Customers of different types choose according to different multinomial logit models whose parameters depend on the proclivities of the customer. This choice model is known as the MMNL model. The goal of the firm is to find an assortment of products to offer to its customers so as to maximize the expected revenue obtained from each customer. The assortment problem is known to be NP-complete when customers

choose according to the MMNL choice model, although it is well known that greedy heuristics perform quite well. One shortcoming of using a heuristic is that there is no immediate way of being confident that the solution provided by a heuristic is actually a good one.

In this chapter, motivated by the difficulty of obtaining optimal solutions and evaluating the quality of the solutions provided by a heuristic, we develop a method to obtain upper bounds on the optimal expected revenue in our assortment problem. We then use these upper bounds to get a measure of the gap between the expected revenue from the solution provided by a heuristic and the optimal expected revenue.

The main combinatorial difficulty of this problem arises because the offer decisions for each product are coupled across customer types. Specifically, this means that a product offered to one customer type must be offered to every customer type. One way to achieve an upper bound on the optimal revenue is to relax the problem by allowing the retailer to offer separate assortments to each customer type. This approach decomposes the problem into a collection of disjoint MNL assortment problems whose optimal assortments can each be found easily. Namely, it is well known that the optimal offer set for the single class assortment problem is some collection of the highest revenue items. This customer type decomposition method not only serves as a benchmark upper bound upon which we hope to improve, but also motivates our upper bound technique.

Our approach expands on this customer type decomposition idea by introducing penalty costs when assortments differ across customer types. We solve a convex optimization to find the penalty costs which give us the tightest upper bounds. We test our upper bound technique on a variety of test problems where the het-

erogeneity of the purchasing preferences across the customer population is varied. Our computational experiments yield upper bounds that are within 0.83% of the greedy heuristic solution for each of the 27,000 test cases. This proves to be a significant improvement over the customer type decomposition approach.

Chapter 2

Revenue Management under the Markov Chain Choice Model

Incorporating customer choice behavior into revenue management models has been seeing increased attention. Traditional revenue management models assume that each customer arrives into the system with the intention of purchasing a certain product. If this product is available for purchase, then the customer purchases it. Otherwise, the customer leaves without a purchase. In reality, however, there may be multiple products that serve the needs of a customer and a customer may observe the set of available products and make a choice among them. This type of customer choice behavior is even more prevalent today with the common use of online sales channels that conveniently bring a variety of options to customers. When customers choose among the available products, the demand for a particular product naturally depends on what other products are made available to the customers, creating interactions between the demands for the different products. When such

interactions exist between the demands for the different products, finding the right set of products to offer to customers can be a challenging task.

In this chapter, we consider revenue management problems when customers choose among the offered products according to the Markov chain choice model. In the Markov chain choice model, a customer arrives into the system to purchase a particular product. If this product is available for purchase, then the customer purchases it. Otherwise, the customer transitions to another product with a certain probability and checks the availability of the other product, or she transitions to the no purchase option with a certain probability and leaves the system without a purchase. In this way, the customer transitions between the products until she reaches a product available for purchase or she reaches the no purchase option. We consider three fundamental classes of revenue management problems when customers choose under the Markov chain choice model. In particular, we consider assortment optimization problems, revenue management problems with a single resource and revenue management problems over a network of resources. We proceed to describing our contributions to these three classes of problems.

Contributions. First, we consider assortment problems. In the assortment setting, there is a revenue for each product. Customers choose among the offered products according to the Markov chain choice model. We want to offer a set of products to maximize the expected revenue from each customer. We relate the probability of purchasing each product under the Markov chain choice model to the extreme points of a polyhedron (Lemma 2). Using this result, we show that the optimal set of products to offer can be obtained by a linear program (Theorem 3). Also, we show that as the revenues of the products increase by the same amount, the optimal subset to offer becomes larger (Lemma 4). This property

becomes critical when we study the optimal policy for the single resource revenue management problem. We show that the optimal subset of products to offer is not nested by revenue in general, so that we may offer a product with a smaller revenue, but not offer a product with a larger revenue. We give one sufficient condition on the Markov chain choice model under which it is optimal to offer a nested by revenue subset (Lemma 5).

Second, we consider revenue management problems with a single resource. In this setting, we need to decide which set of products to make available to customers dynamically over a selling horizon. At each time period, an arriving customer chooses among the set of available products. There is a limited inventory of the resource and the sale of a product consumes one unit of the resource. The goal is to find a policy to decide which subset of products to make available at each time period so as to maximize the total expected revenue. Assuming that customers choose under the Markov chain choice model, we formulate the problem as a dynamic program and show that the optimal policy offers a larger subset of products as we have more remaining capacity at a certain time period or as we get closer to the end of the selling horizon, all else being equal (Theorem 6). That is, as we have more capacity or as we get closer to the end of the selling horizon, the urgency to liquidate the resource inventory takes over and we offer a larger subset of products. Using these properties, we show that the optimal policy can be implemented by associating a protection level with each product so that a product is made available to the customers whenever the remaining inventory of the resource exceeds the protection level of the product. Our results imply that as we have less remaining capacity at a certain time period, we offer a smaller subset of products. In general, we show that as we have less remaining capacity, we may stop offering a product with a larger revenue before we stop offering a product with a smaller

revenue. So, the order in which we stop offering the products does not follow the revenue order of the products. We give one sufficient condition on the Markov chain choice model under which the order in which we stop offering the products follows the revenue order of the products (Lemma 7).

Third, we consider revenue management problems over a network of resources. In the network revenue management setting, we have a number of resources with limited inventories and each product consumes a certain combination of resources. We need to decide which subset of products to make available dynamically over a selling horizon. At each time period, an arriving customer chooses among the subset of available products according to the Markov chain choice model. The goal is to find a policy to decide which subset of products to make available at each time period so as to maximize the total expected revenue. We can formulate the network revenue management problem as a dynamic program, but this dynamic program requires keeping track of the remaining inventory for all resources, so it can be difficult to solve. Instead, we focus on a deterministic linear programming approximation formulated under the assumption that the customer choices take on their expected values. In this linear program, there is one decision variable for each subset of products, which corresponds to the frequency with which we offer a subset of products to customers. So, the number of decision variables increases exponentially with the number of products and the deterministic linear program is solved by using column generation.

Focusing on the deterministic linear program described above, we show that if the customers choose according to the Markov chain choice model, then the deterministic linear program can immediately be reduced to an equivalent linear program whose numbers of decision variables and constraints increase only linearly

with the numbers of products and resources (Theorem 8). We develop an algorithm to recover the optimal solution to the original deterministic linear program by using the optimal solution to the reduced linear program. This algorithm allows us to recover the frequency with which we should offer each subset of products to customers by using the optimal solution to the reduced linear program. Finally, by using the reduced linear program, we show that the optimal solution to the deterministic linear program offers only nested subsets of products (Theorem 11). In other words, the subsets of products offered by the optimal solution to the deterministic linear program can be ordered such that one subset is included in another one. This result implies that the optimal solution to the deterministic linear program offers at most $n + 1$ different subsets, where n is the number of products. Therefore, the optimal solution to the deterministic linear program does not offer too many different subsets and these subsets are related to each other in the sense that they are nested. We note that these properties of the optimal solution are consequences of the Markov chain choice model and they do not necessarily hold under other choice models. Our computational experiments show that using the reduced linear program can provide remarkable computational savings over solving the original deterministic linear program through column generation. For problem instances with 100 resources and 2000 products, we cannot solve the original deterministic linear program within two hours of run time, while we can use the reduced linear program to obtain the optimal subset offer frequencies within four minutes.

In addition to our contributions to solving assortment, single resource revenue management and network revenue management problems, we provide support for the Markov chain choice model by showing that this choice model is consistent with the random utility maximization principle, where each customer associates random

utilities with the available options and chooses the option that provides the largest utility. Furthermore, we provide computational experiments to demonstrate that the Markov chain choice model may capture the customer choice behavior better when compared with the popular multinomial logit model, while ensuring that the corresponding optimization problems still remain tractable.

Related Literature. The Markov chain choice model has recently been proposed by [1]. The authors show that the multinomial logit model, which is often used to model customer choices in practice, is a special case of the Markov chain choice model. They also show that the Markov chain choice model can approximate a rich class of choice models quite accurately. The authors study assortment optimization problems under the Markov chain choice model without inventory considerations and show that the optimal assortment can be computed through a dynamic program. We give an alternative solution approach for the assortment problem that is based on a linear program. [1] do not focus on revenue management problems with limited resources. We show structural properties of the optimal policy for the single resource revenue management problem and show how to reduce the size of the deterministic linear programming formulation for the network revenue management problem.

[11] show that if the customers choose according to the multinomial logit model, then the deterministic linear program for the network revenue management problem can be reduced to an equivalent linear program whose size grows linearly with the numbers of products and resources. We find that an analogue of this result can be established under the Markov chain choice model, which is, as mentioned above, more general than the multinomial logit model. Obtaining the optimal solution to the original deterministic linear program through the optimal solution to

the reduced linear program is substantially more difficult under the Markov chain choice model and our results demonstrate how to accomplish this task. Lastly, [41] formulate the single resource revenue management problem as a dynamic program and derive structural properties of the optimal policy. Our study of the single resource revenue management problem is based on their model. Overall, effectively solving revenue management problems under the Markov chain choice model has important implications since the work of [1] shows that the Markov chain choice model can approximate a rich class of choice models quite accurately.

There is literature on assortment problems without inventory considerations. [34], [46] and [47] study assortment problems when customers choose according to variants of the multinomial logit model. [2], [30] and [5] and [35] focus on assortment problems when the customer choices are governed by a mixture of multinomial logit models. [21], [4], [9] and [22] develop tractable solution methods for the assortment problem when customers choose under the nested logit model. [14] and [6] use a nonparametric choice model to capture customer choices, where each customer arrives with a particular ordering of products in mind and purchases the first available product in her ordering. [23] and [10] study related pricing problems under the nested logit model.

It is common to formulate deterministic linear programming approximations for network revenue management problems under the assumption that customer choices take on their expected values. Such approximations appear in [8], [24], [2], [19], [40], [29] and [44]. [50], [49], [20] and [28] provide tractable methods to approximate the dynamic programming formulations of network revenue management problems.

Organization. In Section 2.1, we formulate the Markov chain choice model

and show that it is consistent with the random utility maximization principle. In Section 2.2, we show how to solve the assortment problem under the Markov chain choice model. In Section 2.3, we give structural properties for the solution to the assortment problem. In Section 2.4, we characterize the optimal policy for the single resource revenue management problem through protection levels. In Section 2.5, we show that the deterministic linear program for the network revenue management problem can be reduced to an equivalent linear program whose size increases linearly with the numbers of products and resources. In Section 2.6, we give an algorithm to recover the optimal solution to the original deterministic linear program by using the optimal solution to the reduced linear program. In Section 2.7, we give computational experiments to demonstrate the benefits from using the Markov chain choice model instead of simpler choice models. In Section 4.9, we give computational experiments to demonstrate the benefits from the reduced linear program. In Section 4.10, we conclude.

2.1 Markov Chain Choice Model

In the Markov chain choice model, there are n products indexed by $N = \{1, \ldots, n\}$. With probability λ_j, a customer arrives into the system to purchase product j. If this product is available for purchase, then the customer purchases it. Otherwise, the customer transitions to product i with probability $\rho_{j,i}$ and checks whether product i is available for purchase. With probability $1 - \sum_{i \in N} \rho_{j,i}$, the customer transitions to the no purchase option and leaves the system without a purchase. In this way, the customer transitions between the products according to a Markov chain until she visits a product available for purchase or she visits the no purchase

option. If she visits a product available for purchase, then she purchases this product. If she visits the no purchase option, then she leaves the system without a purchase. Given that we offer the subset $S \subset N$ of products, we use $P_{j,S}$ to denote the probability that a customer visits product j that is available for purchase. By definition, we have $P_{j,S} = 0$ for all $j \notin S$. If a customer visits product j and product j is available, then the customer purchases this product. Thus, $P_{j,S}$ corresponds to the probability that a customer purchases product j when we offer the subset S of products. Given that we offer the subset $S \subset N$ of products, we use $R_{j,S}$ to denote the probability that a customer visits product j that is not available for purchase. By definition, we have $R_{j,S} = 0$ for all $j \in S$. Using the vectors $P_S = (P_{1,S}, \ldots, P_{n,S})$ and $R_S = (R_{1,S}, \ldots, R_{n,S})$, we observe that (P_S, R_S) satisfies

$$P_{j,S} + R_{j,S} = \lambda_j + \sum_{i \in N} \rho_{i,j} R_{i,S} \quad \forall j \in N, \tag{Balance}$$

$$P_{j,S} = 0 \quad \forall j \notin S, \qquad R_{j,S} = 0 \quad \forall j \in S.$$

We interpret the Balance equations as follows. If $j \in S$, then $P_{j,S}$ is the probability that a customer visits product j during her choice process and we have $R_{j,S} = 0$ by definition. Similarly, if $j \notin S$, then $R_{j,S}$ is the probability that a customer visits product j during her choice process and we have $P_{j,S} = 0$ by definition. Thus, $P_{j,S} + R_{j,S}$ on the left side of the Balance equations corresponds to the probability that a customer visits product j during her choice process. For a customer to visit product j during her choice process, she should either arrive into the system to purchase product j or she should visit some product i that is not available for purchase and transition from product i to product j. Thus, $\lambda_j + \sum_{i \in N} \rho_{i,j} R_{i,S}$ on the right side of the Balance equations corresponds to the probability that a customer arrives into the system to purchase product j or she visits some product i that is not available for purchase and transitions from product i to product j.

There are a total of $2n$ probabilities in (P_S, R_S). The BALANCE equations provide a total of $2n$ equations. Solving these equations for the probabilities in (P_S, R_S), $P_{j,S}$ yields the probability that a customer purchases product j when we offer the subset S of products. In the Markov chain choice model, we can interpret λ_j as the probability that product j is the first choice of a customer and $\rho_{j,i}$ as the probability that the next choice of a customer is product i given that her current choice is product j.

Throughout the paper, we assume that $\lambda_j > 0$ and $\sum_{i \in N} \rho_{j,i} < 1$ for all $j \in N$, in which case, there is a strictly positive probability that a customer arrives into the system to purchase any of the products and a customer can leave the system after she visits any of the products. These assumptions allow us to avoid degenerate cases, but all of our results hold without any modifications when we have $\lambda_j = 0$ or $\sum_{i \in N} \rho_{j,i} = 1$ for some $j \in N$. Since we have $\lambda_j > 0$ and $R_{j,S} = 0$ for all $j \in S$, the BALANCE equations imply that $P_{j,S} > 0$ for all $j \in S$. Thus, there is a strictly positive probability that each product in the offered subset is purchased by an arriving customer.

The Markov chain choice model may not always reflect the thought process of a customer when making a purchase. Nevertheless, the important point is that if the purchase probabilities $(P_{1,S}, \ldots, P_{n,S})$ computed through the Markov chain choice model can accurately reflect the probabilities that a customer purchases different products when we offer the subset S of products, then it is not necessary to insist that the Markov chain choice model reflects the thought process of a customer. It is not difficult to see why the Markov chain choice model may not always reflect the thought process of a customer when making a purchase. For example, the Markov chain choice model implies that the thought process of a

customer is memoryless. Given that the current choice of a customer is product j, the probability $\rho_{j,i}$ that her next choice is product i does not depend on her previous choices. Furthermore, making transitions over the full set of products N implies that a customer has the knowledge of all of the products that can possibly be offered for purchase. In reality, a customer usually has the knowledge of only the products that are actually offered for purchase, possibly with the addition of a few other products that can possibly be offered. When a customer has the knowledge of only the products that are actually offered for purchase, we may have to model the transition probability $\rho_{j,i}$ as a quantity that is dependent on the subset of the offered products, but this extension dramatically complicates the analysis. Finally, the Markov chain choice model implies that the thought process of a customer may visit a product that is not available for purchase multiple times.

An appealing approach for constructing various choice models is based on the random utility maximization principle. Under the random utility maximization principle, a customer associates random utilities with the products. If we offer the subset S of products, then the customer purchases the product in the subset S that provides the largest utility, as long as the utility of this product is positive. If none of the utilities of the products in the subset S is positive, then the customer leaves the system without making a purchase. Thus, the utility of the no purchase option is normalized to zero. The purchase probabilities under many choice models, such as the multinomial logit model and the nested logit model, can be derived by using the random utility maximization principle. In the remainder of this section, we show that the purchase probabilities under the Markov chain choice model can also be derived by using the random utility maximization principle. To show this result, we use the random variable U_j to capture the utility of product j. We use σ_j to denote the rank of product j when we sort the products in the order

of decreasing utilities and focus only on the products with positive utilities. We follow the convention that if the utility of product j is not positive, then we have $\sigma_j = \infty$. For example, if there are five products and the utilities of these products are $(U_1, U_2, U_3, U_4, U_5) = (2.1, 5.9, -2.3, 7.4, 4.8)$, then we have $\sigma_4 = 1$, $\sigma_2 = 2$, $\sigma_5 = 3$, $\sigma_1 = 4$ and $\sigma_3 = \infty$. Since the utilities of the products are random variables, $(\sigma_1, \ldots, \sigma_n)$ are random variables as well. We refer to the random variables $(\sigma_1, \ldots, \sigma_n)$ as rank random variables. Under the random utility maximization principle, if we offer the subset S of products, then a customer purchases product j as long as product j has the largest utility among the products in the subset S and the utility of product j is positive. Therefore, under the random utility maximization principle, if we offer the subset S of products, then the probability that a customer purchases product j is given by $\mathbb{P}\{\sigma_j = \min_{i \in S} \sigma_i \text{ and } \sigma_j < \infty\}$. In the next theorem, we show that the purchase probabilities under the Markov chain choice model can be derived by using the random utility maximization principle.

Theorem 1. *For any Markov chain choice model, there exist rank random variables $(\sigma_1, \ldots, \sigma_n)$ such that if we offer the subset S of products, then the probability that a customer purchases product j under the Markov chain choice model is given by $\mathbb{P}\{\sigma_j = \min_{i \in S} \sigma_i \text{ and } \sigma_j < \infty\}$.*

Proof. Consider the states $\{X_t : t = 1, 2 \ldots\}$ visited by a customer following a Markov chain over the state space $N \cup \{0\}$. The state $j \in N$ corresponds to product j, whereas the state 0 corresponds to the no purchase option. In the Markov chain, the initial probabilities are given by $\mathbb{P}\{X_1 = j\} = \lambda_j$ for all $j \in N$ and $\mathbb{P}\{X_1 = 0\} = 1 - \sum_{j \in N} \lambda_j$, whereas the transition probabilities are given by $\mathbb{P}\{X_{t+1} = i \,|\, X_t = j\} = \rho_{j,i}$ for all $j, i \in N$, $\mathbb{P}\{X_{t+1} = 0 \,|\, X_j = j\} = 1 - \sum_{i \in N} \rho_{j,i}$ and $\mathbb{P}\{X_{t+1} = 0 \,|\, X_t = 0\} = 1$. Thus, if we consider the states visited by a customer that follows this Markov chain, then she starts in product j with

probability λ_j and transitions from product j to product i with probability $\rho_{j,i}$. The customer transitions from product j to state 0 with probability $1 - \sum_{i \in N} \rho_{j,i}$ and the customer never gets out of state 0. We observe that the states $\{X_t : t = 1, 2 \ldots\}$ correspond to the states visited by a customer that makes a purchase under the Markov chain choice model. Furthermore, under the Markov chain choice model, if we offer the subset S of products, then for a customer to purchase product j, product j should be visited before any other product in the subset S and before state 0. We use σ_j to denote the rank of product j when we sort the products in the order of increasing first visit time in the Markov chain and focus only on the products that are visited before state 0. We follow the convention that if product j is not visited before state 0, then we have $\sigma_j = \infty$. For example, if there are five products and the states visited by the Markov chain are given by $\{X_t : t = 0, 1, \ldots\} = \{4, 2, 4, 2, 5, 2, 1, 4, 0, \ldots\}$, then we have $\sigma_4 = 1$, $\sigma_2 = 2$, $\sigma_5 = 3$, $\sigma_1 = 4$ and $\sigma_3 = \infty$. Since the states visited by the Markov chain are random variables, $(\sigma_1, \ldots, \sigma_n)$ are random variables as well. Under the Markov chain choice model, if we offer the subset S of products, then for a customer to purchase product j, product j should be visited before any other product in the subset S and before state 0. Therefore, under the Markov chain choice model, if we offer the subset S of products, then the probability that a customer purchases product j is given by $\mathbb{P}\{\sigma_j = \min_{i \in S} \sigma_i \text{ and } \sigma_j < \infty\}$. □

The proof of Theorem 1 shows that the rank random variables $(\sigma_1, \ldots, \sigma_n)$ have connections to the order in which the Markov chain visits different states for the first time. In the next section, we focus on the assortment optimization problem under the Markov chain choice model.

2.2 Assortment Optimization

In the assortment optimization setting, we have access to a set of products among which we choose a subset to offer to customers. There is a revenue associated with each product. Customers choose among the offered products according to the Markov chain choice model. The goal is to find a subset of products that maximizes the expected revenue obtained from each customer. Indexing the products by $N = \{1, \ldots, n\}$, we use r_j to denote the revenue associated with product j. We recall that if we offer the subset S of products, then a customer purchases product j with probability $P_{j,S}$, where (P_S, R_S) satisfies the BALANCE equations. We can find the subset of products that maximizes the expected revenue obtained from each customer by solving the problem

$$\max_{S \subset N} \left\{ \sum_{j \in N} P_{j,S}\, r_j \right\}. \qquad \text{(ASSORTMENT)}$$

We observe that even computing the objective value of the problem above for a certain subset S is not a trivial task, since computing $P_{j,S}$ requires solving a system of equalities given by the BALANCE equations. In this section, we show that the optimal solution to the 3.1 problem can be obtained by solving a linear program. Furthermore, this linear program allows us to derive certain structural properties of the optimal subset of products to offer and these structural properties become useful later in the paper. To obtain the optimal solution to the problem above by solving a linear program, we exploit a connection between (P_S, R_S) and the extreme points of a suitably defined polytope. Consider the polytope defined as

$$\mathcal{H} = \left\{ (x, z) \in \Re_+^{2n} \;:\; x_j + z_j = \lambda_j + \sum_{i \in N} \rho_{i,j}\, z_i \;\forall\, j \in N \right\},$$

where we use the vectors $x = (x_1, \ldots, x_n)$ and $z = (z_1, \ldots, z_n)$. In the next lemma, we give a connection between (P_S, R_S) and the extreme points of $\mathcal{H}$.

Lemma 2. *For an extreme point* $(\hat{x}, \hat{z})$ *of* $\mathcal{H}$*, define* $S_{\hat{x}} = \{j \in N : \hat{x}_j > 0\}$*. Then, we have* $P_{j,S_{\hat{x}}} = \hat{x}_j$ *and* $R_{j,S_{\hat{x}}} = \hat{z}_j$ *for all* $j \in N$*.*

Proof. We claim that $\hat{z}_j = 0$ for all $j \in S_{\hat{x}}$. To get a contradiction, assume that $\hat{z}_j > 0$ for some $j \in S_{\hat{x}}$. By the definition of $S_{\hat{x}}$, there are $|S_{\hat{x}}|$ nonzero components of the vector $\hat{x}$. We have $\hat{x}_j = 0$ for all $j \notin S_{\hat{x}}$. Since $(\hat{x}, \hat{z})$ satisfies $\hat{x}_j + \hat{z}_j = \lambda_j + \sum_{i \in N} \rho_{i,j}\, \hat{z}_i$ for all $j \in N$ and $\lambda_j > 0$ for all $j \in N$, having $\hat{x}_j = 0$ for all $j \notin S_{\hat{x}}$ implies that $\hat{z}_j > 0$ for all $j \notin S_{\hat{x}}$. Therefore, there are $n - |S_{\hat{x}}|$ nonzero components of the vector $\hat{z}$ corresponding to the products that are not in $S_{\hat{x}}$. By our assumption, there is one more nonzero component of the vector $\hat{z}$ that corresponds to one of the products in $S_{\hat{x}}$. Therefore, it follows that $(\hat{x}, \hat{z})$ has $|S_{\hat{x}}| + n - |S_{\hat{x}}| + 1 = n + 1$ nonzero components. Since an extreme point of a polytope defined by n equalities cannot have more than n nonzero components, we get a contradiction and the claim follows. By the claim and the definition of $S_{\hat{x}}$, we have $\hat{x}_j = 0$ for all $j \notin S_{\hat{x}}$ and $\hat{z}_j = 0$ for all $j \in S_{\hat{x}}$. Furthermore, noting that $(\hat{x}, \hat{z}) \in \mathcal{H}$, we have $\hat{x}_j + \hat{z}_j = \lambda_j + \sum_{i \in N} \rho_{i,j}\, \hat{z}_j$ for all $j \in N$. The last two statements imply that $(\hat{x}, \hat{z})$ satisfies the BALANCE equations with $S = S_{\hat{x}}$. Thus, it must be the case that $(\hat{x}, \hat{z}) = (P_{S_{\hat{x}}}, R_{S_{\hat{x}}})$. □

An important implication of Lemma 2 is that we can obtain the optimal objective value of the 3.1 problem by solving the linear program

$$\max_{(x,z) \in \Re^{2n}_+} \left\{ \sum_{j \in N} r_j\, x_j \;:\; x_j + z_j = \lambda_j + \sum_{i \in N} \rho_{i,j}\, z_i \;\forall\, j \in N \right\}.$$

To see this result, we use $\hat{S}$ to denote the optimal solution to the 3.1 problem. Since $(P_{\hat{S}}, R_{\hat{S}})$ satisfies the BALANCE equations with $S = \hat{S}$, it follows that $(P_{\hat{S}}, R_{\hat{S}})$ is a feasible solution to the linear program above providing the objective value $\sum_{j \in N} r_j\, P_{j,\hat{S}}$. Therefore, there exists a feasible solution to the linear program

above, which provides an objective value that is equal to the optimal objective value of the 3.1 problem, indicating that the optimal objective value of the linear program above is at least as large as the optimal objective value of the 3.1 problem. On the other hand, letting $(\hat{x}, \hat{z})$ be the optimal solution to the linear program above, define the subset $S_{\hat{x}} = \{j \in N : \hat{x}_j > 0\}$. Without loss of generality, we assume that $(\hat{x}, \hat{z})$ is an extreme point solution, in which case, Lemma 2 implies that $P_{j,S_{\hat{x}}} = \hat{x}_j$ for all $j \in N$. Therefore, the subset $S_{\hat{x}}$ provides an objective value of $\sum_{j \in N} r_j \, P_{j,S_{\hat{x}}} = \sum_{j \in N} r_j \, \hat{x}_j$ for the 3.1 problem, indicating that there exists a feasible solution to the 3.1 problem, which provides an objective value that is equal to the optimal objective value of the linear program above. Thus, the optimal objective value of the 3.1 problem is at least as large as the optimal objective value of the linear program above, establishing the desired result. Naturally, we can obtain the optimal objective value of the linear program above by using its dual, which is given by

$$\min_{v \in \Re^n} \left\{ \sum_{j \in N} \lambda_j \, v_j \;\; : \;\; v_j \geq r_j \; \forall \, j \in N, \quad v_j \geq \sum_{i \in N} \rho_{j,\,i} \, v_i \; \forall \, j \in N \right\}, \qquad \text{(DUAL)}$$

where we use the vector $v = (v_1, \ldots, v_n)$. In the next theorem, we show that the DUAL problem can be used to obtain the optimal solution to the 3.1 problem.

Theorem 3. *Letting $\hat{v}$ be the optimal solution to the* DUAL *problem, define $\hat{S} = \big\{j \in N : \hat{v}_j = r_j\big\}$. Then, $\hat{S}$ is the optimal solution to the 3.1 problem.*

Proof. We use $\hat{Z}$ to denote the optimal objective value of the DUAL problem. By the discussion right before the theorem, $\hat{Z}$ also corresponds to the optimal objective value of the 3.1 problem. We note that for each $j \in N$, we have $\hat{v}_j = r_j$ or $\hat{v}_j = \sum_{i \in N} \rho_{j,i} \, \hat{v}_i$. In particular, if we have $\hat{v}_j > r_j$ and $\hat{v}_j > \sum_{i \in N} \rho_{j,i} \, \hat{v}_i$ for some $j \in N$, then we can decrease the value of the decision variable $\hat{v}_j$ by a small amount, while keeping the feasibility of the solution $\hat{v}$ for the DUAL problem. The

solution obtained in this fashion provides a strictly smaller objective value than the optimal solution, which is a contradiction. Since we have $\hat{v}_j = r_j$ or $\hat{v}_j = \sum_{i \in N} \rho_{j,i} \hat{v}_i$ for all $j \in N$, by the definition of $\hat{S}$, it holds that $\hat{v}_j = r_j$ for all $j \in \hat{S}$ and $\hat{v}_j = \sum_{i \in N} \rho_{j,i} \hat{v}_i$ for all $j \notin \hat{S}$. By the BALANCE equations, we also have $P_{j,\hat{S}} = 0$ for all $j \notin \hat{S}$ and $R_{j,\hat{S}} = 0$ for all $j \in \hat{S}$. Therefore, we obtain $P_{j,\hat{S}} \hat{v}_j = P_{j,\hat{S}} r_j$ for all $j \in N$ and $R_{j,\hat{S}} \hat{v}_j = \sum_{i \in N} R_{j,\hat{S}} \rho_{j,i} \hat{v}_i$ for all $j \in N$. Adding the last two equalities over all $j \in N$, it follows that $\sum_{j \in N} P_{j,\hat{S}} \hat{v}_j + \sum_{j \in N} R_{j,\hat{S}} \hat{v}_j = \sum_{j \in N} P_{j,\hat{S}} r_j + \sum_{j \in N} \sum_{i \in N} R_{j,\hat{S}} \rho_{j,i} \hat{v}_i$. If we arrange the terms in this equality, then we obtain

$$\sum_{j \in N} P_{j,\hat{S}} r_j = \sum_{j \in N} \left\{ P_{j,\hat{S}} + R_{j,\hat{S}} - \sum_{i \in N} \rho_{i,j} R_{i,\hat{S}} \right\} \hat{v}_j = \sum_{j \in N} \lambda_j \hat{v}_j = \hat{Z},$$

where the second equality uses the fact that $(P_{\hat{S}}, R_{\hat{S}})$ satisfies the BALANCE equations with $S = \hat{S}$ and the third equality is by the fact that $\hat{v}$ is the optimal solution to the DUAL problem. Since $\hat{Z}$ also corresponds to the optimal objective value of the 3.1 problem, having $\sum_{j \in N} P_{j,\hat{S}} r_j = \hat{Z}$ implies that $\hat{S}$ is the optimal solution to the 3.1 problem. □

Thus, by Theorem 3, we can obtain the optimal solution to the 3.1 problem by solving the DUAL problem, which indicates that the 3.1 problem is tractable.

2.3 Properties of the Optimal Assortment

In this section, we show several structural properties of the optimal solution to the 3.1 problem. In the next lemma, we show that the optimal solution to the 3.1 problem becomes a smaller subset when we decrease the revenues associated with all of the products by the same positive amount. In particular, for $\eta \geq 0$, we let

$\hat{v}^\eta$ be the optimal solution to the DUAL problem when we decrease the revenues of all of the products by η. By Theorem 3, $\{j \in N : \hat{v}^\eta_j = r_j - \eta\}$ is the optimal solution to the 3.1 problem when we decrease the revenues associated with all of the products by η. In the next lemma, we show that $\{j \in N : \hat{v}^\eta_j = r_j - \eta\} \subset \{j \in N : \hat{v}^0_j = r_j\}$, which implies that if we decrease the revenues of all of the products by the same positive amount η, then the optimal solution to the 3.1 problem becomes a smaller subset. This result becomes useful when we show the optimality of protection level policies for the single resource revenue management problem.

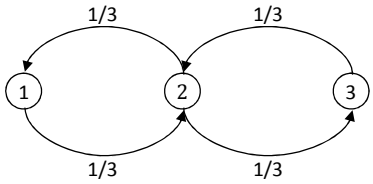
Lemma 4. *For $\eta \geq 0$, let $\hat{v}^\eta$ be the optimal solution to the* DUAL *problem when we decrease the revenues of all products by η. Then, we have $\{j \in N : \hat{v}^\eta_j = r_j - \eta\} \subset \{j \in N : \hat{v}^0_j = r_j\}$.*

Proof. By the same argument in the proof of Theorem 3, we have $\hat{v}^0_j = r_j$ or $\hat{v}^0_j = \sum_{i \in N} \rho_{j,i} \hat{v}^0_i$ for all $j \in N$, in which case, noting the constraints in the DUAL problem, we obtain $\hat{v}^0_j = \max\{r_j, \sum_{i \in N} \rho_{j,i} \hat{v}^0_i\}$ for all $j \in N$. A similar argument yields $\hat{v}^\eta_j = \max\{r_j - \eta, \sum_{i \in N} \rho_{j,i} \hat{v}^\eta_i\}$ for all $j \in N$. We define $\tilde{v} = (\tilde{v}_1, \ldots, \tilde{v}_n)$ as $\tilde{v}_j = \min\{\hat{v}^0_j, \hat{v}^\eta_j + \eta\}$ for all $j \in N$. We claim that $\tilde{v}$ is a feasible solution to the DUAL problem. To see the claim, by the definition of $\tilde{v}$, we have $\tilde{v}_j \leq \hat{v}^0_j$ and $\tilde{v}_j \leq \hat{v}^\eta_j + \eta$ for all $j \in N$. In this case, using the fact that $\hat{v}^0_j = \max\{r_j, \sum_{i \in N} \rho_{j,i} \hat{v}^0_i\}$, we obtain $\max\{r_j, \sum_{i \in N} \rho_{j,i} \tilde{v}_i\} \leq \max\{r_j, \sum_{i \in N} \rho_{j,i} \hat{v}^0_i\} = \hat{v}^0_j$ for all $j \in N$. Similarly, using the fact that $\hat{v}^\eta_j = \max\{r_j - \eta, \sum_{i \in N} \rho_{j,i} \hat{v}^\eta_i\}$, we have $\max\{r_j, \sum_{i \in N} \rho_{j,i} \tilde{v}_i\} \leq \max\{r_j, \sum_{i \in N} \rho_{j,i} (\hat{v}^\eta_i + \eta)\} \leq \max\{r_j - \eta, \sum_{i \in N} \rho_{j,i} \hat{v}^\eta_i\} + \eta = \hat{v}^\eta_j + \eta$ for all $j \in N$. Therefore, the discussion so far shows that $\max\{r_j, \sum_{i \in N} \rho_{j,i} \tilde{v}_i\} \leq \hat{v}^0_j$ and $\max\{r_j, \sum_{i \in N} \rho_{j,i} \tilde{v}_i\} \leq \hat{v}^\eta_j + \eta$, indicating that $\max\{r_j, \sum_{i \in N} \rho_{j,i} \tilde{v}_i\} \leq \min\{\hat{v}^0_j, \hat{v}^\eta_j + \eta\} = \tilde{v}_j$ for all $j \in N$, where the last equality is by the definition of $\tilde{v}$. Having $\max\{r_j, \sum_{i \in N} \rho_{j,i} \tilde{v}_i\} \leq \tilde{v}_j$ for all $j \in N$ implies that $\tilde{v}$ is a feasible

solution to the DUAL problem and the claim follows. To get a contradiction to the result that we want to show, we assume that there exists some $j \in N$ such that $j \in \{i \in N : \hat{v}_i^\eta = r_i - \eta\}$, but $j \notin \{i \in N : \hat{v}_i^0 = r_i\}$. So, we have $\hat{v}_j^\eta + \eta = r_j < \hat{v}_j^0$, in which case, noting that $\tilde{v}_j = \min\{\hat{v}_j^0, \hat{v}_j^\eta + \eta\}$, we get $\tilde{v}_j < \hat{v}_j^0$. Since $\tilde{v}_i \leq \hat{v}_i^0$ for all $i \in N$ by the definition of $\tilde{v}$, it follows that the components of $\tilde{v}$ are no larger than the corresponding components of $\hat{v}^0$ and there exists some $j \in N$ such that $\tilde{v}_j$ is strictly smaller than $\hat{v}_j^0$. Since $\tilde{v}$ is a feasible solution to the DUAL problem, the last observation contradicts the fact that $\hat{v}^0$ is the optimal solution to the DUAL problem. □

If customers choose according to the multinomial logit model, then [41] show that the optimal solution to the 3.1 problem includes a certain number of products with the largest revenues. In other words, if the products are indexed such that $r_1 \geq \ldots \geq r_n$, then a subset of the form $\{1, \ldots, j\}$ for some $j \in N$ is optimal. We refer to a subset that includes a certain number of products with the largest revenues as a nested by revenue subset. In this case, if customers choose according to the multinomial logit model, then the optimal solution can efficiently be obtained by checking the expected revenue from each one of the nested by revenue subsets and choosing the subset that provides the largest expected revenue. A natural question is whether a nested by revenue subset is optimal under the Markov chain choice model. We give a counterexample to show that a nested by revenue subset is not necessarily optimal when customers choose according to the Markov chain choice model.

We consider a problem instance with three products. The revenues of the products are given by $(r_1, r_2, r_3) = (720, 225, 180)$. We observe that nested by revenue subsets for this problem instance are $\{1\}$, $\{1, 2\}$ and $\{1, 2, 3\}$. The probabili-



S	$P_{1,S}$	$P_{2,S}$	$P_{3,S}$	$\sum_{j=1}^{3} P_{j,S}\, r_j$
$\{1\}$	$1/2$	0	0	360
$\{1,2\}$	$1/3$	$4/9$	0	340
$\{1,2,3\}$	$1/3$	$1/3$	$1/3$	375
$\{1,3\}$	$4/9$	0	$4/9$	*400*

Table 2.1: A counterexample where a nested by revenue subset is not optimal.

ties that a customer arrives into the system to purchase each one of the three products are given by $(\lambda_1, \lambda_2, \lambda_3) = (1/3, 1/3, 1/3)$. The transition probabilities are given by $\rho_{1,2} = \rho_{2,1} = \rho_{2,3} = \rho_{3,2} = 1/3$. The other transition probabilities in $\{\rho_{j,i} : j, i \in N\}$ are zero. For example, if a customer visits product 2 and product 2 is not available, then she transitions to product 1 or product 3, each with probability 1/3. With probability 1/3, the customer transitions to the no purchase option. For this problem instance, the left side of Table 2.1 shows the transition probabilities. The right side of Table 2.1 shows the purchase probabilities $(P_{1,S}, P_{2,S}, P_{3,S})$ when we offer each subset S and the expected revenue $\sum_{j=1}^{3} P_{j,S}\, r_j$ from each subset S. The purchase probabilities $(P_{1,S}, P_{2,S}, P_{3,S})$ are obtained by solving the Balance equations for $(P_{1,S}, P_{2,S}, P_{3,S})$ and $(R_{1,S}, R_{2,S}, R_{3,S})$. The results in Table 2.1 indicate the subset $\{1, 3\}$ provides an expected revenue of 400, whereas the expected revenue from the best nested by revenue subset is 375. Therefore, a nested by revenue subset is not optimal for this problem instance. In Appendix A.1, we give a more involved counterexample to show that if there are n products, then the expected revenue provided by the best nested by revenue subset can deviate from the optimal expected revenue by a factor arbitrarily close to $n/2$, indicating that nested by revenue subsets can perform arbitrarily poorly under the Markov chain choice model.

It is tempting to come up with sufficient conditions on the Markov chain choice model to ensure that a nested by revenue subset is optimal for the 3.1 problem.

Unfortunately, it appears to be difficult to give general sufficient conditions. As shown in the previous paragraph, a nested by revenue subset is not necessarily optimal even when the Markov chain choice model follows a simple birth and death process. In the remainder of this section, we give one sufficient condition to ensure that a nested by revenue subset is optimal for the 3.1 problem. It is possible to show that if we have a Markov chain choice model such that $\rho_{j,i} = \lambda_i$ for all $j, i \in N$, then the purchase probabilities under the under the Markov chain choice model are equal to the purchase probabilities under a multinomial logit model. Conversely, if we have a multinomial logit model, then there exists a Markov chain choice model with $\rho_{j,i} = \lambda_i$ for all $j, i \in N$ such that the purchase probabilities under the multinomial logit model are equal to the purchase probabilities under the Markov chain choice model. Thus, a Markov chain choice model with $\rho_{j,i} = \lambda_i$ for all $j, i \in N$ is equivalent to a multinomial logit model. [1] discuss the equivalence of a Markov chain choice model with $\rho_{j,i} = \lambda_i$ for all $j, i \in N$ to a multinomial logit model. Since a nested by revenue subset is optimal under the multinomial logit model, it immediately follows that if we have a Markov chain choice model with $\rho_{j,i} = \lambda_i$ for all $j, i \in N$, then a nested by revenue subset is optimal for the 3.1 problem. Thus, if the rows of the transition probability matrix $\{\rho_{j,i} : j, i \in N\}$ are equal to each other and each row is given by $(\lambda_1, \ldots, \lambda_n)$, then it immediately follows that a nested by revenue subset is optimal for the 3.1 problem. In the next lemma, we give another sufficient condition on the Markov chain choice model to ensure that a nested by revenue subset is optimal. Under this sufficient condition, the behavior of the Markov chain choice model can be quite different from that of the multinomial logit model. The proof of this lemma follows from a simple monotonicity argument and we defer it to Appendix A.2.

Lemma 5. *Assume that the products are indexed such that* $r_1 \geq \ldots \geq r_n$ *and the*

transition probabilities satisfy $\rho_{j,i} \leq \rho_{j+1,i}$ *for all* $j = 1, \ldots, n-1$, $i \in N$. *Then, a nested by revenue subset is optimal for the 3.1 problem.*

The rows of the transition probability matrix in Lemma 5 do not have to be equal to each other and the probabilities $(\lambda_1, \ldots, \lambda_n)$ can be arbitrary. Thus, this transition probability matrix can be quite different from the one that yields a choice model that is equivalent to the multinomial logit model, indicating that the purchase probabilities provided by the transition probability matrix in Lemma 5 can be quite different from the purchase probabilities under the multinomial logit model. It is difficult to give a behavioral justification for the sufficient condition in Lemma 5 but one possible interpretation is as follows. We note that if product j is not available, then the probability that a customer transitions from product j to the no purchase option is given by $1 - \sum_{i \in N} \rho_{j,i}$. The sufficient condition in Lemma 5 implies that $1 - \sum_{i \in N} \rho_{j,i} \geq 1 - \sum_{j \in N} \rho_{j+1,i}$. Thus, if a customer visits a product with a larger revenue and this product is not available for purchase, then she has a larger probability of transitioning to the no purchase option.

2.4 Single Resource Revenue Management

In the single resource revenue management setting, we manage one resource with a limited amount of capacity. At each time period in the selling horizon, we need to decide which subset of products to offer to customers. Customers arrive into the system one by one and choose among the offered products according to the Markov chain choice model. When we sell a product, we generate a revenue and consume one unit of the resource. The goal is to find a policy to dynamically decide which subsets of products to offer over the selling horizon so as to maximize

the total expected revenue. Such single resource revenue management problems arise when airlines control the availability of different fare classes on a single flight leg. Different fare classes correspond to different products and the seats on the flight leg correspond to the resource. [41] consider such single resource revenue management problems when customers choose under a general choice model. In this section, we give structural properties of the optimal policy when customers choose under the Markov chain choice model. Similar to our notation in the previous section, we index the products by $N = \{1, \ldots, n\}$ and denote the revenue associated with product j by r_j. If we offer the subset S of products, then a customer purchases product j with probability $P_{j,S}$. We have T time periods in the selling horizon. Customers arrive one by one at each time period. We have c units of resource available at the beginning of the selling horizon. We let $V_t(x)$ be the optimal total expected revenue over the time periods $t, \ldots, T$, given that we have x units of remaining capacity at the beginning of time period t. We can compute $\{V_t(x) : x = 0, \ldots, c,\ t = 1, \ldots, T\}$ by solving the dynamic program

$$\begin{aligned} V_t(x) &= \max_{S \subset N} \left\{ \sum_{j \in N} P_{j,S} \left\{ r_j + V_{t+1}(x-1) \right\} + \left\{ 1 - \sum_{j \in N} P_{j,S} \right\} V_{t+1}(x) \right\} \\ &= \max_{S \subset N} \left\{ \sum_{j \in N} P_{j,S} \left\{ r_j + V_{t+1}(x-1) - V_{t+1}(x) \right\} \right\} + V_{t+1}(x), \end{aligned}$$

(Single Resource)

with the boundary conditions that $V_{T+1}(x) = 0$ for all $x = 0, \ldots, c$ and $V_t(0) = 0$ for all $t = 1, \ldots, T$. The optimal total expected revenue is given by $V_1(c)$.

We let $\hat{S}_t(x)$ be the optimal subset of products to offer given that we have x units of remaining capacity at the beginning of time period t, in which case, $\hat{S}_t(x)$ is given by the optimal solution to the problem on the right side of the Single Resource dynamic program. In this section, we show that there exists an optimal policy that satisfies the properties $\hat{S}_t(x-1) \subset \hat{S}_t(x)$ and $\hat{S}_{t-1}(x) \subset \hat{S}_t(x)$. The first

property implies that if we have fewer units of remaining capacity at a particular time period, then the optimal subset of products to offer becomes smaller. The second property implies that if we have more time periods left in the selling horizon with a particular number of units of remaining capacity, then the optimal subset of products to offer becomes smaller. The first property has an important implication when implementing the policy obtained from the SINGLE RESOURCE dynamic program. Since the optimal subset of products to offer becomes smaller when we have fewer units of remaining capacity at a time period, we let $\bar{x}_{jt}$ be the smallest value of the remaining capacity such that it is still optimal to offer product j at time period t. In this case, if we have x units of remaining capacity at the beginning of time period t and $x \geq \bar{x}_{jt}$, then it is optimal to offer product j. Otherwise, it is optimal not to offer product j. Therefore, we can associate a threshold value $\bar{x}_{jt}$ for each product j and time period t such that we can decide whether it is optimal to offer product j at time period t by comparing the remaining resource capacity with the threshold value. The threshold value $\bar{x}_{jt}$ is referred to as the protection level for product j at time period t and the resulting policy is referred to as a protection level policy.

Optimality of a protection level policy significantly simplifies the implementation of the optimal policy since we can separately decide whether to offer each product by comparing the remaining resource capacity with the threshold value of the product. Since protection level policies form the standard capacity control tool in revenue management systems, optimality of protection level policies is also likely to enhance the practical appeal of the Markov chain choice model. In the next theorem, we show that the optimal subset of products to offer becomes smaller as we have fewer units of remaining capacity at a particular time period or as we have more time periods left in the selling horizon with a particular number of units of

remaining capacity.

Theorem 6. *There exists an optimal policy for the* Single Resource *dynamic program such that* $\hat{S}_t(x-1) \subset \hat{S}_t(x)$ *and* $\hat{S}_{t-1}(x) \subset \hat{S}_t(x)$.

Proof. It is a standard result that the first differences of the value functions computed through the Single Resource dynamic program increases as we have fewer units of remaining capacity or as we have more time periods left in the selling horizon. In particular, letting $\Delta V_t(x) = V_t(x) - V_t(x-1)$, [41] show that $\Delta V_{t+1}(x) \leq \Delta V_{t+1}(x-1)$ and $\Delta V_{t+1}(x) \leq \Delta V_t(x)$ under any choice model. Letting $r_{jt}(x) = r_j - \Delta V_{t+1}(x)$, by definition, $\hat{S}_t(x)$ is the optimal solution to the problem $\max_{S \subset N} \sum_{j \in N} P_{j,s} \left(r_j - \Delta V_{t+1}(x)\right) = \max_{S \subset N} \sum_{j \in N} P_{j,s}\, r_{jt}(x)$, whereas $\hat{S}_t(x-1)$ is the optimal solution to the problem $\max_{S \subset N} \sum_{j \in N} P_{j,s} \left(r_j - \Delta V_{t+1}(x-1)\right) = \max_{S \subset N} \sum_{j \in N} P_{j,s} \left(r_{jt}(x) - (\Delta V_{t+1}(x-1) - \Delta V_{t+1}(x))\right)$. Identifying $r_{jt}(x)$ with the revenue of product j in the 3.1 problem, the problem that computes $\hat{S}_t(x)$ has the same form as the 3.1 problem. The problem that computes $\hat{S}_t(x-1)$ also has the same form as the 3.1 problem as long as we identify $r_{jt}(x) - (\Delta V_{t+1}(x-1) - \Delta V_{t+1}(x))$ with the revenue of product j in the 3.1 problem. Thus, the revenue of each product in the problem that computes $\hat{S}_t(x-1)$ is obtained by subtracting $\Delta V_{t+1}(x-1) - \Delta V_{t+1}(x)$ from the revenue of the corresponding product in the problem that computes $\hat{S}_t(x)$. By the discussion at the beginning of the proof, we have $\Delta V_{t+1}(x-1) - \Delta V_{t+1}(x) \geq 0$ and Lemma 4 implies that if we decrease the revenue of each product by the same positive amount, then the optimal solution to the 3.1 problem becomes a smaller subset. Thus, it follows that $\hat{S}_t(x-1) \subset \hat{S}_t(x)$. Following the same argument but using the fact that the first differences of the value functions increase as we have more time periods left in the selling horizon, we can show that $\hat{S}_{t-1}(x) \subset \hat{S}_t(x)$. □

Therefore, implementing the optimal policy obtained from the SINGLE RESOURCE dynamic program requires only keeping track of the optimal protection level $\bar{x}_{jt}$ for all $j \in N$ and $t = 1, \ldots, T$, instead of keeping track of the optimal subset of products $\hat{S}_t(x)$ to offer for all $x = 0, \ldots, c$ and $t = 1, \ldots, T$. By comparing the remaining resource capacity with $\bar{x}_{jt}$, we can decide whether product j should be offered at time period t.

Theorem 6 shows that the optimal subset of products to offer becomes smaller as we have fewer units of remaining capacity at a particular time period. In other words, we stop offering certain products as we have fewer units of remaining capacity at a particular time period. An interesting question is whether the order in which we stop offering the products follows the revenue order of the products so that we stop offering products with smaller revenues before we stop offering products with larger revenues. We give a counterexample to show that we may actually stop offering a product with a larger revenue, while continuing to offer a product with a smaller revenue. We consider a problem instance with three products and two time periods in the selling horizon. We have two units of resource at the beginning of the selling horizon. The revenues of the products are given by $(r_1, r_2, r_3) = (320, 195, 185)$. At each time period, the probabilities that a customer arrives into the system to purchase each one of the three products are given by $(\lambda_1, \lambda_2, \lambda_3) = (1/5, 1/5, 1/5)$. With probability $2/5$, there is no customer arrival. The transition probabilities are given by $\rho_{1,2} = \rho_{2,1} = \rho_{2,3} = \rho_{3,2} = 1/3$. The other transition probabilities in $\{\rho_{j,i} : j, i \in N\}$ are zero. These transition probabilities are the same as those on the left side of Table 2.1. For this problem instance, we show that if we have two units of remaining capacity at time period one, then it is optimal to offer the subset $\{1, 2, 3\}$, whereas if we have one unit of remaining capacity at time period one, then it is optimal to offer the subset $\{1, 3\}$. Thus, if

S	$P_{1,S}$	$P_{2,S}$	$P_{3,S}$	$\sum_{j=1}^{3} P_{j,S}\, r_j$	$\sum_{j=1}^{3} P_{j,S}\, (r_j - V_2(1))$
$\{1\}$	3/10	0	0	96	54
$\{2\}$	0	1/3	0	65	18.33
$\{3\}$	0	0	3/10	55.5	13.5
$\{1,2\}$	1/5	4/15	0	116	50.67
$\{1,3\}$	4/15	0	4/15	134.67	*60*
$\{2,3\}$	0	4/15	1/5	89	23.67
$\{1,2,3\}$	1/5	1/5	1/5	*140*	56

Table 2.2: A counterexample where we stop offering a product with larger revenue before we stop offering a product with smaller revenue as we have fewer units of remaining capacity.

the remaining capacity at time period one goes down from two to one, then we stop offering product 2, but we continue offering product 3, whose revenue is smaller than the revenue of product 2. In other words, we may stop offering a product with a larger revenue, while continuing to offer a product with a smaller revenue.

For this problem instance, Table 2.2 shows the purchase probabilities ($P_{1,S}$, $P_{2,S}$, $P_{3,S}$) when we offer each subset S along with the expected revenue $\sum_{j \in N} P_{j,S}\, r_j$ and $\sum_{j \in N} P_{j,S}\, (r_j - V_2(1))$ from each subset S. Since there are two time periods in the selling horizon, by the boundary conditions of the Single Resource dynamic program, we have $V_3(x) = 0$ for all $x = 0, 1, 2$ and $V_t(0) = 0$ for all $t = 1, 2$. In this case, the Single Resource dynamic program implies that $V_2(x) = \max_{S \subset N} \sum_{j \in N} P_{j,S}\, r_j$ for all $x = 1, 2$. In Table 2.2, we observe that the largest value in $\left\{\sum_{j \in N} P_{j,S}\, r_j : S \subset N\right\}$ is 140. Therefore, we have $V_2(x) = \max_{S \subset N} \sum_{j \in N} P_{j,S}\, r_j = 140$ for all $x = 1, 2$. Consider the optimal subset of products to offer at time period one. If we have two units of remaining capacity at time period one, then the Single Resource dynamic program solves the problem $\max_{S \subset N} \sum_{j \in N} P_{j,S}\, (r_j + V_2(1) - V_2(2)) = \max_{S \subset N} \sum_{j \in N} P_{j,S}\, r_j$ to the find the optimal subset of products to offer, where the equality uses the fact that $V_2(2) = V_2(1) = 140$. Since the largest value of $\sum_{j=1}^{n} P_{j,S}\, r_j$ in Table 2.2 occurs

when we use the subset $S = \{1, 2, 3\}$, it is optimal to offer the subset $\{1, 2, 3\}$ when we have two units of remaining capacity at time period one. If we have one unit of remaining capacity at time period one, then the SINGLE RESOURCE dynamic program solves the problem $\max_{S \subset N} \sum_{j \in N} P_{j,S} (r_j + V_2(0) - V_2(1)) = \max_{S \subset N} \sum_{j \in N} P_{j,S} (r_j - V_2(1)) = \max_{S \subset N} \sum_{j \in N} P_{j,S} (r_j - 140)$ to find the optimal subset of products to offer, where the two equalities use the fact that $V_2(0) = 0$ and $V_2(1) = 140$. Since the largest value of $\sum_{j \in N} P_{j,S} (r_j - 140)$ in Table 2.2 occurs when we use the subset $S = \{1, 3\}$, it is optimal to offer the subset $\{1, 3\}$ when we have one unit of remaining capacity at time period one. Therefore, if the remaining capacity at time period one goes down from two to one, then we stop offering product 2, but we continue offering product 3, whose revenue is smaller than the revenue of product 2. That is, as the remaining capacity at a time period decreases, we may stop offering a product with a larger revenue before we stop offering a product with a smaller revenue.

Under the Markov chain choice model, Theorem 6 shows that the optimal subsets of products to offer with different units of remaining capacities are nested such that $\hat{S}_t(0) \subset \hat{S}_t(1) \subset \ldots \subset \hat{S}_t(c)$, but the counterexample in Table 2.2 shows that the nesting order of the subsets may not follow the revenue order of the products. In the remainder of this section, we give one sufficient condition on the Markov chain choice model to ensure that the nesting order of the subsets $\{\hat{S}_t(x) : x = 0, \ldots, c\}$ follows the revenue order of the products. We index the products such that $r_1 \geq \ldots \geq r_n$. In the next lemma, we show that if the Markov chain choice model satisfies the sufficient condition in Lemma 5, then $\hat{S}_t(x)$ is of the form $\{1, \ldots, j_t(x)\}$, where $j_t(x)$ is increasing in x and increasing in t. Since $j_t(x)$ is increasing in x, we have $\{1, \ldots, j_t(x-1)\} \subset \{1, \ldots, j_t(x)\}$. So, if we have fewer units of remaining capacity at time period t, then we stop offering the

products with smaller revenues before we stop offering the products with larger revenues. The proof is in Appendix A.3.

Lemma 7. *Assume that the products are indexed such that* $r_1 \geq \ldots \geq r_n$ *and the transition probabilities satisfy* $\rho_{j,i} \leq \rho_{j+1,i}$ *for all* $j = 1, \ldots, n-1$, $i \in N$. *Then, we have* $\hat{S}_t(x) = \{1, \ldots, j_t(x)\}$ *for some* $j_t(x) \in N \cup \{0\}$, *where* $j_t(x) = 0$ *corresponds to the case with* $\hat{S}_t(x) = \varnothing$. *Furthermore,* $j_t(x)$ *is increasing in* x *and increasing in* t.

The sufficient condition in Lemma 5 ensures that the optimal solution to the 3.1 problem is a nested by revenue subset. Lemma 7 shows that this sufficient condition also ensures that the nesting order of the subsets $\{\hat{S}_t(x) : x = 0, \ldots, c\}$ in the Single Resource dynamic program follows the revenue order of the products. It is worthwhile to observe that since $j_t(x)$ in Lemma 7 is increasing in t, this lemma also implies that if we have more time periods left in the selling horizon with x units of remaining capacity, then we stop offering products with smaller revenues before we stop offering products with larger revenues. Therefore, Lemma 7 also gives a sufficient condition to characterize the nesting order of the subsets $\{\hat{S}_t(x) : t = 1, \ldots, T\}$ by following the revenue order of the products.

2.5 Network Revenue Management

In the network revenue management setting, we manage a network of resources, each of which has a limited amount of capacity. At each time period in the selling horizon, we need to decide which subset of products to offer to customers. Customers arrive into the system one by one and choose among the offered products

according to the Markov chain choice model. Each product uses a certain combination of resources. When we sell a product, we generate a revenue and consume the capacities of the resources used by the product. The goal is to find a policy to dynamically decide which subsets of products to make available over the selling horizon so as to maximize the total expected revenue. Such network revenue management problems model the situation faced by an airline making itinerary availability decisions over a network of flight legs. Different itineraries correspond to different products and seats on different flight legs correspond to different resources. [8] and [24] study network revenue management problems when customers choose according to a general choice model. The dynamic programming formulation of the problem requires keeping track of the remaining capacities on all of the flight legs, resulting in a high dimensional state variable. To deal with this difficulty, the authors formulate a linear program under the assumption that the customer choices take on their expected values. In practice, this linear program may have a large number of decision variables and it is solved through column generation. In this section, we show that the size of the linear program can be drastically reduced when customers choose under the Markov chain choice model.

There are m resources indexed by $M = \{1, \ldots, m\}$. Similar to the earlier sections, we index the products by $N = \{1, \ldots, n\}$. Each customer chooses among the offered products according to the Markov chain choice model. Therefore, if we offer the subset S of products, then a customer purchases product j with probability $P_{j,S}$, where (P_S, R_S) satisfies the BALANCE equations. There are T time periods in the selling horizon. For simplicity of notation, we assume that at most one customer arrives at each time period. We have c_q units of resource q available at the beginning of the selling horizon. If we sell one unit of product j, then we generate a revenue of r_j and consume $a_{q,j}$ units of resource q. Assuming

that the customer choices take on their expected values, we can formulate the network revenue management problem as a linear program. In particular, using the decision variable u_S to capture the probability of offering subset S of products at a time period, we can find good subset offer probabilities $u = \{u_S : S \subset N\}$ by solving the linear program

$$\max_{u \in \Re_+^{2^n}} \left\{ \sum_{S \subset N} \sum_{j \in N} T\, r_j\, P_{j,S}\, u_S \;\; : \;\; \sum_{S \subset N} \sum_{j \in N} T\, a_{q,j}\, P_{j,S}\, u_S \leq c_q \quad \forall\, q \in M \right.$$
$$\left. \sum_{S \subset N} u_S = 1 \right\}. \qquad \text{(Choice Based)}$$

Noting the definition of the decision variable u_S, the expression $\sum_{S \subset N} T\, P_{j,S}\, u_S$ in the objective function and the first constraint of the problem above corresponds to the total expected number of sales for product j over the selling horizon. Therefore, the objective function in the problem above computes the total expected revenue over the selling horizon. The first constraint ensures that the total expected capacity of resource q consumed over the selling horizon does not exceed the capacity of this resource. The second constraint ensures that we offer a subset of products with probability one at each time period, but this subset can be the empty set.

In our formulation of the network revenue management problem, the probabilities (P_S, R_S) do not depend on the time period, which allows us to use a single decision variable u_S in the Choice Based linear program to capture the probability of offering subset S at any time period. In practical applications, customers arriving at different time periods can choose according to choice models with different parameters. If the choice models governing the customer choices at different time periods have different parameters, then we can address this situation by using the decision variable $u_{S,t}$, which corresponds to the probability of offering subset S of products at time period t. Our results in this section continue to hold with simple modifications when we work with the decision variables

$\{u_{S,t} : S \subset N,\ t = 1, \ldots, T\}$. We can interpret the CHOICE BASED linear program as a crude approximation to the network revenue management problem formulated under the assumption that the customer choices take on their expected values. [24] propose various approximate policies that use the solution to the CHOICE BASED linear program to decide which subset of products to offer at each time period.

Since the CHOICE BASED linear program has one decision variable for each subset of products, its number of decision variables grows exponentially with the number of products. Therefore, a common approach for solving this problem is to use column generation. In this section, we show that if customers choose according to the Markov chain choice model, then the CHOICE BASED linear program can equivalently be formulated as a linear program with only $2\,n$ decision variables and $m+n$ constraints. To give the equivalent formulation, we use the decision variable x_j to represent the fraction of customers that visit the available product j during their choice process and purchase this product. Also, we use the decision variable z_j to represent the fraction of customers that visit the unavailable product j during their choice process and do not purchase this product due to the unavailability of this product. Our main result in this section shows that the CHOICE BASED linear program is equivalent to the linear program

$$\max_{(x,z)\in\Re^{2n}_+} \left\{ \sum_{j\in N} T\, r_j\, x_j \ : \ \sum_{j\in N} T\, a_{q,j}\, x_j \le c_q \qquad \forall\, q \in M \right.$$
$$\left. x_j + z_j = \lambda_j + \sum_{i\in N} \rho_{i,j}\, z_i \quad \forall\, j \in N \right\}. \qquad \text{(REDUCED)}$$

By the definition of the decision variable x_j, the expression Tx_j in the objective function and the first constraint of the problem above corresponds to the expected number of sales for product j over the selling horizon. In this case, the objective function of the problem above computes the total expected revenue over the selling horizon. The first constraint ensures that the total expected capacity of resource

q consumed over the selling horizon does not exceed the capacity of this resource. The second constraint is similar to the BALANCE equations. Noting the definitions of the decision variables x_j and z_j, $x_j + z_j$ is the fraction of customers that visit product j during their choice process. For a customer to visit product j during her choice process, she should either arrive into the system to purchase product j or visit some unavailable product i, not purchase this product and transition from product i to product j. We note that there are $2\,n$ decision variables and $m + n$ constraints in the REDUCED linear program and it can be possible to solve the REDUCED linear program for practical networks directly by using linear programming software. In the next theorem, we show that the REDUCED linear program is equivalent to the CHOICE BASED linear program in the sense that we can use the optimal solution to the REDUCED linear program to obtain the optimal solution to the CHOICE BASED one.

Theorem 8. *Letting $(\hat{x}, \hat{z})$ be the optimal solution to the* REDUCED *linear program, there exist subsets $S^1, \ldots, S^K$ and positive scalars $\gamma^1, \ldots, \gamma^K$ summing to one such that $\hat{x} = \sum_{k=1}^{K} \gamma^k P_{S^k}$ and $\hat{z} = \sum_{k=1}^{K} \gamma^k R_{S^k}$. In this case, the optimal objective values of the* REDUCED *and* CHOICE BASED *linear programs are the same and the solution $\hat{u}$ obtained by letting $\hat{u}_{S^k} = \gamma^k$ for all $k = 1, \ldots, K$ and $\hat{u}_S = 0$ for all $S \notin \{S^1, \ldots, S^K\}$ is optimal to the* CHOICE BASED *linear program.*

Proof. Noting the definition of $\mathcal{H}$ in Section 2.2, since $(\hat{x}, \hat{z})$ is a feasible solution to the REDUCED linear program, we have $(\hat{x}, \hat{z}) \in \mathcal{H}$. Thus, there exist extreme points $(x^1, z^1), \ldots, (x^K, z^K)$ of $\mathcal{H}$ and positive scalars $\gamma^1, \ldots, \gamma^K$ summing to one such that $\hat{x} = \sum_{k=1}^{K} \gamma^k x^k$ and $\hat{z} = \sum_{k=1}^{K} \gamma^k z^k$. For all $k = 1, \ldots, K$, define the subset $S^k = \{j \in N : x_j^k > 0\}$. By Lemma 2, we have $P_{j,S^k} = x_j^k$ and $R_{j,S^k} = z_j^k$ for all $j \in N$, which implies that $\hat{x} = \sum_{k=1}^{K} \gamma^k x^k = \sum_{k=1}^{K} \gamma^k P_{S^k}$ and $\hat{z} = \sum_{k=1}^{K} \gamma^k z^k = \sum_{k=1}^{K} \gamma^k R_{S^k}$, showing the first part of the theorem. To show the second part of

the theorem, by the definition of $\hat{u}$, we have $\sum_{S \subset N} P_{j,S}\, \hat{u}_S = \sum_{k=1}^{K} P_{j,S^k}\, \gamma^k = \hat{x}_j$. Thus, we obtain $\sum_{j \in N} \sum_{S \subset N} T\, a_{q,j}\, P_{j,S}\, \hat{u}_S = \sum_{j \in N} T\, a_{q,j}\, \hat{x}_j \leq c_q$, where the inequality uses the fact that $(\hat{x}, \hat{z})$ is a feasible solution to the REDUCED linear program. Furthermore, we have $\sum_{S \subset N} \hat{u}_S = \sum_{k=1}^{K} \gamma^k = 1$. Therefore, $\hat{u}$ is a feasible solution to the CHOICE BASED linear program. Since $\sum_{S \subset N} P_{j,S}\, \hat{u}_S = \hat{x}_j$, we obtain $\sum_{S \subset N} \sum_{j \in N} T\, r_j\, P_{j,S}\, \hat{u}_S = \sum_{j \in N} T\, r_j\, \hat{x}_j$, which implies that the solution $\hat{u}$ provides an objective value for the CHOICE BASED linear program that is equal to the optimal objective value of the REDUCED linear program. Thus, the optimal objective value of the CHOICE BASED linear program is at least as large as that of the REDUCED linear program.

On the other hand, let $\tilde{u}$ be the optimal solution to the CHOICE BASED linear program. Define $(\tilde{x}, \tilde{z})$ as $\tilde{x}_j = \sum_{S \subset N} P_{j,S}\, \tilde{u}_S$ and $\tilde{z}_j = \sum_{S \subset N} R_{j,S}\, \tilde{u}_S$ for all $j \in N$. We have $\sum_{j \in N} T\, a_{q,j}\, \tilde{x}_j = \sum_{S \subset N} \sum_{j \in N} T\, a_{q,j}\, P_{j,S}\, \tilde{u}_S \leq c_q$, where we use the fact that $\tilde{u}$ satisfies the first constraint in the CHOICE BASED linear program. Since (P_S, R_S) satisfies the BALANCE equations, we have $P_{j,S} + R_{j,S} = \lambda_j + \sum_{i \in N} \rho_{i,j}\, R_{i,S}$ for all $S \subset N$. Multiplying this equality by $\tilde{u}_S$, adding over all $S \subset N$ and noting that $\tilde{u}$ is a feasible solution to the CHOICE BASED linear program satisfying $\sum_{S \subset N} \tilde{u}_S = 1$, we get $\tilde{x}_j + \tilde{z}_j = \sum_{S \subset N} P_{j,S}\, \tilde{u}_S + \sum_{S \subset N} R_{j,S}\, \tilde{u}_S = \lambda_j + \sum_{S \subset N} \sum_{i \in N} \rho_{i,j}\, R_{i,S}\, \tilde{u}_S = \lambda_j + \sum_{i \in N} \rho_{i,j}\, \tilde{z}_i$, where the first and last equalities use the definitions of $\tilde{x}$ and $\tilde{z}$. Thus, $(\tilde{x}, \tilde{z})$ is a feasible solution to the REDUCED linear program. Noting that $\sum_{j \in N} T\, r_j\, \tilde{x}_j = \sum_{S \subset N} \sum_{j \in N} T\, r_j\, P_{j,S}\, \tilde{u}_S$, the solution $(\tilde{x}, \tilde{z})$ provides an objective value for the REDUCED linear program that is equal to the optimal objective value of the CHOICE BASED linear program. Therefore, the optimal objective value of the REDUCED linear program is at least as large as that of the CHOICE BASED linear program. Noting the discussion at the end of the previous paragraph, the CHOICE BASED and REDUCED linear programs have the

same optimal objective value and the solution $\hat{u}$ defined in the theorem is optimal to the CHOICE BASED linear program. $\square$

By Theorem 8, we have a general approach for recovering the optimal solution to the CHOICE BASED linear program by using the optimal solution to the REDUCED linear program. Letting $(\hat{x}, \hat{z})$ be the optimal solution to the REDUCED linear program, the key is to find subsets $S^1, \ldots, S^K$ and positive scalars $\gamma^1, \ldots, \gamma^K$ summing to one such that $(\hat{x}, \hat{z})$ can be expressed as $\hat{x} = \sum_{k=1}^{K} \gamma^k \, P_{S^k}$ and $\hat{z} = \sum_{k=1}^{K} \gamma^k \, R_{S^k}$. Finding such subsets and scalars is not a straightforward task. In the next section, we give a tractable algorithm for this purpose.

2.6 Recovering the Optimal Solution

In this section, we consider the question of recovering the optimal solution to the CHOICE BASED linear program by using the optimal solution to the REDUCED linear program. By the discussion at the end of the previous section, letting $(\hat{x}, \hat{z})$ be the optimal solution to the REDUCED linear program, the key is to find subsets $S^1, \ldots, S^K$ and positive scalars $\gamma^1, \ldots, \gamma^K$ summing to one such that $(\hat{x}, \hat{z})$ can be expressed as $\hat{x} = \sum_{k=1}^{K} \gamma^k \, P_{S^k}$ and $\hat{z} = \sum_{k=1}^{K} \gamma^k \, R_{S^k}$. In this case, the solution $\hat{u}$ obtained by letting $\hat{u}_{S^k} = \gamma^k$ for all $k = 1, \ldots, K$ and $\hat{u}_S = 0$ for all $S \notin \left\{S^1, \ldots, S^K\right\}$ is optimal to the CHOICE BASED linear program. One possible approach to find the subsets $S^1, \ldots, S^K$ and scalars $\gamma^1, \ldots, \gamma^K$ is to express $(\hat{x}, \hat{z})$ as $\hat{x} = \sum_{S \subset N} \gamma_S \, P_S$ and $\hat{z} = \sum_{S \subset N} \gamma_S \, R_S$ for unknown positive scalars $\left\{\gamma_S : S \subset N\right\}$ satisfying $\sum_{S \subset N} \gamma_S = 1$ and solve these equations for the unknown scalars. Since there are $2\,n + 1$ equations, there exists a solution where at most $2\,n + 1$ of the unknown scalars take strictly positive values, but solving these equations directly

is not tractable since the number of unknown scalars grows exponentially with the number of products. We can use an idea similar to column generation, where we focus on a small subset of the unknown scalars $\{\gamma_S : S \subset N\}$ and iteratively extend this subset, but this approach can be as computationally intensive as solving the CHOICE BASED linear program directly by using column generation, which ultimately becomes problematic when dealing with large problem instances.

In this section, we give a tractable dimension reduction approach to find subsets $S^1, \ldots, S^K$ and positive scalars $\gamma^1, \ldots, \gamma^K$ summing to one such that the optimal solution $(\hat{x}, \hat{z})$ to the REDUCED linear program can be expressed as $\hat{x} = \sum_{k=1}^{K} \gamma^k P_{S^k}$ and $\hat{z} = \sum_{k=1}^{K} \gamma^k R_{S^k}$. For large network revenue management problem instances, our computational experiments show that this dimension reduction approach can recover the optimal solution to the CHOICE BASED linear program remarkably fast. Our dimension reduction approach uses the following recursive idea. Since $(\hat{x}, \hat{z})$ is a feasible solution to the REDUCED linear program, we have $(\hat{x}, \hat{z}) \in \mathcal{H}$. We define $S_{\hat{x}} = \{j \in N : \hat{x}_j > 0\}$ to capture the nonzero components of $\hat{x}$ and consider three cases. In the first case, we assume that $S_{\hat{x}} = \varnothing$, implying that $\hat{x}_j = 0$ for all $j \in N$. Since $(\hat{x}, \hat{z}) \in \mathcal{H}$ and $\hat{x}_j = 0$ for all $j \in N$, $(\hat{x}, \hat{z})$ satisfies the BALANCE equations with $S = \varnothing$, so that we must have $(\hat{x}, \hat{z}) = (P_\varnothing, R_\varnothing)$. Thus, we can express $(\hat{x}, \hat{z})$ simply as $\hat{x} = P_\varnothing$ and $\hat{z} = R_\varnothing$ and we are done. In the rest of the discussion, we assume that $S_{\hat{x}} \neq \varnothing$. We define $\alpha_{\hat{x}} = \min\{\hat{x}_j / P_{j,S_{\hat{x}}} : j \in S_{\hat{x}}\}$.

In the second case, we assume that $\alpha_{\hat{x}} \geq 1$. Having $\alpha_{\hat{x}} \geq 1$ implies that $\hat{x}_j \geq P_{j,S_{\hat{x}}}$ for all $j \in S_{\hat{x}}$. By the definition of $S_{\hat{x}}$ and the BALANCE equations, we also have $\hat{x}_j = 0 = P_{j,S_{\hat{x}}}$ for all $j \notin S_{\hat{x}}$. Thus, we have $\hat{x}_j \geq P_{j,S_{\hat{x}}}$ for all $j \in N$. Since $(\hat{x}, \hat{z}) \in \mathcal{H}$, Lemma 22 in Appendix A.4 shows that we also have $\hat{z}_j \geq R_{j,S_{\hat{x}}}$ for all $j \in N$. In this case, using the fact that $(\hat{x}, \hat{z}) \in \mathcal{H}$, if we add

the equalities in the definition of $\mathcal{H}$ for all $j \in N$ and arrange the terms, then we get $\sum_{j\in N} \hat{x}_j + \sum_{j\in N}(1-\sum_{i\in N} \rho_{j,i})\, \hat{z}_j = \sum_{j\in N} \lambda_j$. Similarly, adding the BALANCE equations for all $j \in N$, we get $\sum_{j\in N} \lambda_j = \sum_{j\in N} P_{j,S_{\hat{x}}} + \sum_{j\in N}(1-\sum_{i\in N} \rho_{j,i})\, R_{j,S_{\hat{x}}}$. Thus, we obtain $\sum_{j\in N} \hat{x}_j + \sum_{j\in N}(1-\sum_{i\in N} \rho_{j,i})\, \hat{z}_j = \sum_{j\in N} P_{j,S_{\hat{x}}} + \sum_{j\in N}(1-\sum_{i\in N} \rho_{j,i})\, R_{j,S_{\hat{x}}}$. Since we have $\hat{x}_j \geq P_{j,S_{\hat{x}}}$ and $\hat{z}_j \geq R_{j,S_{\hat{x}}}$ for all $j \in N$, the last equality implies that $\hat{x}_j = P_{j,S_{\hat{x}}}$ and $\hat{z}_j = R_{j,S_{\hat{x}}}$ for all $j \in N$. Thus, we can express $(\hat{x}, \hat{z})$ simply as $\hat{x} = P_{S_{\hat{x}}}$ and $\hat{z} = R_{S_{\hat{x}}}$ and we are done. As a side note, if $\alpha_{\hat{x}} > 1$, then we have $\hat{x}_j > P_{j,S_{\hat{x}}}$ for all $j \in N$. Using the fact that $\hat{z}_j \geq R_{j,S_{\hat{x}}}$ for all $j \in N$, we get a contradiction to the last equality. So, the largest possible value of $\alpha_{\hat{x}}$ is one.

In the third case, we assume that $\alpha_{\hat{x}} < 1$. We define $(\hat{u}, \hat{v})$ as $\hat{u}_j = (\hat{x}_j - \alpha_{\hat{x}}\, P_{j,S_{\hat{x}}})/(1-\alpha_{\hat{x}})$ and $\hat{v}_j = (\hat{x}_j - \alpha_{\hat{x}}\, R_{j,S_{\hat{x}}})/(1-\alpha_{\hat{x}})$ for all $j \in N$. In this case, by the definition of $(\hat{u}, \hat{v})$, we have $\hat{x} = \alpha_{\hat{x}}\, P_{S_{\hat{x}}} + (1-\alpha_{\hat{x}})\, \hat{u}$ and $\hat{z} = \alpha_{\hat{x}}\, R_{S_{\hat{x}}} + (1-\alpha_{\hat{x}})\, \hat{v}$. We define $S_{\hat{u}} = \big\{j \in N : \hat{u}_j > 0\big\}$ to capture the nonzero components of $\hat{u}$. In the next lemma, we show that $(\hat{u}, \hat{v})$ defined above satisfies $(\hat{u}, \hat{v}) \in \mathcal{H}$ and the nonzero components of $\hat{u}$ captured by $S_{\hat{u}}$ is a strict subset of the nonzero components of $\hat{x}$ captured by $S_{\hat{x}}$. This result implies that $(\hat{x}, \hat{z})$ can be expressed as a convex combination of $(P_{S_{\hat{x}}}, R_{\hat{x}})$ and $(\hat{u}, \hat{v})$, where $(\hat{u}, \hat{v}) \in \mathcal{H}$ and the set of nonzero components of $\hat{u}$ is a strict subset of the set of nonzero components of $\hat{x}$. If we have $S_{\hat{u}} = \varnothing$, then we can follow the argument in the first case above to show that $(\hat{u}, \hat{v}) = (P_{\varnothing}, R_{\varnothing})$, in which case, we are done since we can express $(\hat{x}, \hat{z})$ as a convex combination of $(P_{S_{\hat{x}}}, R_{S_{\hat{x}}})$ and $(P_{\varnothing}, R_{\varnothing})$. On the other hand, defining $\alpha_{\hat{u}} = \big\{\hat{u}_j \,/P_{j,S_{\hat{u}}} : j \in S_{\hat{u}}\big\}$, if we have $\alpha_{\hat{u}} \geq 1$, then we can follow the argument in the second case above to show that $(\hat{u}, \hat{v}) = (P_{S_{\hat{u}}}, R_{S_{\hat{u}}})$. In this case, we are done since we can express $(\hat{x}, \hat{z})$ as a convex combination of $(P_{S_{\hat{x}}}, R_{S_{\hat{x}}})$ and $(P_{S_{\hat{u}}}, R_{S_{\hat{u}}})$. Finally, if we have $\alpha_{\hat{u}} < 1$, then we can use the argument in the third case above

to show that $(\hat{u}, \hat{v})$ can be expressed as a convex combination of $(P_{S_{\hat{u}}}, R_{\hat{u}})$ and $(\hat{p}, \hat{q})$, where $(\hat{p}, \hat{q}) \in \mathcal{H}$ and the set of nonzero components of $\hat{p}$ is a strict subset of the set of nonzero components of $\hat{u}$. Repeating the same argument recursively, the process stops after a finite number of iterations since the vector $\hat{p}$ has strictly fewer nonzero components than $\hat{u}$, which has, in turn, strictly fewer nonzero components than $\hat{x}$.

Intuitively, the first case above corresponds to the situation where the optimal solution to the CHOICE BASED linear program only offers the empty subset. The second case above corresponds to the situation where the optimal solution to the CHOICE BASED linear program only offers the subset of products $S_{\hat{x}} = \{j \in N : \hat{x}_j > 0\}$. The third case above corresponds to the situation where the optimal solution to the CHOICE BASED linear program offers the subset of products $S_{\hat{x}} = \{j \in N : \hat{x}_j > 0\}$, along with other subsets that can only include the products in $S_{\hat{u}} = \{j \in N : \hat{u}_j > 0\}$. Thus, since $S_{\hat{u}}$ is a strict subset of $S_{\hat{x}}$, the third case above corresponds to the situation where the optimal solution to the CHOICE BASED linear program offers the subset of products $S_{\hat{x}} = \{j \in N : \hat{x}_j > 0\}$, along with other subsets that are strict subsets of $S_{\hat{x}}$. In the next lemma, we show that $(\hat{u}, \hat{v})$, as defined at the beginning of the previous paragraph, indeed satisfies $(\hat{u}, \hat{v}) \in \mathcal{H}$ and $\hat{u}$ has strictly fewer nonzero components than $\hat{x}$.

Lemma 9. *For $(\hat{x}, \hat{z}) \in \mathcal{H}$, define $S_{\hat{x}} = \{j \in N : \hat{x}_j > 0\}$, $\alpha_{\hat{x}} = \min\{\hat{x}_j / P_{j,S_{\hat{x}}} : j \in S_{\hat{x}}\}$ and $j_{\hat{x}} = \arg\min\{\hat{x}_j / P_{j,S_{\hat{x}}} : j \in S_{\hat{x}}\}$. Assuming that $\alpha_{\hat{x}} < 1$, define $(\hat{u}, \hat{v})$ as*

$$\hat{u}_j = \frac{\hat{x}_j - \alpha_{\hat{x}} P_{j,S_{\hat{x}}}}{1 - \alpha_{\hat{x}}} \quad \text{and} \quad \hat{v}_j = \frac{\hat{z}_j - \alpha_{\hat{x}} R_{j,S_{\hat{x}}}}{1 - \alpha_{\hat{x}}}$$

for all $j \in N$ and $S_{\hat{u}} = \{j \in N : \hat{u}_j > 0\}$. Then, we have $(\hat{u}, \hat{v}) \in \mathcal{H}$ and $S_{\hat{u}} \subset S_{\hat{x}} \setminus \{j_{\hat{x}}\}$.

Proof. First, we show that $(\hat{u}, \hat{v}) \in \mathcal{H}$. We observe that $(\hat{u}, \hat{v})$ is obtained by multiplying $(\hat{x}, \hat{z})$ by $1/(1-\alpha_{\hat{x}})$ and $(P_{S_{\hat{x}}}, R_{S_{\hat{x}}})$ by $-\alpha_{\hat{x}}/(1-\alpha_{\hat{x}})$ and adding them up. Using the fact that $(\hat{x}, \hat{z}) \in \mathcal{H}$ and noting the BALANCE equations, we have $\hat{x}_j + \hat{z}_j = \lambda_j + \sum_{i \in N} \rho_{i,j}\, \hat{z}_i$ and $P_{j,S_{\hat{x}}} + R_{j,S_{\hat{x}}} = \lambda_j + \sum_{i \in N} \rho_{i,j}\, R_{i,S_{\hat{x}}}$ for all $j \in N$. Multiplying these two equations respectively by $1/(1-\alpha_{\hat{x}})$ and $-\alpha_{\hat{x}}/(1-\alpha_{\hat{x}})$ and adding, it follows that $(\hat{u}, \hat{v})$ satisfies $\hat{u}_j + \hat{v}_j = \lambda_j + \sum_{i \in N} \rho_{i,j}\, \hat{v}_i$ for all $j \in N$. Thus, to show that $(\hat{u}, \hat{v}) \in \mathcal{H}$, it is enough to check that $\hat{u}_j \geq 0$ and $\hat{v}_j \geq 0$ for all $j \in N$. Since $\alpha_{\hat{x}} \leq \hat{x}_j / P_{j,S_{\hat{x}}}$ for all $j \in S_{\hat{x}}$, we have $\hat{u}_j = (\hat{x}_j - \alpha_{\hat{x}}\, P_{j,S_{\hat{x}}})/(1-\alpha_{\hat{x}}) \geq 0$ for all $j \in S_{\hat{x}}$. By definition, we have $\hat{x}_j = 0 = P_{j,S_{\hat{x}}}$ for all $j \notin S_{\hat{x}}$. Thus, we also have $\hat{u}_j = (\hat{x}_j - \alpha_{\hat{x}}\, P_{j,S_{\hat{x}}})/(1-\alpha_{\hat{x}}) = 0$ for all $j \notin S_{\hat{x}}$. By Lemma 22 in Appendix A.4, we have $\hat{z}_j \geq R_{j,S_{\hat{x}}}$ for all $j \in N$. Using the assumption that $\alpha_{\hat{x}} < 1$, we have $\hat{v}_j = (\hat{z}_j - \alpha_{\hat{x}}\, R_{j,S_{\hat{x}}})/(1-\alpha_{\hat{x}}) \geq 0$ for all $j \in N$ and the first part of the lemma follows. Second, we show that $S_{\hat{u}} \subset S_{\hat{x}} \setminus \{j_{\hat{x}}\}$. Consider $j \in S_{\hat{u}}$ so that $\hat{u}_j > 0$. Since $\hat{u}_j = (\hat{x}_j - \alpha_{\hat{x}}\, P_{j,S_{\hat{x}}})/(1-\alpha_{\hat{x}}) > 0$, it must be the case that $\hat{x}_j > 0$, indicating that $j \in S_{\hat{x}}$. So, we get $S_{\hat{u}} \subset S_{\hat{x}}$. Furthermore, we have $j_{\hat{x}} \notin S_{\hat{u}}$ since $u_{j_{\hat{x}}} = (\hat{x}_{j_{\hat{x}}} - \alpha_{\hat{x}}\, P_{j_{\hat{x}},S_{\hat{x}}})/(1-\alpha_{\hat{x}}) = 0$, where the second equality uses the fact that $\alpha_{\hat{x}} = \hat{x}_{j_{\hat{x}}} / P_{j_{\hat{x}},S_{\hat{x}}}$. Thus, we obtain $S_{\hat{u}} \subset S_{\hat{x}} \setminus \{j_{\hat{x}}\}$. □

Building on the discussion that we have so far in this section, we propose the following algorithm to find subsets $S^1, \ldots, S^K$ and positive scalars $\gamma^1, \ldots, \gamma^K$ summing to one such that the optimal solution $(\hat{x}, \hat{z})$ to the REDUCED linear program can be expressed as $\hat{x} = \sum_{k=1}^{K} \gamma^k\, P_{S^k}$ and $\hat{z} = \sum_{k=1}^{K} \gamma^k\, R_{S^k}$. Thus, we can use this algorithm to recover the optimal solution to the CHOICE BASED linear program from the optimal solution to the REDUCED linear program.

DIMENSION REDUCTION

STEP 0. Using $(\hat{x}, \hat{z})$ to denote the optimal solution to the REDUCED linear

program, set $(x^1, z^1) = (\hat{x}, \hat{z})$. Initialize the iteration counter by setting $k = 1$.

STEP 1. Set $S^k = \{j \in N : x_j^k > 0\}$. If $S^k = \varnothing$, then set $\alpha^k = 1$ and stop.

STEP 2. Set $\alpha^k = \min\{x_j^k / P_{j,S^k} : j \in S^k\}$. If $\alpha^k \geq 1$, then stop.

STEP 3. Set (x^{k+1}, z^{k+1}) as $x_j^{k+1} = (x_j^k - \alpha^k P_{j,S^k})/(1 - \alpha^k)$ and $z_j^{k+1} = (z_j^k - \alpha^k R_{j,S^k})/(1 - \alpha^k)$ for all $j \in N$. Increase k by one and go to Step 1.

Steps 1, 2 and 3 in the DIMENSION REDUCTION algorithm respectively correspond to the first, second and third cases considered at the beginning of this section. We can use induction over the iterations of the algorithm to show that $(x^k, z^k) \in \mathcal{H}$ at each iteration k. In particular, since $(\hat{x}, \hat{z})$ is a feasible solution to the REDUCED linear program, we have $(x^1, z^1) \in \mathcal{H}$. Assuming that $(x^k, z^k) \in \mathcal{H}$, if we identify (x^k, z^k) and (x^{k+1}, z^{k+1}) in the algorithm respectively with $(\hat{x}, \hat{z})$ and $(\hat{u}, \hat{v})$ in Lemma 9, then this lemma implies that $(x^{k+1}, z^{k+1}) \in \mathcal{H}$, completing the induction. On the other hand, letting $j^k = \arg\min\{x_j^k / P_{j,S^k} : j \in S^k\}$ and noting that $(x^k, z^k) \in \mathcal{H}$, if we identify S^k and S^{k+1} in the algorithm with $S_{\hat{x}}$ and $S_{\hat{u}}$ in Lemma 9, then this lemma implies that $S^{k+1} \subset S^k \setminus \{j^k\}$. Therefore, x^{k+1} has strictly fewer nonzero components than x^k. This observation implies that the algorithm stops after at most $n + 1$ iterations with $S^{n+1} = \varnothing$. In the next theorem, we show that we can use the DIMENSION REDUCTION algorithm to find subsets $S^1, \ldots, S^K$ and positive scalars $\gamma^1, \ldots, \gamma^K$ summing to one such that the optimal solution $(\hat{x}, \hat{z})$ to the REDUCED linear program can be expressed as $\hat{x} = \sum_{k=1}^K \gamma^k P_{S^k}$ and $\hat{z} = \sum_{k=1}^K \gamma^k R_{S^k}$.

Theorem 10. *Assume that the* DIMENSION REDUCTION *algorithm stops at iteration K, generating the subsets $S^1, \ldots, S^K$ and the scalars $\alpha^1, \ldots, \alpha^K$. Then, letting $\gamma^k = (1 - \alpha^1) \ldots (1 - \alpha^{k-1}) \alpha^k$ for all $k = 1, \ldots, K$, we have $\hat{x} = \sum_{k=1}^K \gamma^k P_{S^k}$, $\hat{z} = \sum_{k=1}^K \gamma^k R_{S^k}$ and $\sum_{k=1}^K \gamma^k = 1$.*

Proof. We use induction over the iterations of the DIMENSION REDUCTION algorithm to show that $\hat{x} = \gamma^1 P_{S^1} + \ldots + \gamma^{k-1} P_{S^{k-1}} + (\gamma^k/\alpha^k)\, x^k$ for all $k = 1, \ldots, K$. Since $\gamma^1 = \alpha^1$, we have $\hat{x} = x^1 = (\gamma^1/\alpha^1)\, x^1$, so that the result holds at the first iteration. We assume that the result holds at iteration k. Noting Step 3 of the algorithm, we have $x^k = \alpha^k P_{S^k} + (1 - \alpha^k)\, x^{k+1}$. In this case, since $\hat{x} = \gamma^1 P_{S^1} + \ldots + \gamma^{k-1} P_{S^{k-1}} + (\gamma^k/\alpha^k)\, x^k$ by the induction assumption, we have

$$\begin{aligned}\hat{x} = \gamma^1 P_{S^1} + \ldots + \gamma^{k-1} P_{S^{k-1}} + \frac{\gamma^k}{\alpha^k}\left(\alpha^k P_{S^k} + (1 - \alpha^k)\, x^{k+1}\right)& \\ = \gamma^1 P_{S^1} + \ldots + \gamma^k P_{S^k} + \frac{\gamma^{k+1}}{\alpha^{k+1}}\, x^{k+1},&\end{aligned}$$

where the second equality uses the fact that $(\gamma^k/\alpha^k)\,(1 - \alpha^k) = \gamma^{k+1}/\alpha^{k+1}$ by the definition of γ^k. Therefore, the result holds at iteration $k + 1$ and the induction is complete. A similar argument also shows that $\hat{z} = \gamma^1 R_{S^1} + \ldots + \gamma^{k-1} R_{S^{k-1}} + (\gamma^k/\alpha^k)\, z^k$. On the other hand, we note that the algorithm stops in either Step 1 or Step 2. If the algorithm stops in Step 1, then $S^K = \varnothing$ and $\alpha^K = 1$. By the discussion that follows the description of the DIMENSION REDUCTION algorithm, we have $(x^K, z^K) \in \mathcal{H}$, in which case, we can follow the same argument in the first case at the beginning of this section to conclude that $(x^K, z^K) = (P_\varnothing, R_\varnothing)$. Thus, we obtain $(x^K, z^K) = (P_{S^K}, R_{S^K})$. If the algorithm stops in Step 2, then $\alpha^K \geq 1$ and we can follow the same argument in the second case at the beginning of this section to conclude that $(x^K, z^K) = (P_{S^K}, R_{S^K})$. Furthermore, if the algorithm stops in Step 2, then the discussion in the second case at the beginning of this section shows that $\alpha^K \leq 1$. Thus, we must have $\alpha^K = 1$. Therefore, we always have $(x^K, z^K) = (P_{S^K}, R_{S^K})$ and $\alpha^K = 1$ when the DIMENSION REDUCTION algorithm stops. In this case, using the equalities $\hat{x} = \gamma^1 P_{S^1} + \ldots + \gamma^{k-1} P_{S^{k-1}} + (\gamma^k/\alpha^k)\, x^k$ and $\hat{z} = \gamma^1 R_{S^1} + \ldots + \gamma^{k-1} R_{S^{k-1}} + (\gamma^k/\alpha^k)\, z^k$ with $k = K$, we obtain $\hat{x} = \sum_{k=1}^K \gamma^k P_{S^k}$ and $\hat{z} = \sum_{k=1}^K \gamma^k R_{S^k}$. Finally, noting that $\alpha^K = 1$ and using the definition of γ^k in the theorem, it follows that $\gamma^{K-1} + \gamma^K = (1 - \alpha^1) \ldots (1 - \alpha^{K-2})$. Repeating

the same argument recursively, we obtain $\gamma^2 + \ldots + \gamma^K = (1 - \alpha^1)$, in which case, since $\gamma^1 = \alpha^1$ by the definition of γ^k, we get $\gamma^1 + \gamma^2 + \ldots + \gamma^K = 1$. $\square$

By Theorem 10, we can use the DIMENSION REDUCTION algorithm to find subsets $S^1, \ldots, S^K$ and positive scalars $\gamma^1, \ldots, \gamma^K$ summing to one such that the optimal solution $(\hat{x}, \hat{z})$ to the REDUCED linear program can be expressed as $\hat{x} = \sum_{k=1}^K \gamma^k P_{S^k}$ and $\hat{z} = \sum_{k=1}^K \gamma^k R_{S^k}$. In this case, Theorem 8 implies that the solution $\hat{u}$ obtained by letting $\hat{u}_{S^k} = \gamma^k$ for all $k = 1, \ldots, K$ and $\hat{u}_S = 0$ for all $S \notin \left\{S^1, \ldots, S^K\right\}$ is optimal to the CHOICE BASED linear program. These observations provide a tractable approach for recovering the optimal solution to the CHOICE BASED linear program by using the optimal solution to the REDUCED linear program. In the next theorem, we use the DIMENSION REDUCTION algorithm to show structural properties of the subsets of products offered by the optimal solution to the CHOICE BASED linear problem. In particular, we show that the optimal solution to the CHOICE BASED linear program offers at most $n + 1$ different subsets and these subsets are nested in the sense that one of the subsets is included in another one, naturally with the exception of the largest subset.

Theorem 11. *There exists an optimal solution $\hat{u}$ to the* CHOICE BASED *linear program, where at most $n+1$ of the decision variables can take on nonzero values. Furthermore, letting $\hat{u}_{S^1}, \ldots, \hat{u}_{S^{n+1}}$ be these decision variables, we have $S^1 \supset \ldots \supset S^{n+1}$.*

Proof. Letting the subsets $S^1, \ldots, S^K$ be generated by the DIMENSION REDUCTION algorithm, Theorems 8 and 10 imply that there exists an optimal solution $\hat{u}$ to the CHOICE BASED linear program, where we have $\hat{u}_{S^k} \geq 0$ for all $k = 1, \ldots, K$ and $\hat{u}(S) = 0$ for all $S \notin \left\{S^1, \ldots, S^K\right\}$. By discussion that follows the description of the DIMENSION REDUCTION algorithm, the algorithm stops after at most $n+1$

iterations, which implies that $K \leq n+1$ and the first part of the theorem follows. By the discussion that follows the description of the DIMENSION REDUCTION algorithm, we have $S^k \supset S^{k+1}$ for all $k = 1, \ldots, K-1$ and the second part of the theorem follows. $\square$

One of the practical implications of Theorem 11 is that the optimal solution to the CHOICE BASED linear program does not randomize over offering too many subsets and the offered subsets are related in the sense that one subset is included in another one. Randomizing over nested subsets also avoids offering drastically different subsets of products to customers.

The DIMENSION REDUCTION algorithm has an appealing geometric interpretation. To give this interpretation, we recall that Lemma 2 gives a connection between (P_S, R_S) and the extreme points of $\mathcal{H}$. In particular, for any extreme point $(\hat{x}, \hat{z})$ of $\mathcal{H}$, if we let $S_{\hat{x}} = \{j \in N : \hat{x}_j > 0\}$, then we have $P_{S_{\hat{x}}} = \hat{x}$ and $R_{S_{\hat{x}}} = \hat{z}$. Treating $\{x_j : j \in N\}$ as the slack variables, we write $\mathcal{H}$ equivalently as $\mathcal{G} = \{z \in \Re^n_+ : z_j \leq \lambda_j + \sum_{i \in N} \rho_{i,j} z_i \; \forall \, j \in N\}$. The polytopes $\mathcal{H}$ and $\mathcal{G}$ are equivalent in the sense that for any $z \in \mathcal{G}$, if we define $x \in \Re^n_+$ as $x_j = \lambda_j + \sum_{i \in N} \rho_{i,j} z_i - z_j$ for all $j \in N$, then we have $(x, z) \in \mathcal{H}$. To give an interpretation for the DIMENSION REDUCTION algorithm, we focus on the case where $n = 2$ so that $N = \{1, 2\}$. For the case where $n = 2$, we show the polytope $\mathcal{G}$ in Figure 2.1. For the extreme point $z^a = (z^a_1, z^a_2)$ of $\mathcal{G}$ in Figure 2.1, we have $z^a_1 = z^a_2 = 0$. Thus, we obtain $x^a_j = \lambda_j + \sum_{i \in N} \rho_{i,j} z^a_i - z^a_j > 0$ for all $j = 1, 2$, where we use the fact that $\lambda_j > 0$ and $z^a_j = 0$ for all $j = 1, 2$. In this case, defining the subset $S^a = \{j \in N : x^a_j > 0\}$, we have $S^a = \{j \in N : x^a_j > 0\} = \{1, 2\}$. Since z^a is an extreme point of $\mathcal{G}$, it is straightforward to check that (x^a, z^a) is an extreme point of $\mathcal{H}$, in which case, Lemma 2 implies that $P_{S^a} = x^a$ and $R_{S^a} = z^a$. Similarly, for the extreme point

$z^b = (z_1^b, z_2^b)$ in Figure 2.1, we have $z_1^b = 0$ and $z_2^b = \lambda_2 + \sum_{i \in N} \rho_{i,2}\, z_i^b$. So, we obtain $x_1^b = \lambda_1 + \sum_{i \in N} \rho_{i,1}\, z_i^b - z_1^b > 0$ and $x_2^b = \lambda_2 + \sum_{i \in N} \rho_{i,2}\, z_i^b - z_2^b = 0$. In this case, defining the subset $S^b = \big\{j \in N : x_j^b > 0\big\}$, we have $S^b = \big\{j \in N : x_j^b > 0\big\} = \big\{1\big\}$ and Lemma 2 implies that $P_{S^b} = x^b$ and $R_{S^b} = z^b$. Finally, for the extreme point $z^c = (z_1^c, z_2^c)$ in Figure 2.1, we have $z_j^c = \lambda_j + \sum_{i \in N} \rho_{i,j}\, z_i^c$ for all $j = 1, 2$. Therefore, we obtain $x_j^c = \lambda_j + \sum_{i \in N} \rho_{i,j}\, z_i^c - z_j^c = 0$ for all $j = 1, 2$. In this case, if we define the subset $S^c = \big\{j \in N : x_j^c > 0\big\}$, then we have $S^c = \big\{j \in N : x_j^c > 0\big\} = \varnothing$ and Lemma 2 implies that $P_{S^c} = x^c$ and $R_{S^c} = z^c$.

The discussion above shows that $S^a = \big\{1, 2\big\} \supset S^b = \big\{1\big\} \supset S^c = \varnothing$. Furthermore, we have $R_{S^a} = z^a$, $R_{S^b} = z^b$ and $R_{S^c} = z^c$. Letting $(\hat{x}, \hat{z})$ be the optimal solution to the REDUCED linear program, the second constraint in this problem implies that $\hat{z} \in \mathcal{G}$. We show the point $\hat{z}$ in Figure 2.1. We observe that $\hat{z}$ can be expressed as $\hat{z} = \alpha^a\, z^a + (1 - \alpha^a)\, \tilde{z}$ for some $\alpha^a \in [0, 1]$ and $\tilde{z} \in \mathcal{G}$, where the point $\tilde{z}$ is as shown in Figure 2.1. On the other hand, $\tilde{z}$ can be expressed as $\tilde{z} = \alpha^b\, z^b + (1 - \alpha^b)\, z^c$ for some $\alpha^b \in [0, 1]$. Therefore, $\hat{z}$ can be expressed as $\hat{z} = \alpha^a\, z^a + (1 - \alpha^a)\, (\alpha^b\, z^b + (1 - \alpha^b)\, z^c) = \alpha^a\, R_{S^a} + (1 - \alpha^a)\, \alpha^b\, R_{S^b} + (1 - \alpha^a)\, (1 - \alpha^b)\, R_{S^c}$. Thus, letting $\gamma^a = \alpha^a$, $\gamma^b = (1 - \alpha^a)\, \alpha^b$ and $\gamma^c = (1 - \alpha^a)\, (1 - \alpha^b)$, it follows that $\gamma^a + \gamma^b + \gamma^c = 1$ and $\hat{z} = \gamma^a\, R_{S^a} + \gamma^b\, R_{S^b} + \gamma^c\, R_{S^c}$, indicating that $\hat{z}$ can be expressed as a convex combination of R_{S^a}, R_{S^b} and R_{S^c}. To sum up, by connecting the points z^a and $\hat{z}$ through a line segment and finding the intersection point of this line segment with a face of $\mathcal{G}$, we can identify the point $\tilde{z}$ so that we can express $\hat{z}$ as a convex combination of z^a and $\tilde{z}$. Since $S^a = \big\{1, 2\big\}$, expressing $\hat{z}$ as a convex combination of z^a and $\tilde{z}$ implies that the optimal solution to the REDUCED linear program offers the subset $\big\{1, 2\big\}$, along with some other subsets. From this point on, we focus on a smaller dimensional space characterized by only one face of $\mathcal{G}$ that is given by $\lambda_2 + \sum_{i \in N} \rho_{i,2}\, z_i - z_2 = 0$. Since we

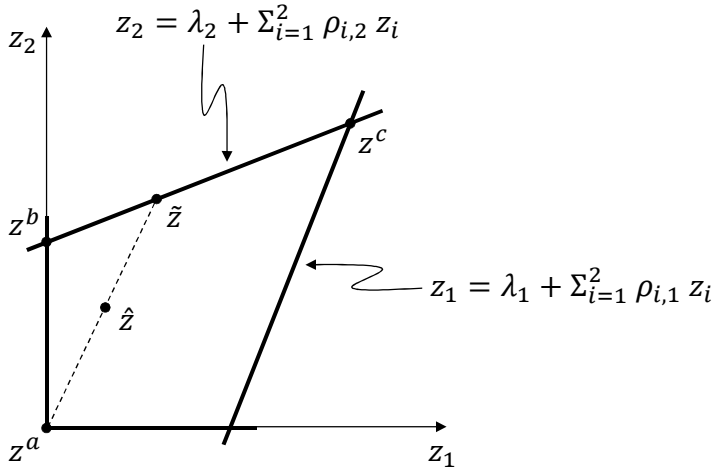


Figure 2.1: The polytope $\mathcal{G} = \left\{ z \in \Re_+^n : z_j \leq \lambda_j + \sum_{i \in N} \rho_{i,j}\, z_i \;\forall\, j \in N \right\}$ for the case where $n = 2$.

have $x_2 = \lambda_2 + \sum_{i \in N} \rho_{i,2}\, z_i - z_2 = 0$ on this face, we can drop product 2 from consideration and only focus on offering other subsets that do not include product 2. Following a similar argument, we can show that if $(\hat{x}, \hat{z})$ is the optimal solution to the REDUCED linear program, then $\hat{x}$ can also be expressed as a convex combination of P_{S^a}, P_{S^b} and P_{S^c} as well. The DIMENSION REDUCTION follows the same approach to express $(\hat{x}, \hat{z})$ as a convex combination of the extreme points of $\mathcal{H}$. If we let $j^k = \arg\min\left\{ x_j^k \,/ P_{j,S^k} : j \in S^k \right\}$ in Step 2 of the DIMENSION REDUCTION algorithm, then we drop product j^k from consideration at iteration k and focus on offering other subsets that do not include product j^k. In this way, the DIMENSION REDUCTION algorithm finds nested subsets $S^1 \supset \ldots \supset S^K$ and positive scalars $\gamma^1, \ldots, \gamma^K$ summing to one such that the optimal solution $(\hat{x}, \hat{z})$ to the REDUCED linear program can be expressed as $\hat{x} = \sum_{k=1}^{K} \gamma^k\, P_{S^k}$ and $\hat{z} = \sum_{k=1}^{K} \gamma^k\, R_{S^k}$.

2.7 Capturing Customer Choices through the Markov Chain Choice Model

In this section, we provide computational experiments to demonstrate that if we fit a Markov chain choice model to the past purchase history of the customers and make our decisions based on the fitted Markov chain choice model, then we may make noticeably better decisions when compared with the case where we use a simpler choice model, such as the multinomial logit model.

2.7.1 Experimental Setup

In our computational experiments, we generate the past purchase history of the customers under the assumption that the customers choose among the offered products according to a complicated ground choice model that does not comply with the Markov chain choice model or the multinomial logit model. We fit both a Markov chain choice model and a multinomial logit model to the past purchase history of the customers and compare the two fitted choice models by using three approaches. First, we check how well the two fitted choice models allow us to predict the purchases of the customers that are not in the past purchase history. Second, we compute the optimal subsets in the 3.1 problem under the assumption that the customers choose according to either one of the two fitted choice models and check how much expected revenue we obtain by offering these subsets. Third, we compute the optimal policies in the Single Resource dynamic program under the assumption that the customers choose according to either one of the two fitted

choice models and check how much total expected revenue we obtain by using these policies.

In the ground choice model that governs the customer choices, there are m customer types indexed by $M = \{1, \ldots, m\}$. A customer of a particular type ranks the products and the no purchase option according to a particular order and purchases the offered product that has the highest rank in this order. If the ranking of none of the offered products is higher than the ranking of the no purchase option, then the customer leaves without a purchase. Thus, each customer type in the ground choice model is associated with a particular order of the products and the no purchase option. We use β^g to denote the probability that a customer of type g arrives into the system. In the order that a customer of type g associates with the products and the no purchase option, σ_j^g is the ranking of product j and σ_0^g is the ranking of the no purchase option. [6] study this choice model and discuss that this choice model captures any choice model that is based on the random utility maximization principle. In our computational experiments, to generate the customer types, we set $\beta^g = 1/m$ so that each customer type arrives with equal probability. For each customer type g, we set $(\sigma_0^g, \sigma_1^g, \ldots, \sigma_n^g)$ to a permutation of the integers $\{1, \ldots, n+1\}$ sampled uniformly over all possible permutations.

Once we generate the ground choice model that governs the choice process of the customers, we generate the past purchase history of a number of customers under the assumption that the customers choose among the offered products according to this ground choice model. We capture the past purchase history for τ customers by using $\{(S_t, j_t) : t = 1, \ldots, \tau\}$, where S_t is the subset of products offered to customer t and j_t is the product purchased by customer t. If customer t leaves without making a purchase, then we have $j_t = 0$. In our computational experiments, there

are five customer types and the number of products is $n = 10$. We use the following approach to generate the past purchase history of τ customers. We sample the subset S_t such that each product is included in this subset with probability 0.5. In our ground choice model, a customer is of type g with probability β^g. Using g_t to denote the type of customer t in the past purchase history, we sample g_t by using the probabilities $(\beta^1, \ldots, \beta^m)$. In this case, the product j_t purchased by customer t is the highest ranking product in the subset S_t of offered products when the customer sorts the products according to the rankings $(\sigma_0^{g_t}, \sigma_1^{g_t}, \ldots, \sigma_n^{g_t})$. If none of the products in S_t has a ranking higher than the ranking of the no purchase option, then the customer leaves without a purchase and we set $j_t = 0$. We vary the number of customers in the past purchase history over $\tau \in \{1000, 1750, 2500\}$, which allows us to check our ability to fit a Markov chain choice model and a multinomial logit model with different levels of data availability.

2.7.2 Fitting Choice Models

We fit a Markov chain choice model and a multinomial logit model to the past purchase history of the customers and compare the two fitted choice models through various performance measures. Our approach for fitting a Markov chain choice model and a multinomial logit model to the past purchase history of the customers uses the following three steps. (1) We generate the ground choice model that governs the choice process of the customers by using the approach described in the previous section. Throughout our computational experiments, the ground choice model that governs the choice process of the customers is fixed. (2) Using the approach described in the previous section, we generate the past purchase history for τ customers when the customers choose according to the ground choice

model. We use H_τ to denote the past purchase history for τ customers. We generate three sets of past purchase histories by varying the number of customers over $\tau \in \big\{1000, 1750, 2500\big\}$, corresponding to three different levels of data availability when we fit a Markov chain choice model and a multinomial logit model. We refer to each one of these past purchase histories as a training data set. (3) We use maximum likelihood estimation to fit a Markov chain choice model and a multinomial logit model to each one of the training data sets. We use MC_τ and ML_τ to respectively denote the Markov chain choice model and the multinomial logit model that are fitted to the training data set H_τ. Our ultimate goal is to compare the fitted choice models MC_τ and ML_τ for $\tau \in \big\{1000, 1750, 2500\big\}$ through various performance measures.

There are computational challenges when we fit a Markov chain choice model to the past purchase history. To describe these computational challenges, we use $\big\{(S_t, j_t) : t = 1, \ldots, \tau\big\}$ to denote the subsets of products offered to the customers and the products purchased by the customers in the training data set H_τ. Given that customers choose according to the Markov chain choice model characterized by the parameters $\lambda = (\lambda_1, \ldots, \lambda_n)$ and $\rho = \big\{\rho_{j,i} : j, i \in N\big\}$, we use $P_{j,S}(\lambda, \rho)$ to denote the probability that a customer purchases product j when we offer the subset S of products. When the customers choose according to the Markov chain choice model characterized by the parameters λ and ρ, the log likelihood of the training data set H_τ can be written as $\mathcal{L}(\lambda, \rho \,|\, H_\tau) = \sum_{t=1}^{\tau} \log P_{j_t,S_t}(\lambda, \rho)$. There is no closed form expression for the purchase probability $P_{j,S}(\lambda, \rho)$, which implies that there is no closed form expression for the log likelihood $\mathcal{L}(\lambda, \rho \,|\, H_\tau)$. However, although there is no closed form expression for the purchase probability $P_{j,S}(\lambda, \rho)$, we can use the Balance equations to compute the purchase probability $P_{j,S}(\lambda, \rho)$ for any λ and ρ, which implies that we can compute the log likelihood $\mathcal{L}(\lambda, \rho \,|\, H_\tau)$

for any λ and ρ. Thus, to obtain the maximum likelihood estimators of λ and ρ, we can maximize the log likelihood $\mathcal{L}(\lambda, \rho \,|\, H_\tau)$ by using optimization software that only needs evaluating the value of the objective function. We use `fmincon` routine in `Matlab` for this purpose. Optimization algorithms that only need evaluating the value of the objective function can be less effective than their counterparts that need evaluating the derivatives of the objective function. Furthermore, the log likelihood $\mathcal{L}(\lambda, \rho \,|\, H_\tau)$ is not necessarily concave in λ and ρ. Therefore, we emphasize that the approach that we use to obtain the maximum likelihood estimators of λ and ρ is not entirely sophisticated, but this approach turns out to be adequate to demonstrate the benefits from fitting a Markov chain choice model, instead of a simpler choice model such as the multinomial logit model. It is also worthwhile to emphasize that our main focus in this paper is on solving various revenue management problems under the Markov chain choice model and effective methods for obtaining the maximum likelihood estimators of the parameters of the Markov chain choice model is an important research question that remains open for further investigation.

2.7.3 Predicting Customer Purchases

By using the approach described in the previous section, we fit a Markov chain choice model and a multinomial logit model to each one of the three training data sets. We recall that we use MC_τ and ML_τ to respectively denote the Markov chain choice model and the multinomial logit model that are fitted to the training data set H_τ. One approach for comparing the two fitted choice models is based on checking how well the two fitted choice models predict the purchases of the customers that are not in the training data set. For this purpose, we generate the past purchase

history of another 2500 customers when the customers choose according to the ground choice model. We refer to this past purchase history as the testing data set. To check how well the two fitted choice models predict the purchases of the customers that are not in the training data set, we compute the log likelihood of the testing data set under the assumption that the customers choose according to either one of the two fitted choice models. A larger value for the log likelihood of the testing data set indicates that the corresponding fitted choice model predicts the purchases of the customers that are not in the training data set better. In this way, we check the out of sample performance of the two fitted choice models. To make sure that our results are relatively robust, we repeat our computational experiments for 10 different ground choice models that we generate.

Table 2.3 shows our computational results. Each row in Table 2.3 shows the results for each one of the 10 different ground choice models that we generate. There are three blocks of three consecutive columns in Table 2.3 and each of these three blocks corresponds to one of the three values for $\tau \in \{1000, 1750, 2500\}$, capturing different levels of data availability when we fit a Markov chain choice model and a multinomial logit model to the past purchase history. Given that each block of three columns corresponds to a particular value of τ, the first column in each block shows the log likelihood of the testing data set under the assumption that the customers choose according to the Markov chain choice model MC_τ estimated from the past purchase history H_τ, whereas the second column in each block shows the log likelihood of the testing data set under the assumption that the customers choose according to the multinomial logit model ML_τ estimated from the past purchase history H_τ. In particular, for any choice model $\mathsf{CM} \in \{\mathsf{MC}_\tau, \mathsf{ML}_\tau\}$, we use $P_{j,S}(\mathsf{CM})$ to denote the probability that a customer purchases product j when we offer the subset S of products, given that customers choose under the choice

Grn. Cho. Mod.	$\tau = 1000$ Like. MC_τ	Like. ML_τ	Perc. Gap	$\tau = 1750$ Like. MC_τ	Like. ML_τ	Perc. Gap	$\tau = 2500$ Like. MC_τ	Like. ML_τ	Perc. Gap
1	−3073	−3463	12.69	−3069	−3460	12.77	−3067	−3457	12.72
2	−3108	−3457	11.23	−3099	−3452	11.40	−3095	−3447	11.39
3	−3175	−3466	9.18	−3077	−3464	12.56	−3075	−3465	12.69
4	−3113	−3454	10.93	−3085	−3444	11.64	−3081	−3440	11.66
5	−2937	−3398	15.72	−2932	−3391	15.66	−2939	−3390	15.32
6	−3165	−3449	8.98	−3104	−3453	11.26	−3090	−3450	11.62
7	−3089	−3454	11.83	−3081	−3461	12.32	−3093	−3459	11.83
8	−3241	−3443	6.23	−3084	−3433	11.31	−3068	−3427	11.70
9	−3092	−3455	11.74	−3076	−3445	11.99	−3060	−3443	12.51
10	−3113	−3451	10.86	−3084	−3436	11.43	−3072	−3435	11.79
Avg.	−3110	−3449	10.94	−3069	−3444	12.23	−3064	−3441	12.32

Table 2.3: Log likelihood of the testing data set under the assumption that the customers choose according to the fitted choice models.

model CM. Furthermore, we use $\big\{(S_t, j_t) : t = 1, \ldots, 2500\big\}$ to denote the subsets of products offered to the customers and the products purchased by the customers in the testing data set. In this case, the first and second columns in each block shows the log likelihoods $\sum_{t=1}^{2500} \log P_{j_t,S_t}(\mathsf{MC}_\tau)$ and $\sum_{t=1}^{2500} \log P_{j_t,S_t}(\mathsf{ML}_\tau)$. Finally, the third column in each block shows the percent gap between the first and second columns.

The results in Table 2.3 indicate that the log likelihoods of the testing data set under the fitted Markov chain choice model are significantly larger than those under the fitted multinomial logit model, indicating that the fitted Markov chain choice model does a better job of predicting the purchases of the customers. It is worthwhile to observe that as the number of customers in the training data set increases and we have more data availability in the training data set, the log likelihood of the testing data set under the fitted Markov chain choice model increases more noticeably when compared with that under the fitted multinomial logit model. For example, the average log likelihood under the fitted Markov chain choice model increases from −3110 to −3064 as τ increases from 1000 to

2500, whereas the average log likelihood under the fitted multinomial logit model increases from -3449 to -3441. Markov chain choice model has $O(n^2)$ parameters corresponding to $(\lambda_1, \ldots, \lambda_n)$ and $\{\rho_{j,i} : j, i \in N\}$, whereas the multinomial logit model has $O(n)$ parameters, corresponding to the mean utilities of the products. As we increase the data availability, we can exploit the modeling flexibility of the Markov chain choice model and the improvement in the quality of the fitted Markov chain choice model is more significant, but the multinomial logit model quickly reaches the limits of its modeling flexibility and the improvement in the quality of the fitted multinomial logit model is less significant.

2.7.4 Finding Good Assortments

Another approach for comparing the Markov chain choice model MC_τ and the multinomial logit model ML_τ that are fitted to the training data set H_τ is based on checking how well the two fitted choice models allow us to solve the 3.1 problem. For this purpose, we generate 100 samples of the product revenues. We use $\{(r_1^k, \ldots, r_n^k) : k = 1, \ldots, 100\}$ to denote the samples of the product revenues, where we sample r_j^k from the uniform distribution over $[0, 100]$. For each sample of the product revenues, we compute the optimal subset in the 3.1 problem under the assumption the customers choose according to either one of the two fitted choice models. In particular, we let $S^k(\mathsf{MC}_\tau) = \arg\max_{S \subset N} \sum_{j \in N} P_{j,S}(\mathsf{MC}_\tau)\, r_j^k$ and $S^k(\mathsf{ML}_\tau) = \arg\max_{S \subset N} \sum_{j \in N} P_{j,S}(\mathsf{ML}_\tau)\, r_j^k$ so that $S^k(\mathsf{MC}_\tau)$ and $S^k(\mathsf{ML}_\tau)$ correspond to the optimal subsets when the product revenues are $(r_1^k, \ldots, r_n^k)$ and we solve the 3.1 problem under the assumption that the customers choose according to either one of the two fitted choice models. In the last two problems, for any choice model $\mathsf{CM} \in \{\mathsf{MC}_\tau, \mathsf{ML}_\tau\}$, the purchase probability $P_{j,S}(\mathsf{CM})$ is as de-

fined in the previous section. Since the customers actually choose according to the ground choice model, if we offer the subset $S^k(\mathsf{MC}_\tau)$ or $S^k(\mathsf{ML}_\tau)$ of products, then we obtain an actual expected revenue of $\mathsf{Rev}^k(\mathsf{MC}_\tau) = \sum_{j \in N} P_{j,S^k(\mathsf{MC}_\tau)}(\mathsf{GC})\, r_j^k$ or $\mathsf{Rev}^k(\mathsf{ML}_\tau) = \sum_{j \in N} P_{j,S^k(\mathsf{ML}_\tau)}(\mathsf{GC})\, r_j^k$, where we use $P_{j,S}(\mathsf{GC})$ to denote the probability that a customer purchases product j when we offer the subset S of products, given that customers choose under the ground choice model. Our goal is to compare the actual expected revenues $\big\{\mathsf{Rev}^k(\mathsf{MC}_\tau) : k = 1, \ldots, 100\big\}$ and $\big\{\mathsf{Rev}^k(\mathsf{ML}_\tau) : k = 1, \ldots, 100\big\}$ obtained under 100 samples of the product revenues. Similar to our results in the previous section, we repeat our computational experiments for 10 different ground choice models that we generate.

Table 2.4 shows our computational results. Each row in Table 2.4 shows the results for each one of the 10 different ground choice models that we generate. There are three blocks of five consecutive columns in Table 2.4 and each of these three blocks corresponds to one of the three values for $\tau \in \big\{1000, 1750, 2500\big\}$. The first column in each block shows the average of $\big\{\mathsf{Rev}^k(\mathsf{MC}_\tau) : k = 1, \ldots, 100\big\}$, which is the average expected revenue over all product revenue samples when we use the optimal subset that is computed under the assumption that the customers choose according to the Markov chain choice model MC_τ fitted to the past purchase history. The second column in each block shows the average of $\big\{\mathsf{Rev}^k(\mathsf{ML}_\tau) : k = 1, \ldots, 100\big\}$. The third column in each block shows the percent gap between the first and second columns. The fourth column in each block shows the number of samples of product revenues for which $\mathsf{Rev}^k(\mathsf{MC}_\tau) > \mathsf{Rev}^k(\mathsf{ML}_\tau)$. In other words, this column shows $\sum_{k=1}^{100} \mathbf{1}(\mathsf{Rev}^k(\mathsf{MC}_\tau) > \mathsf{Rev}^k(\mathsf{ML}_\tau))$, where $\mathbf{1}(\cdot)$ is the indicator function. Finally, the fifth column in each block shows the number of samples of product revenues for which $\mathsf{Rev}^k(\mathsf{MC}_\tau) < \mathsf{Rev}^k(\mathsf{ML}_\tau)$.

Grn. Cho. Mod.	$\tau = 1000$ MC_τ Exp. Rev.	ML_τ Exp. Rev.	Per. Gap.	$\mathsf{MC}_\tau \succ \mathsf{ML}_\tau$	$\mathsf{MC}_\tau \prec \mathsf{ML}_\tau$	$\tau = 1750$ MC_τ Exp. Rev.	ML_τ Exp. Rev.	Per. Gap.	$\mathsf{MC}_\tau \succ \mathsf{ML}_\tau$	$\mathsf{MC}_\tau \prec \mathsf{ML}_\tau$	$\tau = 2500$ MC_τ Exp. Rev.	ML_τ Exp. Rev.	Per. Gap.	$\mathsf{MC}_\tau \succ \mathsf{ML}_\tau$	$\mathsf{MC}_\tau \prec \mathsf{ML}_\tau$
1	74.72	67.28	9.96	62	15	74.75	67.63	9.53	58	14	74.58	67.76	9.14	59	16
2	74.80	69.15	7.55	51	14	74.89	68.95	7.92	53	11	74.35	68.33	8.09	52	13
3	74.81	66.90	10.58	56	12	75.32	67.89	9.86	56	10	75.10	68.02	9.43	55	13
4	73.78	68.46	7.21	54	24	73.98	69.67	5.82	53	24	74.69	69.58	6.84	55	26
5	75.62	66.27	12.36	74	14	75.47	66.24	12.23	73	13	75.54	66.24	12.31	73	13
6	74.02	65.39	11.67	61	12	74.22	65.66	11.54	60	17	75.07	65.66	12.54	60	12
7	72.60	64.90	10.61	58	15	73.35	64.61	11.92	65	15	73.25	64.77	11.57	59	14
8	76.16	70.09	7.96	62	19	75.73	70.13	7.39	56	20	76.79	70.01	8.84	60	14
9	75.53	71.39	5.48	50	28	76.62	71.46	6.74	49	22	76.18	71.32	6.38	50	24
10	75.01	66.57	11.25	66	13	74.78	65.93	11.83	62	12	75.27	66.44	11.74	64	9
Avg.	74.70	67.64	9.46	59	17	74.91	67.82	9.48	59	16	75.08	67.81	9.69	59	15

Table 2.4: Expected revenues from the optimal subsets that are computed under the assumption that the customers choose according to the fitted choice models.

The results in Table 2.4 indicate that the expected revenues from the subsets that are computed under the assumption that the customers choose according to the fitted Markov chain choice model are significantly larger than the expected revenues from the subsets that are computed under the assumption that the customers choose according to the fitted multinomial logit model. On average, the gap between the two expected revenues is 9.54%. Over a total of 100 samples for the product revenues, in 59% of the samples, the subsets computed by using the fitted Markov chain choice model perform better than those computed by using the fitted multinomial logit model, whereas in 16% of the samples, the subsets computed by using the fitted multinomial logit model perform better than those computed by using the fitted Markov chain choice model. Although we do not report this figure in Table 2.4, over all of the test problems, on average, the optimal solution to the 3.1 problem offers about half of the products that we have at our disposal.

2.7.5 Computing Good Policies

To compare the Markov chain choice model MC_τ and the multinomial logit model ML_τ that are fitted to the training data set H_τ, we can also check the performance of the policies that the two fitted choice models allow us to obtain for the Single Resource dynamic program. For this purpose, similar to our approach in the previous section, we generate 100 samples of the product revenues and denote these samples by $\big\{(r_1^k, \ldots, r_n^k) : k = 1, \ldots, 100\big\}$. For each sample of the product revenues, we compute the optimal policy in the Single Resource dynamic program under the assumption that the customers choose according to either one of the two fitted choice models. In particular, given that we have x units of remaining capacity

at the beginning of time period t, we let $S_t^k(\mathsf{MC}_\tau, x)$ be the optimal subset of products to offer when the product revenues are $(r_1^k, \ldots, r_n^k)$ and we compute the optimal policy under the assumption that the customers choose according to the Markov chain choice model MC_τ. The subsets $\left\{S_t^k(\mathsf{MC}_\tau, x) : x = 0, \ldots, c,\ t = 1, \ldots, T\right\}$ can be computed by solving the Single Resource dynamic program after replacing $(r_1, \ldots, r_n)$ with $(r_1^k, \ldots, r_n^k)$ and $P_{j,S}$ with $P_{j,S}(\mathsf{MC}_\tau)$. These subsets characterize a policy in the sense that if we have x units of remaining capacity at the beginning of time period t, then we offer the subset $S_t^k(\mathsf{MC}_\tau, x)$. We check the quality of this policy by computing the actual total expected revenue obtained by this policy, given that the customers actually choose according to the ground choice model. In particular, we can compute the actual total expected revenue obtained by the policy characterized by the subsets $\left\{S_t^k(\mathsf{MC}_\tau, x) : x = 0, \ldots, c,\ t = 1, \ldots, T\right\}$ by solving the dynamic program

$$V_t^k(\mathsf{MC}_\tau, x) = \sum_{j \in N} P_{j, S_t^k(\mathsf{MC}_\tau, x)}(\mathsf{GC}) \left\{r_j + V_{t+1}^k(\mathsf{MC}_\tau, x-1) - V_{t+1}^k(\mathsf{MC}_\tau, x)\right\} + V_{t+1}^k(\mathsf{MC}_\tau, x),$$

which is the analogue of the Single Resource dynamic program without the maximum operator. In this case, $V_1^k(\mathsf{MC}_\tau, c)$ is the actual total expected revenue obtained by the policy characterized by the subsets $\left\{S_t^k(\mathsf{MC}_\tau, x) : x = 0, \ldots, c,\ t = 1, \ldots, T\right\}$. Using the same approach, given that we have x units of remaining capacity at the beginning of time period t, we let $S_t^k(\mathsf{ML}_\tau, x)$ be the optimal subset of products to offer when the product revenues are $(r_1^k, \ldots, r_n^k)$ and we compute the optimal policy under the assumption that the customers choose according to the multinomial logit model ML_τ. Also, we let $V_1^k(\mathsf{ML}_\tau, c)$ be the actual total expected revenue obtained by the policy characterized by the subsets $\left\{S_t^k(\mathsf{ML}_\tau, x) : x = 0, \ldots, c,\ t = 1, \ldots, T\right\}$. Our goal is to compare the actual total expected revenues $\left\{V_1^k(\mathsf{MC}_\tau, c) : k = 1, \ldots, 100\right\}$ and $\left\{V_1^k(\mathsf{ML}_\tau, c) : k =\right.$

$1, \ldots, 100\}$ obtained under 100 samples of the product revenues. Similar to earlier results, we repeat our computational experiments for 10 different ground choice models that we generate.

Table 2.5 shows our computational results. In all of our test problems, we have $c = 36$ units of resource at the beginning of the selling horizon and $T = 50$ time periods in the selling horizon. Each row in Table 2.5 shows the results for each one of the 10 different ground choice models that we generate. There are three blocks of five consecutive columns in Table 2.5 and each of these three blocks corresponds to one of the three values for $\tau \in \{1000, 1750, 2500\}$. The first column in each block show the average of $\{V_1^k(\mathsf{MC}_\tau, c) : k = 1, \ldots, 100\}$, whereas the second column shows the average of $\{V_1^k(\mathsf{ML}_\tau, \mathsf{c}) : k = 1, \ldots, 100\}$. The third column in each block shows the percent gap between the first and second columns. The fourth column in each block shows the number of samples of product revenues for which $V_1^k(\mathsf{MC}_\tau, c) > V_1^k(\mathsf{ML}_\tau, c)$. Finally, the fifth column in each block shows the number of samples of product revenues for which $V_1^k(\mathsf{MC}_\tau, c) < V_1^k(\mathsf{ML}_\tau, c)$.

The results in Table 2.5 indicate that the total expected revenues from the policies that are computed under the assumption that the customers choose according to the fitted Markov chain choice model are significantly larger than the total expected revenues from the policies that are computed under the assumption that the customers choose according to the fitted multinomial logit model. When we have 2500 customers in the training data sets, the average gap between the total expected revenues obtained by fitting a Markov chain choice model and a multinomial logit choice model is 4.74%. Over a total of 100 samples of product revenues, in 75% of the samples, the policies obtained by using the fitted Markov chain choice model perform better than those obtained by using the fitted multi-

Grn. Cho. Mod.	$\tau = 1000$ MC_τ Exp. Rev.	ML_τ Exp. Rev.	Per. Gap.	$\mathsf{MC}_\tau \succ \mathsf{ML}_\tau$	$\mathsf{MC}_\tau \prec \mathsf{ML}_\tau$	$\tau = 1750$ MC_τ Exp. Rev.	ML_τ Exp. Rev.	Per. Gap.	$\mathsf{MC}_\tau \succ \mathsf{ML}_\tau$	$\mathsf{MC}_\tau \prec \mathsf{ML}_\tau$	$\tau = 2500$ MC_τ Exp. Rev.	ML_τ Exp. Rev.	Per. Gap.	$\mathsf{MC}_\tau \succ \mathsf{ML}_\tau$	$\mathsf{MC}_\tau \prec \mathsf{ML}_\tau$
1	2952	2856	3.24	80	17	3003	2863	4.64	78	20	3000	2864	4.55	75	24
2	3068	2963	3.43	73	24	3071	2956	3.75	73	24	3067	2935	4.30	74	21
3	3030	2887	4.71	75	19	3038	2912	4.17	77	17	3038	2914	4.10	77	18
4	3047	2916	4.30	75	20	3051	2946	3.43	72	25	3057	2945	3.66	74	26
5	2979	2795	6.16	67	15	2979	2793	6.25	68	14	2979	2794	6.20	68	14
6	3007	2835	5.71	78	16	3007	2838	5.62	75	20	3027	2836	6.31	83	13
7	2964	2782	6.15	76	21	2970	2780	6.39	77	21	2969	2786	6.15	76	22
8	3111	2983	4.11	75	22	3092	2980	3.62	74	22	3108	2982	4.06	79	19
9	3050	2961	2.92	61	38	3055	2962	3.03	67	32	3042	2955	2.86	64	35
10	3028	2868	5.28	81	19	3022	2847	5.77	83	17	3021	2863	5.22	79	21
Avg.	3023	2885	4.60	74	21	3029	2888	4.67	74	21	3031	2887	4.74	75	21

Table 2.5: Total expected revenues from the optimal policies that are computed under the assumption that the customers choose according to the fitted choice models.

nomial logit model. Overall, our results point out that fitting a Markov chain choice model to the past purchase history of the customers can provide significant improvements over fitting a multinomial logit model.

2.8 Performance Improvements for Network Revenue Management

In this section, we provide computational experiments to demonstrate the benefits from using the REDUCED linear program to obtain the optimal solution to the CHOICE BASED linear program, rather than solving the CHOICE BASED linear program directly by using column generation.

2.8.1 Experimental Setup

In our computational experiments, we generate a number of network revenue management problem instances. For each problem instance, we use two strategies to obtain the optimal solution to the CHOICE BASED linear program. The first strategy solves the CHOICE BASED linear program directly by using column generation. We refer to this strategy as CG, standing for column generation. The second strategy solves the REDUCED linear program and carries out the DIMENSION REDUCTION algorithm to recover the optimal solution to the CHOICE BASED linear program by using the optimal solution to the REDUCED linear program. We refer to this strategy as DR, standing for dimension reduction. Our objective is to compare the performances of CG and DR. We use the following approach to generate our test problems. We sample β_j from the uniform distribution over $[0,1]$ and set $\lambda_j = \beta_j / \sum_{i \in N} \beta_i$. Similarly, we sample $\zeta_{j,i}$ from the uniform distribution over $[0,1]$ and set $\rho_{j,i} = (1 - P_0)\, \zeta_{j,i} / \sum_{k \in N} \zeta_{j,k}$, where P_0 is a parameter that we vary. In this case, if a customer considers purchasing product j and product

j is not available, then she leaves without making a purchase with probability $1 - \sum_{i \in N} \rho_{j,i} = 1 - (1 - P_0) \sum_{i \in N} \zeta_{j,i} / \sum_{k \in N} \zeta_{j,k} = P_0$. Thus, the parameter P_0 controls the tendency of the customers to leave without a purchase.

In all of our problem instances, we normalize the length of the selling horizon to $T = 100$. We sample the revenue r_j of product j from the uniform distribution over $[200, 600]$. For each product j, we randomly choose a resource q_j and set $a_{q_j,j} = 1$. For the other resources, we set $a_{q,j} = 1$ with probability ξ and $a_{q,j} = 0$ with probability $1-\xi$ for all $q \in M \backslash \{q_j\}$, where ξ is a parameter that we vary. Therefore, the expected number of resources used by a product is $1 + (m-1)\,\xi$ and ξ controls the expected number of resources used by a product. To generate the capacities of the resources, we solve the problem $\max_{S \subset N} \sum_{j \in N} r_j\, P_{j,S}$ to obtain the optimal subset of products to offer under the assumption that we have unlimited resource capacities. Letting S^* be the optimal solution to this problem, we set the capacity of resource q as $c_q = \kappa \sum_{j \in N} T\, a_{q,j}\, P_{j,S^*}$, where κ is another parameter that we vary. We note that $\sum_{j \in N} T\, a_{q,j}\, P_{j,S^*}$ is the total expected capacity consumption of resource q when we offer the subset S^* of products that is computed under the assumption that we have unlimited resource capacities. Therefore, the capacity of resource q is obtained by multiplying this total expected capacity consumption by κ and the parameter κ controls the tightness of the resource capacities.

We vary (m, n, P_0, ξ, κ) over $\{25, 50\} \times \{250, 500\} \times \{0.1, 0.3\} \times \{0.02, 0.2\} \times \{0.6, 0.8\}$, where m is the number of resources, n is the number of products and the parameters P_0, ξ and κ are as defined in the previous two paragraphs. This setup yields 32 problem instances.

2.8.2 Computational Results

Table 2.6 summarizes our main computational results. The first column in this table shows the problem instances by using the tuple (m, n, P_0, ξ, κ). The second column shows the CPU seconds for CG. The third column shows the CPU seconds for DR. The fourth column shows the ratio of the CPU seconds for CG and DR, corresponding to the relative improvement in the CPU seconds obtained by recovering the optimal solution to the Choice Based linear program by using the Reduced linear program, instead of solving the Choice Based linear program directly by using column generation. The fifth column shows what portion of the CPU seconds for DR is spent on solving the Reduced linear program, whereas the sixth column shows what portion of the CPU second for DR is spent on carrying out the Dimension Reduction algorithm.

The results in Table 2.6 indicate that DR can provide remarkable performance improvements over CG. Over all of our test problems, DR improves the CPU seconds for CG by an average factor of 1836 and there is a test problem where DR can improve the CPU seconds for CG by a factor exceeding 10000. We observe a number of test problems where the CPU seconds for CG exceed 7200, corresponding to more than two hours of run time. For these test problems, DR can obtain the optimal solution to the Choice Based linear program within only 2.5 seconds. About 80% of the CPU seconds for DR is spent on solving the Reduced linear program and the rest is spent on carrying out the Dimension Reduction algorithm. In Table 2.6, we observe two trends in the performance improvements provided by DR. First, as the number of resources and the number of products increase and the size of the problem becomes larger, the performance improvements provided by DR become more pronounced. For the smallest test problems with

$m = 25$ and $n = 250$, DR improves the CPU seconds for CG by an average factor of 503, whereas for the largest test problems with $m = 50$ and $n = 500$, DR improves the CPU seconds for CG by an average factor of 4229. Second, as κ decreases and the capacities of the resources become tighter, the performance improvement provided by CG also tends to increase. For the test problems with $\kappa = 0.8$, DR improves the CPU seconds for CG by an average factor of 1173, whereas for the test problems with $\kappa = 0.6$, DR improves the CPU seconds for CG by an average factor of 2498. Overall, our computational results indicate that DR can be faster than CG by orders of magnitude in obtaining the optimal solution to the Choice Based linear program.

Column generation approaches can be fast in obtaining a near optimal solution, but they can also be slow in ultimately reaching the optimal solution, which can be a factor in the relatively poor performance of CG in Table 2.6. To ensure that this concern is not a significant factor in our computational experiments, Table 2.7 shows the CPU seconds for CG to obtain a solution that provides an objective value within 1% of the optimal objective value of the Choice Based linear program. Similar to Table 2.6, the first column in this table shows the problem instances. The second column shows the CPU seconds for CG to obtain a solution within 1% of the optimal objective value. For comparison purposes, the third column shows the CPU seconds for DR to obtain the optimal solution to the Choice Based linear program. Thus, the entries of the third column are identical to those of the third column of Table 2.6. The fourth column in Table 2.7 shows the ratios of the CPU seconds in the second and third columns. The results indicate that DR continues to provide substantial performance improvements over CG even when we terminate CG once it obtains a solution within 1% of the optimal objective value. There are problem instances, where DR improves the CPU seconds for

CG by factors exceeding 1000. Over all of our test problems, DR improves the performance of CG by an average factor of 286. Similar to the trends in Table 2.6, the results in Table 2.7 indicate that the performance improvements provided by DR become more pronounced as the size of the problem, measured by the number of resources and products, increases, or as the capacities on the flight legs become tighter.

Table 2.8 provides computational experiments for test problems with 100 resources and 2000 products. For these test problems, we vary (P_0, ξ, κ) over $\{0.1, 0.3\}$ $\times \{0.02, 0.2\} \times \{0.6, 0.8\}$. We note that these test problems are significantly larger than the earlier ones. For these test problems, two hours of run time is not enough for CG to provide a solution within 1% of the optimal objective value of the Choice Based linear program. The first column in Table 2.8 lists the test problems by using the tuple (m, n, P_0, ξ, κ). The second column shows the optimality gaps obtained by CG after two hours of run time. The third column shows the CPU seconds for DR to obtain the optimal solution to the Choice Based linear program. On average, DR obtains the optimal solution to the Choice Based linear program in less than three minutes. The average optimality gap of the solutions obtained by CG even after two hours of run time is 7.2%. There are test problems where CG terminates with more than 10% optimality gap after two hours of run time. In contrast, the largest CPU seconds for DR is about 225. Our results indicate that DR continues to provide dramatic improvements over CG for larger test problems.

2.9 Conclusions

We studied three classes of revenue management problems under the Markov chain choice model. For the assortment optimization setting, we showed that the choice probabilities under the Markov chain model can be associated with the extreme points of a polyhedron. Using this result, we showed that the assortment optimization problem can be formulated as a linear program. We derived a structural property that shows that the optimal assortment becomes smaller as we decrease the product revenues by the same positive amount. For the single resource revenue management setting, we showed that the optimal policy can be implemented by associating a protection level with each product so that it is optimal to offer a product only when the remaining resource capacity exceeds the protection level for the product. For the network revenue management setting, we considered a deterministic linear programming approximation whose number of decision variables grows exponentially with the number of products. We showed how to reduce this linear program into an equivalent one with a much smaller size. Furthermore, we gave an algorithm to recover the optimal solution to the original linear program by using the optimal solution to the reduced linear program. Our computational experiments served two purposes. First, they indicated that the Markov chain choice model may capture the customer choice behavior better when compared with the multinomial logit model, while ensuring that the corresponding optimization problems remain tractable. Second, they demonstrated that the reduced linear program is remarkably effective, improving the solution times for the original linear program by factors exceeding 10000.

One can extend the Markov chain choice model to limit the number of tran-

sitions of a customer so that the customer leaves the system without a purchase when the number of transitions reaches a certain limit. Computing the purchase probabilities of the products under this version of the Markov chain choice model is not difficult. In particular, we let $P^k_{j,S}$ be the probability that a customer visits the available product j in transition k, whereas we let $R^k_{j,S}$ be the probability that a customer visits the unavailable product j in transition k. Using m to denote the limit on the number of transitions, these probabilities satisfy a version of the BALANCE equations given by $P^1_{j,S} + R^1_{j,S} = \lambda_j$ for all $j \in N$, $P^k_{j,S} + R^k_{j,S} = \sum_{i \in N} \rho_{i,j} R^{k-1}_{i,S}$ for all $j \in N$, $k = 2, \ldots, m$, $P^k_{j,S} = 0$ for all $j \notin S$, $k = 1, \ldots, m$ and $R^k_{j,S} = 0$ for all $j \in S$, $k = 1, \ldots, m$. For example, to see the interpretation of the second one of these equations, we observe that $P^k_{j,S} + R^k_{j,S}$ on the left side corresponds to the probability that a customer visits product j in transition k. To visit product j in transition k, the customer should visit some unavailable product i in transition $k-1$ and transition from product i to product j, yielding $\sum_{i \in N} \rho_{i,j} R^{k-1}_{i,S}$ on the right side. Therefore, we can compute the purchase probabilities of the products when we have a limit on the number of transitions. However, finding the optimal subset of products to offer in the 3.1 problem is difficult. In particular, Lemma 2, which is critical to all of our results in the paper, does not hold when we have a limit on the number of transitions. To see the source of difficulty, we can verify that if we are allowed to offer different subsets to customers in different transitions, then Lemma 2 has a natural extension to the case where we have a limit on the number of transitions, but this extension assumes that there are subsets $S^1, \ldots, S^m$ such that we offer subset S^k to a customer in transition k. This assumption is not realistic since we cannot change the subset of offered products during the choice process of a customer. Therefore, if we have a limit on the number of transitions, then the main difficulty in the 3.1 problem is due to the fact that the same subset

of products should be offered to customers in different transitions. It may be possible to come up with an integer programming formulation of the 3.1 problem to ensure that the same subset of products should be offered to customers in different transitions. Such an integer programming formulation may allow solving the 3.1 problem when we have a limit on the number of transitions, but extensions to the single resource and network revenue management problems are still difficult since the use of duality theory on the linear programming formulation of the 3.1 problem plays a critical role in our results for the single resource and network revenue management problems. More work is needed in this direction.

There are other research directions for future work. To begin with, it is interesting to incorporate constraints on the offered subsets of products. When there is limited space availability to display products, one may be interested in limiting the total number of offered products. Similarly, if each product occupies a different amount of space, then one may be interested in limiting the total amount of space consumed by the offered products. Also, the Markov chain choice model is rich in parameters and it can potentially capture a variety of customer choice patterns. In this paper, we use a relatively straightforward maximum likelihood estimation method to estimate the parameters of the Markov chain choice model, which utilizes optimization software that only needs evaluating the value of the objective function. Finding effective approaches to estimate the parameters of this choice model is another important research direction to pursue.

Test Problem (m, n, P_0, ξ, κ)	CPU Secs. CG	CPU Secs. DR	CPU Secs. Ratio	Red. Lin.	Dim. Red.
(25, 250, 0.1, 0.02, 0.8)	162	0.23	722	0.18	0.05
(25, 250, 0.1, 0.02, 0.6)	200	0.35	571	0.27	0.08
(25, 250, 0.1, 0.2, 0.8)	112	0.25	447	0.20	0.05
(25, 250, 0.1, 0.2, 0.6)	170	0.22	772	0.18	0.04
(25, 250, 0.3, 0.02, 0.8)	59	0.22	270	0.17	0.05
(25, 250, 0.3, 0.02, 0.6)	137	0.19	722	0.15	0.04
(25, 250, 0.3, 0.2, 0.8)	36	0.26	139	0.22	0.04
(25, 250, 0.3, 0.2, 0.6)	104	0.27	384	0.23	0.04
(25, 500, 0.1, 0.02, 0.8)	1575	1.58	997	1.22	0.36
(25, 500, 0.1, 0.02, 0.6)	1957	1.36	1439	0.98	0.38
(25, 500, 0.1, 0.2, 0.8)	1772	2.00	886	1.63	0.37
(25, 500, 0.1, 0.2, 0.6)	2487	1.63	1526	1.24	0.39
(25, 500, 0.3, 0.02, 0.8)	556	1.61	345	1.31	0.30
(25, 500, 0.3, 0.02, 0.6)	1174	1.52	772	1.18	0.34
(25, 500, 0.3, 0.2, 0.8)	559	1.99	281	1.67	0.32
(25, 500, 0.3, 0.2, 0.6)	1383	2.01	688	1.66	0.35
(50, 250, 0.1, 0.02, 0.8)	410	0.22	1831	0.18	0.05
(50, 250, 0.1, 0.02, 0.6)	926	0.22	4208	0.16	0.06
(50, 250, 0.1, 0.2, 0.8)	398	0.29	1374	0.24	0.05
(50, 250, 0.1, 0.2, 0.6)	468	0.25	1901	0.20	0.05
(50, 250, 0.3, 0.02, 0.8)	213	0.26	820	0.22	0.04
(50, 250, 0.3, 0.02, 0.6)	387	0.22	1758	0.18	0.04
(50, 250, 0.3, 0.2, 0.8)	155	0.28	552	0.24	0.04
(50, 250, 0.3, 0.2, 0.6)	389	0.26	1495	0.21	0.05
(50, 500, 0.1, 0.02, 0.8)	6628	1.75	3788	1.40	0.35
(50, 500, 0.1, 0.02, 0.6)	15500	1.50	10333	1.11	0.39
(50, 500, 0.1, 0.2, 0.8)	7739	2.12	3651	1.74	0.38
(50, 500, 0.1, 0.2, 0.6)	10659	1.90	5610	1.50	0.40
(50, 500, 0.3, 0.02, 0.8)	3008	1.84	1635	1.54	0.30
(50, 500, 0.3, 0.02, 0.6)	6780	1.64	4134	1.29	0.35
(50, 500, 0.3, 0.2, 0.8)	2392	2.32	1031	1.99	0.33
(50, 500, 0.3, 0.2, 0.6)	7600	2.08	3654	1.71	0.37
Avg.	2378	1.03	1836	0.87	0.16

Table 2.6: CPU seconds for CG and DR.

Test Problem (m, n, P_0, ξ, κ)	CPU Secs. for CG 1%	CPU Secs. for DR Opt.	CPU Secs. Ratio
$(25, 250, 0.1, 0.02, 0.8)$	25	0.23	110
$(25, 250, 0.1, 0.02, 0.6)$	48	0.35	137
$(25, 250, 0.1, 0.2, 0.8)$	17	0.25	68
$(25, 250, 0.1, 0.2, 0.6)$	34	0.22	153
$(25, 250, 0.3, 0.02, 0.8)$	12	0.22	53
$(25, 250, 0.3, 0.02, 0.6)$	32	0.19	171
$(25, 250, 0.3, 0.2, 0.8)$	7	0.26	26
$(25, 250, 0.3, 0.2, 0.6)$	16	0.27	59
$(25, 500, 0.1, 0.02, 0.8)$	228	1.58	144
$(25, 500, 0.1, 0.02, 0.6)$	447	1.36	329
$(25, 500, 0.1, 0.2, 0.8)$	123	2.00	62
$(25, 500, 0.1, 0.2, 0.6)$	334	1.63	205
$(25, 500, 0.3, 0.02, 0.8)$	86	1.61	53
$(25, 500, 0.3, 0.02, 0.6)$	228	1.52	150
$(25, 500, 0.3, 0.2, 0.8)$	63	1.99	32
$(25, 500, 0.3, 0.2, 0.6)$	146	2.01	73
Avg.	115	0.98	114

Test Problem (m, n, P_0, ξ, κ)	CPU Secs. for CG 1%	CPU Secs. for DR Opt.	CPU Secs. Ratio
$(50, 250, 0.1, 0.02, 0.8)$	60	0.22	269
$(50, 250, 0.1, 0.02, 0.6)$	231	0.22	1050
$(50, 250, 0.1, 0.2, 0.8)$	64	0.29	220
$(50, 250, 0.1, 0.2, 0.6)$	96	0.25	389
$(50, 250, 0.3, 0.02, 0.8)$	40	0.26	154
$(50, 250, 0.3, 0.02, 0.6)$	91	0.22	413
$(50, 250, 0.3, 0.2, 0.8)$	18	0.28	65
$(50, 250, 0.3, 0.2, 0.6)$	70	0.26	268
$(50, 500, 0.1, 0.02, 0.8)$	805	1.75	460
$(50, 500, 0.1, 0.02, 0.6)$	2,601	1.50	1734
$(50, 500, 0.1, 0.2, 0.8)$	715	2.12	337
$(50, 500, 0.1, 0.2, 0.6)$	1,326	1.90	698
$(50, 500, 0.3, 0.02, 0.8)$	390	1.84	212
$(50, 500, 0.3, 0.02, 0.6)$	1,065	1.64	649
$(50, 500, 0.3, 0.2, 0.8)$	176	2.32	76
$(50, 500, 0.3, 0.2, 0.6)$	705	2.08	339
Avg.	528	1.07	458

Table 2.7: CPU seconds for CG and DR when we terminate CG once it obtains a solution within 1% of the optimal objective value.

Test Problem $(m, \quad n, \ P_0, \quad \xi, \quad \kappa)$	CG % Opt. Gap	DR CPU Secs.
$(100, 2000, \ 0.1, 0.02, 0.8)$	5.87	158.58
$(100, 2000, \ 0.1, 0.02, 0.6)$	12.53	144.50
$(100, 2000, \ 0.1, \ \ 0.2, 0.8)$	5.38	194.69
$(100, 2000, \ 0.1, \ \ 0.2, 0.6)$	10.27	165.95
$(100, 2000, \ 0.3, 0.02, 0.8)$	4.89	194.75
$(100, 2000, \ 0.3, 0.02, 0.6)$	9.69	224.79
$(100, 2000, \ 0.3, \ \ 0.2, 0.8)$	2.83	173.70
$(100, 2000, \ 0.3, \ \ 0.2, 0.6)$	6.13	178.08
Avg.	7.20	179.38

Table 2.8: Optimality gaps for CG when we terminate it after two hours of run time and CPU seconds for DR to obtain the optimal solution.

Chapter 3

Capacity Constraints across Nests in Assortment Optimization under the Nested Logit Model

3.1 Introduction

In this chapter, we consider assortment optimization problems when customers choose according to the NL model and there is limited capacity for the products in the offered assortment. We consider a setting where we need to decide which assortment of products to offer. Each arriving customer chooses among the offered products according to the NL model. Under the NL model, the products are organized in nests. Each customer, after viewing the offered assortment, decides either to make a purchase within one of the nests or to leave the system without

purchasing anything. If a nest is chosen, then the customer purchases one of the products within the chosen nest. There is a capacity constraint limiting the total capacity consumption of the products in the offered assortment. The goal is to choose an assortment of products to offer so as to maximize the expected revenue obtained from each customer. We consider two types of capacity constraints. In the first type of constraints, each product occupies one unit of space, in which case, the capacity constraint limits the total number of products in the offered assortment. We refer to this type of a capacity constraint as a *cardinality constraint*. In the second type of constraints, the capacity consumption of a product is arbitrary, possibly reflecting the space or capital requirement of a product. We refer to this type of a capacity constraint as a *space constraint*.

Under a cardinality constraint, we show that we can obtain an optimal assortment by solving a linear program with $O(m^2n)$ decision variables and $O(m^2n^4)$ constraints, where m in the number of nests and n is the number of products in each nest. As far as we are aware, the assortment problem was not known to be tractable when customers choose according to the NL model and there is a cardinality constraint limiting the total number of products in the offered assortment. This chapter gives the first exact solution method for this problem. On the other hand, under a space constraint, we show that we can obtain a 4-approximate solution by solving a linear program with $O(m)$ decision variables and $O(mn^4)$ constraints. The running time of this algorithm scales polynomially with the number of products and the number of nests. To our knowledge, this chapter gives the first algorithm for the assortment problem that scales polynomially with the number of nests, when there is a capacity constraint on the space consumption of all offered products and customers choose according to the NL model. In addition to giving algorithms to solve the assortment problem, we give a tractable linear program

that computes an upper bound on the optimal expected revenue. By comparing the expected revenues of the assortments obtained by our 4-approximation algorithm with the upper bounds on the optimal expected revenues, we demonstrate that our 4-approximation algorithm performs quite well in practice.

An attractive approach for modeling the customer choice process is to use the utility maximization principle, where a customer associates a random utility with each product and chooses the product with the largest utility. MNL model is one of the popular choice models that are based on the utility maximization principle where the utilities of the products are independent of each other; see [37] and [25]. Due to the independence of the utilities of the products, the MNL model implicitly assumes that how highly a customer evaluates a certain product has nothing to do with how highly the same customer evaluates another product. The NL model remedies this shortcoming by organizing the products in nests such that the utilities of the products in the same nest can be dependent on each other; see [42]. This feature allows the modeler to capture situations where the products in the same nest are alike and how highly a customer evaluates a certain product can be a good indicator of how highly the same customer evaluates another product.

[41] and [8] consider revenue management problems under customer choice behavior. During the course of their analyses, they give efficient approaches to solve the assortment problem under the MNL model without any constraints. [34], [46] and [47] consider assortment problems under variants of the MNL model with a cardinality constraint on the offered assortment and show that the problem can be solved efficiently. [2], [30] and [35] consider assortment problems where there are multiple customer types and customers of different types choose according to MNL models with different parameters. The authors show that the problem is

NP-complete, study heuristics and investigate valid cuts for integer programming formulations.

[4] show how to solve the assortment problem under the NL model without any constraints. [21] give a greedy algorithm for the same problem. [9] study constrained assortment problems under the NL model, but they impose capacity constraints separately on the assortment offered in each nest. Similar to us, [33] and [5] consider assortment problems under the NL model, where there is a constraint on the total capacity consumption of the products offered in all nests. They give approximation schemes that tradeoff running time with solution quality, but the running time for their approaches grows exponentially with the number of nests. For example, to obtain a 4-approximate solution, [33] need $O(m(m^6n^6\log(mn))^m)$ operations, which gets prohibitive when m exceeds two or three but there are practical applications where the number of nests easily exceeds two or three; see [43], [39] and [12].

As mentioned above, [9] study assortment problems under the NL model, but they impose capacity constraints separately on the assortment offered in each nest. To understand their constraint structure, consider a retailer that is interested in finding an assortment of cars to offer to its customers. We model the demand for cars through a NL model, where each car category, such as compact, mid-size and sedan, corresponds to a different nest. The products within a nest correspond to cars of different models within the car category corresponding to the nest. The constraint structure in [9] assumes that there is a separate fixed amount of space reserved for the cars offered in each car category and the retailer is interested in finding a set of cars to offer such that the set of cars offered in each car category does not violate the space reserved for that car category. In contrast, we impose a

capacity constraint on the assortment offered over all nests. Thus, our constraint structure assumes that there is a fixed amount of space available for all car categories and the retailer is interested in finding a set of cars to offer such that the total amount of space consumed by all offered cars in all car categories does not violate the space availability.

One line of attack for assortment optimization under the NL model has been to identify a collection of good candidate subsets to offer in each nest. Once these collections are identified, it is possible to solve a separate linear program to pick a subset to offer in each nest so that the combined subsets over all nests provide a good assortment. This is the strategy followed by [4] and [9]. We follow an approach similar to theirs in identifying the collections of candidate subsets in each nest, but due to the fact that our capacity constraint limits the total capacity consumption of the subsets of products offered in all nests, different nests interact with each other, making the assortment problem substantially more difficult. The linear programs used by [4] and [9] become ineffective and we cannot build on the earlier work to figure out how to pick a subset to offer in each nest so that the combined subsets over all nests provide the highest possible expected revenue.

We get around this difficulty by using the following general methodology. The expected revenue function under the NL model is a fraction. We convert the problem of finding an assortment that maximizes the expected revenue into the problem of finding the fixed point of a function. Computing the value of this function at any point requires solving an optimization problem, which involves finding a subset of products to offer in each nest so as to maximize a nonlinear function. Under a cardinality constraint, we show that we can consider a small number of candidate subsets in each nest without incurring any loss. Furthermore,

we can maximize the nonlinear function by using dynamic programming. Under a space constraint, we show that we can consider a small number of candidate subsets in each nest while incurring a constant factor loss. Furthermore, we can maximize the nonlinear function with a constant factor loss by solving the linear programming relaxation of a multiple choice knapsack problem.

The discussion in the previous paragraph indicates that our approach is related to maximizing a fraction. [27] shows how to build on a tractable algorithm for a combinatorial optimization problem with a linear objective function to develop a tractable algorithm for the same combinatorial optimization problem with a fractional objective function, where the numerator and the denominator of the fraction are linear. [13] extend this result by showing how to build on an approximation algorithm for the former problem to give an approximation algorithm for the latter, but this extension requires the existence of an approximation algorithm when the linear objective function has negative coefficients. [3] show how to drop this requirement. [31] make generalizations to sums of fractions. One of the important distinguishing features of our assortment problem is that the objective function is a fraction where the numerator and the denominator involve nonlinearities. Thus, the general methods in the papers outlined in this paragraph do not immediately apply.

3.2 Problem Formulation

In this section, we formulate the assortment optimization problem that we want to solve. There are m nests indexed by $M = \{1, \ldots, m\}$. In each nest, there are n products that we can offer to customers and we index the products by

$N = \{1, \ldots, n\}$. Although we assume that each nest has the same number of products, this assumption is only for notational brevity and it is straightforward to extend our results to the case where different nests have different numbers of products. Under the NL model, a customer decides either to make a purchase within one of the nests or to leave without purchasing anything. If the customer decides to make a purchase within one of the nests, then the customer chooses one of the products offered in this nest. We let v_{ij} be the preference weight associated with product j in nest i. Given that we offer the assortment $S_i \subset N$ of products in nest i, we use $V_i(S_i) = \sum_{j \in S_i} v_{ij}$ to denote the total preference weight of the products in the offered assortment. Under the NL model, if we offer the assortment S_i in nest i and a customer has already decided to make a purchase in this nest, then this customer chooses product $j \in S_i$ in nest i with probability $v_{ij}/V_i(S_i)$. We let r_{ij} be the revenue associated with product j in nest i. In this case, given that we offer the assortment S_i in nest i and a customer has already decided to make a purchase in this nest, the expected revenue that we obtain from this customer can be written as

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

Associated with each nest, there is a parameter $\gamma_i \in [0, 1]$ capturing the degree of dissimilarity between the products in nest i. The preference weight of the no purchase option is v_0. Under the NL model, if we offer the assortment $(S_1, \ldots, S_m)$ over all nests with $S_i \subset N$ for all $i \in M$, then a customer chooses nest i with probability $Q_i(S_1, \ldots, S_m) = V_i(S_i)^{\gamma_i} / (v_0 + \sum_{l \in M} V_l(S_l)^{\gamma_l})$, which corresponds to the probability that a customer is attracted to nest i as a function of the assortment $(S_1, \ldots, S_m)$ offered over all nests. So, if we offer the assortment $(S_1, \ldots, S_m)$ over

all nests, then we obtain an expected revenue of

$$\Pi(S_1, \ldots, S_m) = \sum_{i \in M} Q_i(S_1, \ldots, S_m)\, R_i(S_i) = \frac{\sum_{i \in M} V_i(S_i)^{\gamma_i} R_i(S_i)}{v_0 + \sum_{i \in M} V_i(S_i)^{\gamma_i}}$$

from each customer. Our goal is to find an assortment of products so as to maximize the expected revenue from each customer, subject to a capacity constraint on the offered assortment.

We consider two types of capacity constraints. In the first type of constraint, we limit the total number of products offered over all nests to c. Thus, the set of feasible assortments can be written as $\{(S_1, \ldots, S_m) : \sum_{i \in M} |S_i| \leq c,\ S_i \subset N\ \forall\, i \in M\}$. We refer to this constraint as a cardinality constraint. In the second type of constraint, we let w_{ij} be the space requirement of product j in nest i and limit the total space requirement of the products offered over all nests to c. In this case, the set of feasible assortments is $\{(S_1, \ldots, S_m) : \sum_{i \in M} \sum_{j \in S_i} w_{ij} \leq c,\ S_i \subset N\ \forall\, i \in M\}$. We refer to this constraint as a space constraint. For uniformity, we use $C_i(S_i)$ to denote the capacity consumption of the assortment S_i offered in nest i. We have $C_i(S_i) = |S_i|$ under a cardinality constraint and $C_i(S_i) = \sum_{j \in S_i} w_{ij}$ under a space constraint. In this case, we can write the set of feasible assortments as $\{(S_1, \ldots, S_m) : \sum_{i \in M} C_i(S_i) \leq c,\ S_i \subset N\ \forall\, i \in M\}$ under capacity or space constraints. We want to find an assortment that maximizes the expected revenue from each customer subject to a capacity constraint, yielding the problem

$$z^* = \max_{\substack{(S_1, \ldots, S_m):\\ \sum_{i \in M} C_i(S_i) \leq c,\\ S_i \subset N\ \forall\, i \in M}} \Big\{\Pi(S_1, \ldots, S_m)\Big\}, \tag{3.1}$$

where $C_i(S_i)$ may correspond to a cardinality or space constraint. If $C_i(S_i)$ corresponds to a cardinality constraint, then we can assume without loss of generality that c is an integer. In this chapter, we show that if we have a cardinality constraint

on the offered assortment, then we can obtain an optimal solution to problem (3.1) by solving a tractable linear program. On the other hand, if we have a space constraint, then Lemma 2.1 in [33] shows that problem (3.1) is NP-hard even when there is a single nest with a dissimilarity parameter of one. Therefore, obtaining an optimal solution to problem (3.1) under a space constraint is likely to be intractable. In this chapter, we show that if we have a space constraint, then we can obtain a 4-approximate solution to problem (3.1) by solving a tractable linear program.

3.3 Fixed Point Representation

In this section, we describe the connection of problem (3.1) to the problem of computing the fixed point of a function. This connection plays an important role throughout the chapter and it becomes critical for constructing an efficient solution approach for problem (3.1) under a cardinality or space constraint. To connect problem (3.1) to the problem of computing the fixed point of a function, we define the function $f(\cdot) : \Re_+ \rightarrow \Re_+$ as

$$f(z) = \max_{\substack{(S_1,\ldots,S_m):\\ \sum_{i\in M} C_i(S_i) \leq c,\\ S_i \subset N \ \forall i \in M}} \left\{ \sum_{i\in M} V_i(S_i)^{\gamma_i} (R_i(S_i) - z) \right\}. \tag{3.2}$$

Offering the empty assortment over all nests is a feasible solution to the problem above providing the objective value of zero, so that $f(z) \geq 0$ for all $z \in \Re_+$. Consider a value of $\hat{z}$ that satisfies $f(\hat{z}) = v_0\,\hat{z}$, corresponding to the fixed point of the function $f(\cdot)/v_0$. Such a value of $\hat{z} \geq 0$ always exists since $f(\cdot)$ is a decreasing function and $f(0) \geq 0$. The next theorem shows that the value of $\hat{z}$ that satisfies

$f(\hat{z}) = v_0\,\hat{z}$ is useful in identifying an optimal solution to problem (3.1).

Theorem 12. *Let $\hat{z}$ be such that $f(\hat{z}) = v_0\,\hat{z}$. If the assortment $(\hat{S}_1, \ldots, \hat{S}_m)$ satisfies*

$$\sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i}(R_i(\hat{S}_i) - \hat{z}) \geq f(\hat{z}),$$

then we have $\Pi(\hat{S}_1, \ldots, \hat{S}_m) \geq z^*$, where z^* is the optimal objective value of problem (3.1).

Proof. We claim that $z^* = \hat{z}$. First, we show that $z^* \geq \hat{z}$. We let $(\tilde{S}_1, \ldots, \tilde{S}_m)$ be an optimal solution to problem (3.2) when we solve this problem with $z = \hat{z}$. Thus, we have $v_0\,\hat{z} = f(\hat{z}) = \sum_{i \in M} V_i(\tilde{S}_i)^{\gamma_i}(R_i(\tilde{S}_i) - \hat{z})$. Focusing on the first and last expressions in this chain of equalities and solving for $\hat{z}$ yields $\hat{z} = \sum_{i \in M} V_i(\tilde{S}_i)^{\gamma_i} R_i(\tilde{S}_i)/(v_0 + \sum_{i \in M} V_i(\tilde{S}_i)^{\gamma_i}) = \Pi(\tilde{S}_1, \ldots, \tilde{S}_m)$. Noting that $(\tilde{S}_1, \ldots, \tilde{S}_m)$ is a feasible solution to problem (3.1), we have $\Pi(\tilde{S}_1, \ldots, \tilde{S}_m) \leq z^*$. Using this inequality with the last chain of equalities, we obtain $z^* \geq \hat{z}$. Second, we show that $z^* \leq \hat{z}$. Using $(S_1^*, \ldots, S_m^*)$ to denote an optimal solution to problem (3.1), we have $z^* = \Pi(S_1^*, \ldots, S_m^*) = \sum_{i \in M} V_i(S_i^*)^{\gamma_i} R_i(S_i^*)/(v_0 + \sum_{i \in M} V_i(S_i^*)^{\gamma_i})$. Focusing on the first and last expressions in this chain of equalities and solving for z^*, we obtain $v_0\,z^* = \sum_{i \in M} V_i(S_i^*)^{\gamma_i}(R_i(S_i^*) - z^*)$. In this case, we obtain $v_0\,\hat{z} = f(\hat{z}) \geq \sum_{i \in M} V_i(S_i^*)^{\gamma_i}(R_i(S_i^*) - \hat{z}) \geq \sum_{i \in M} V_i(S_i^*)^{\gamma_i}(R_i(S_i^*) - z^*) = v_0\,z^*$, where the first inequality follows by the fact that $(S_1^*, \ldots, S_m^*)$ is a feasible solution to problem (3.2) when we solve this problem with $z = \hat{z}$ and the second inequality uses the fact that $z^* \geq \hat{z}$, which is shown above. The last chain of inequalities indicate that $z^* \leq \hat{z}$, establishing the claim. Since $f(\hat{z}) = v_0\,\hat{z}$, we write the inequality in the theorem as $\sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i}(R_i(\hat{S}_i) - \hat{z}) \geq v_0\,\hat{z}$. Solving for $\hat{z}$ in this inequality yields $\hat{z} \leq \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i} R_i(\hat{S}_i)/(v_0 + \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i}) = \Pi(\hat{S}_1, \ldots, \hat{S}_m)$, in which case, the desired result follows by noting that $\hat{z} = z^*$. □

Theorem 12 suggests the following procedure to obtain an optimal solution to problem (3.1). We find $\hat{z}$ such that $f(\hat{z}) = v_0\,\hat{z}$ and solve problem (3.2) with $z = \hat{z}$ to obtain an optimal solution $(\hat{S}_1, \ldots, \hat{S}_m)$. In this case, it is possible to show that $(\hat{S}_1, \ldots, \hat{S}_m)$ is an optimal solution to problem (3.1). To see this result, we have $f(\hat{z}) = \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i}(R_i(\hat{S}_i) - \hat{z})$ by the definition of $(\hat{S}_1, \ldots, \hat{S}_m)$, in which case, $(\hat{S}_1, \ldots, \hat{S}_m)$ satisfies the inequality in Theorem 12 and we obtain $\Pi(\hat{S}_1, \ldots, \hat{S}_m) \geq z^*$. Since the solution $(\hat{S}_1, \ldots, \hat{S}_m)$ is feasible to problem (3.2), it is feasible to problem (3.1) as well and we have $\Pi(\hat{S}_1, \ldots, \hat{S}_m) \leq z^*$. Therefore, we have $\Pi(\hat{S}_1, \ldots, \hat{S}_m) = z^*$, establishing that $\Pi(\hat{S}_1, \ldots, \hat{S}_m)$ is an optimal solution to problem (3.1), as desired. Later in the chapter, we show that we can efficiently find $\hat{z}$ that satisfies $f(\hat{z}) = v_0\,\hat{z}$ when we have a cardinality constraint. However, finding such $\hat{z}$ may be difficult when we have a space constraint. The next corollary gives an approximate version of Theorem 12 that does not require finding $\hat{z}$ such that $f(\hat{z}) = v_0\,\hat{z}$. To state this corollary, we let $f^R(\cdot)$ be an approximation to $f(\cdot)$ that satisfies $\alpha\, f^R(z) \geq f(z)$ for all $z \in \Re_+$ for some $\alpha \geq 1$. We do not yet specify how to construct this approximation. We only assume that $f^R(\cdot)$ is a decreasing function similar to $f(\cdot)$ and $f^R(0) \geq 0$, in which case, we can always find $\hat{z} \geq 0$ satisfying $f^R(\hat{z}) = v_0\,\hat{z}$. The next corollary shows that we can use this value of $\hat{z}$ to get an approximation guarantee for problem (3.1). Its proof is similar to that of Theorem 12 and deferred to Online Appendix B.1.

Corollary 13. *Let $f^R(\cdot)$ be an approximation to $f(\cdot)$ that satisfies $\alpha\, f^R(z) \geq f(z)$ for all $z \in \Re_+$ for some $\alpha \geq 1$ and $\hat{z}$ be such that $f^R(\hat{z}) = v_0\,\hat{z}$. If the assortment $(\hat{S}_1, \ldots, \hat{S}_m)$ satisfies*

$$\beta \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i}(R_i(\hat{S}_i) - \hat{z}) \geq f^R(\hat{z})$$

for some $\beta \geq 1$, then we have $\alpha\,\beta\,\Pi(\hat{S}_1, \ldots, \hat{S}_m) \geq z^*$, where z^* is the optimal objective value of problem (3.1).

3.4 Cardinality Constraint

In this section, we consider problem (3.1) under a cardinality constraint. Thus, we have $C_i(S_i) = |S_i|$ throughout this section. First, we show how to solve problem (3.2), which allows us to compute $f(z)$ at a particular value of z. Second, we show how to find a value of $\hat{z}$ that satisfies $f(\hat{z}) = v_0\,\hat{z}$. In this case, noting the discussion that follows Theorem 12, we can find an optimal solution to problem (3.1) by finding a value of $\hat{z}$ that satisfies $f(\hat{z}) = v_0\,\hat{z}$ and solving problem (3.2) with $z = \hat{z}$.

3.4.1 Computation at a Particular Point

We consider solving problem (3.2), which allows us to compute $f(z)$ at a particular value of z. The starting point for our discussion is a result due to [9], who study the problem of maximizing the expected revenue obtained from each customer subject to a separate cardinality constraint on the assortment offered in each nest. In particular, for any $z \in \Re_+$ and $b_i \in \mathbb{Z}_+$, [9] focus on the problem

$$\max_{\substack{S_i\,:\,C_i(S_i)\le b_i,\\ S_i\subset N}} \Big\{V_i(S_i)^{\gamma_i}\,(R_i(S_i)-z)\Big\} \quad = \quad \max_{\substack{S_i\,:\,|S_i|\le b_i,\\ S_i\subset N}} \Big\{V_i(S_i)^{\gamma_i}\,(R_i(S_i)-z)\Big\}. \tag{3.3}$$

The authors construct $O(n^2)$ different orderings of the products such that for any $z \in \Re_+$ and $b_i \in \mathbb{Z}_+$, an optimal solution to problem (3.3) can be obtained by sorting the products according to one of these orderings and using an assortment that includes some number of earliest products in this ordering. In other words, letting $\{\sigma^g : g \in \mathcal{G}_i\}$ with $|\mathcal{G}_i| = O(n^2)$ be the orderings constructed by [9] and

$S_i(\sigma^g, k)$ be the assortment that includes the first k products when the products in nest i are sorted according to the ordering σ^g, for any $z \in \Re_+$ and $b_i \in \mathbb{Z}_+$, an optimal solution to problem (3.3) can always be found in the collection of assortments $\{S_i(\sigma^g, k) : g \in \mathcal{G}_i,\ k = 0, \ldots, n\}$. Since $|\mathcal{G}_i| = O(n^2)$, there are $O(n^3)$ assortments in this collection. For notational brevity, we use $\mathcal{A}_i = \{S_{it} : t \in \mathcal{T}_i\}$ with $|\mathcal{T}_i| = O(n^3)$ to denote the collection of assortments $\{S_i(\sigma^g, k) : g \in \mathcal{G}_i,\ k = 0, \ldots, n\}$ and the next lemma follows.

Lemma 14. *There exists a collection of assortments $\mathcal{A}_i = \{S_{it} : t \in \mathcal{T}_i\}$ with $|\mathcal{T}_i| = O(n^3)$ such that for any $z \in \Re_+$ and $b_i \in \mathbb{Z}_+$, an optimal solution to problem (3.3) can be found in $\mathcal{A}_i$.*

Lemma 14 allows us to focus only on the assortments in the collections $\mathcal{A}_1, \ldots, \mathcal{A}_m$ in problem (3.2). In particular, we can write problem (3.2) equivalently as

$$f(z) = \max_{\substack{(S_1, \ldots, S_m) : \\ \sum_{i \in M} C_i(S_i) \leq c, \\ S_i \in \mathcal{A}_i \ \forall i \in M}} \left\{ \sum_{i \in M} V_i(S_i)^{\gamma_i} (R_i(S_i) - z) \right\}. \tag{3.4}$$

To see that problems (3.2) and (3.4) have the same optimal objective values, we note that problem (3.2) allows using assortments of the form $(S_1, \ldots, S_m)$ with $S_i \subset N$ for all i, whereas problem (3.4) allows using assortments of the form $(S_1, \ldots, S_m)$ with $S_i \in \mathcal{A}_i$ for all $i \in M$. Therefore, the optimal objective value of problem (3.2) is at least as large as the optimal objective value of problem (3.4). On the other hand, letting $(\hat{S}_1, \ldots, \hat{S}_m)$ be an optimal solution to problem (3.2) and $C_i(\hat{S}_i) = \hat{b}_i$, since $(\hat{S}_1, \ldots, \hat{S}_m)$ is a feasible solution to problem (3.2), we have $\sum_{i \in M} \hat{b}_i = \sum_{i \in M} C_i(\hat{S}_i) \leq c$. By Lemma 14, the collection of assortments $\mathcal{A}_i$ includes an optimal solution to problem (3.3) for any $z \in \Re_+$ and $b_i \in \mathbb{Z}_+$. Using this result with $b_i = \hat{b}_i$ and noting that $\hat{S}_i$ is a feasible solution to problem (3.3)

when we solve this problem with $b_i = \hat{b}_i$, it follows that there exists $\tilde{S}_i \in \mathcal{A}_i$ such that $V_i(\tilde{S}_i)^{\gamma_i}(R_i(\tilde{S}_i) - z) \geq V_i(\hat{S}_i)^{\gamma_i}(R_i(\hat{S}_i) - z)$ and $C_i(\tilde{S}_i) \leq \hat{b}_i$. Adding the last two inequalities over all $i \in M$, we have $\sum_{i \in M} V_i(\tilde{S}_i)^{\gamma_i}(R_i(\tilde{S}_i) - z) \geq \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i}(R_i(\hat{S}_i) - z)$ and $\sum_{i \in M} C_i(\tilde{S}_i) \leq \sum_{i \in M} \hat{b}_i \leq c$. Since $\tilde{S}_i \in \mathcal{A}_i$ for all $i \in M$, the last two chains of inequalities show that $(\tilde{S}_1, \ldots, \tilde{S}_m)$ is a feasible solution to problem (3.4) and the objective value provided by this solution for problem (3.4) is at least as large as the optimal objective value of problem (3.2). Therefore, the optimal objective value of problem (3.4) is at least as large as the optimal objective value of problem (3.2), establishing that problems (3.2) and (3.4) have the same optimal objective values.

The discussion above indicates that we can compute $f(z)$ at a particular value of z by solving problem (3.4), instead of problem (3.2). However, solving problem (3.4) in a brute force fashion is still difficult since there are $|\mathcal{A}_1| \times \ldots \times |\mathcal{A}_m|$ different combinations of assortments that we can choose from different nests and the number of such possible combinations grows exponentially fast with the number of nests. To solve problem (3.4) in a tractable fashion, the critical observation is that the objective function of this problem is separable by the nests. If we offer the assortment S_i in nest i, then we obtain a contribution of $V_i(S_i)^{\gamma_i}(R_i(S_i) - z)$. Problem (3.4) finds one assortment to offer in each nest to maximize the total contribution subject to the constraint that the total cardinality of the assortments offered over all nests does not exceed c. Therefore, we can solve problem (3.4) by using a dynamic program. In this dynamic program, the decision epochs correspond to the nests. The state variable in each decision epoch is the remaining capacity left from the earlier nests just before choosing the assortment offered in a particular nest. Finally, the action variable in each decision epoch is the assortment offered in a particular nest. Thus, we can compute $f(z)$ at a particular value of z by

solving the dynamic program

$$J_i(b \,|\, z) = \max_{\substack{S_i \,:\, C_i(S_i) \le b \\ S_i \in \mathcal{A}_i}} \left\{ V_i(S_i)^{\gamma_i} (R_i(S_i) - z) + J_{i+1}(b - C_i(S_i) \,|\, z) \right\}, \qquad (3.5)$$

with the boundary condition $J_{m+1}(\cdot \,|\, z) = 0$. Under a cardinality constraint, we can assume that c is an integer that does not exceed mn, which is the total number of products in all of the nests. Thus, the state space in the dynamic program above is $0, \ldots, mn$. Computing the value functions $\{J_i(b \,|\, z) : b = 0, \ldots, mn, \ i \in M\}$, the value of $J_1(c \,|z)$ corresponds to $f(z)$.

The dynamic program in (3.5) provides an efficient approach for computing $f(z)$ at a particular value of z. Since there are m decision epochs, the state space is $0, \ldots, mn$ and $|\mathcal{A}_i| = O(n^3)$, this dynamic program can be solved in $O(m^2\, n^4)$ operations. In the next section, we build on the dynamic program to find $\hat{z}$ that satisfies $f(\hat{z}) = v_0\, \hat{z}$.

3.4.2 Finding the Fixed Point

We consider the problem of finding $\hat{z}$ that satisfies $f(\hat{z}) = v_0\, \hat{z}$. For this purpose we use the linear programming representation of the dynamic program in (3.5). A dynamic program with finite states and actions has a linear programming representation. In this linear program, there is one decision variable for each state and decision epoch corresponding to the value function at each state and decision

epoch. Inspired by this linear program, we propose solving

$$\begin{aligned}
\min \quad & \Theta_1(c) && (3.6)\\
\text{st} \quad & \Theta_i(b) \geq V_i(S_i)^{\gamma_i}(R_i(S_i) - z) + \Theta_{i+1}(b - C_i(S_i)) \;\; \forall\, i \in M,\; b = 0, \ldots, mn, \\
& S_i \in \mathcal{F}_i(b) && (3.7)\\
& \Theta_1(c) = v_0\, z, && (3.8)
\end{aligned}$$

to find $\hat{z}$ satisfying $f(\hat{z}) = v_0\,\hat{z}$. The decision variables are $\Theta = \{\Theta_i(b) : i \in M,\; b = 0, \ldots, mn\}$ and z in the linear program above. We use the convention that $\Theta_{m+1}(b) = 0$ for all $b = 0, \ldots, mn$. The set $\mathcal{F}_i(b)$ is given by $\mathcal{F}_i(b) = \{S_i : C_i(S_i) \leq b,\; S_i \in \mathcal{A}_i\}$, capturing the set of feasible actions at decision epoch i and state b. If we drop the second constraint in problem (3.6)-(3.8) and minimize the objective function subject to the first set of constraints for a fixed value of z, then it is well known that the optimal value of the decision variable $\Theta_1(c)$ gives the value function $J_1(b \,|\, z)$ computed through the dynamic program in (3.5); see [32]. Interestingly, if we solve problem (3.6)-(3.8) as formulated, then the optimal value of the decision variable z gives the value of $\hat{z}$ satisfying $f(\hat{z}) = v_0\,\hat{z}$. The next theorem shows this result.

Theorem 15. *Letting $(\hat{\Theta}, \hat{z})$ be an optimal solution to problem* (3.6)-(3.8)*, $\hat{z}$ satisfies $f(\hat{z}) = v_0\,\hat{z}$.*

Proof. We let $(\hat{S}_1, \ldots, \hat{S}_m)$ be an optimal solution to problem (3.2) when we solve this problem with $z = \hat{z}$. We define $\{\hat{b}_i : i \in M\}$ as $\hat{b}_1 = c$ and $\hat{b}_{i+1} = \hat{b}_i - C_i(\hat{S}_i)$ so that $\hat{b}_i$ corresponds to the total capacity consumption of the assortment $(\hat{S}_1, \ldots, \hat{S}_m)$ in nests $1, \ldots, i-1$. Since $(\hat{\Theta}, \hat{z})$ is a feasible solution to problem (3.6)-(3.8), it satisfies the first set of constraints for state and action $(\hat{b}_i, \hat{S}_i)$ for all $i \in M$. So, we have $\hat{\Theta}_i(\hat{b}_i) \geq V_i(\hat{S}_i)^{\gamma_i}\,(R_i(\hat{S}_i) - \hat{z}) + \hat{\Theta}_{i+1}(\hat{b}_i - C_i(\hat{S}_i))$ for all $i \in M$.

Noting that $\hat{b}_{i+1} = \hat{b}_i - C_i(\hat{S}_i)$, adding these inequalities gives $v_0\,\hat{z} = \hat{\Theta}_1(c) \geq \sum_{i\in M} V_i(\hat{S}_i)^{\gamma_i}\,(R_i(\hat{S}_i) - \hat{z}) = f(\hat{z})$, where the first equality uses the fact that $(\hat{\Theta}, \hat{z})$ satisfies the second constraint in problem (3.6)-(3.8) and the second equality is by the definition of $(\hat{S}_1, \ldots, \hat{S}_m)$. So, we have $v_0\,\hat{z} \geq f(\hat{z})$. To get a contradiction, assume that $v_0\,\hat{z} > f(\hat{z})$ in the rest of the proof and let $\tilde{z}$ be such that $f(\tilde{z}) = v_0\,\tilde{z}$.

Compute the value functions $J(\tilde{z}) = \{J_i(b \,|\, \tilde{z}) : i \in M,\ b = 0, \ldots, mn\}$ through the dynamic program in (3.5) with $z = \tilde{z}$. Noting the way the value functions are computed in (3.5), we have $J_i(b \,|\, \tilde{z}) \geq V_i(S_i)^{\gamma_i}(R_i(S_i) - \tilde{z}) + J_{i+1}(b - C_i(S_i) \,|\, \tilde{z})$ for all $i \in M$, $b = 0, \ldots, mn$ and $S_i \in \mathcal{A}_i$ such that $C_i(S_i) \leq b$, which indicates that $(J(\tilde{z}), \tilde{z})$ satisfies the first set of constraints in problem (3.6)-(3.8). Furthermore, we know that $J_1(c \,|\, \tilde{z})$ provides the optimal objective value of problem (3.4) when this problem is solved with $z = \tilde{z}$, so that $J_1(c \,|\, \tilde{z}) = f(\tilde{z}) = v_0\,\tilde{z}$. Thus, the solution $(J(\tilde{z}), \tilde{z})$ satisfies the second constraint in problem (3.6)-(3.8) as well. The optimal objective value $\hat{\Theta}_1(c)$ of problem (3.6)-(3.8) must be no larger than the objective value $J_1(c \,|\, \tilde{z})$ at the feasible solution $(J(\tilde{z}), \tilde{z})$, implying $v_0\,\hat{z} = \hat{\Theta}_1(c) \leq J_1(c \,|\, \tilde{z}) = v_0\,\tilde{z}$. So, we obtain $f(\hat{z}) < v_0\,\hat{z} \leq v_0\,\tilde{z} = f(\tilde{z})$, but since $f(\cdot)$ is decreasing, we cannot have $v_0\,\hat{z} \leq v_0\,\tilde{z}$ and $f(\hat{z}) < f(\tilde{z})$, yielding a contradiction. □

To sum up, we can solve the linear program in (3.6)-(3.8) to obtain $\hat{z}$ satisfying $f(\hat{z}) = v_0\,\hat{z}$. Since $|\mathcal{F}_i(b)| \leq |\mathcal{A}_i| = O(n^3)$, there are $O(m^2 n)$ decision variables and $\sum_{i\in M} O(mn|\mathcal{A}_i|) = O(m^2 n^4)$ constraints in this linear program. Once we have $\hat{z}$, noting the discussion that follows Theorem 12, we can solve problem (3.4) with $z = \hat{z}$ to obtain an optimal solution to problem (3.1). To solve problem (3.4) with $z = \hat{z}$, we can use the dynamic program in (3.5). The dynamic program in (3.5) can be solved in $O(m^2\,n^4)$ operations and the computational effort for solving this dynamic program is negligible when compared with that for solving the linear

program in (3.6)-(3.8).

3.5 Space Constraint

In this section, we consider problem (3.1) under a space constraint. Thus, we have $C_i(S_i) = \sum_{j \in S_i} w_{ij}$ throughout this section. First, we show how to construct an approximation $f^R(\cdot)$ to $f(\cdot)$ such that $2\, f^R(z) \geq f(z)$ for all $z \in \Re_+$. Second, we show how to find $\hat{z}$ satisfying $f^R(\hat{z}) = v_0\, \hat{z}$. Third, we show how to find an assortment $(\hat{S}_1, \ldots, \hat{S}_m)$ such that $2 \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i} (R_i(\hat{S}_i) - \hat{z}) \geq f^R(\hat{z})$ and $\sum_{i \in M} C_i(\hat{S}_i) \leq c$. In this case, we obtain $4\, \Pi(\hat{S}_1, \ldots, \hat{S}_m) \geq z^*$ by Corollary 13 and $(\hat{S}_1, \ldots, \hat{S}_m)$ is a feasible solution to problem (3.1). Therefore, it follows that $(\hat{S}_1, \ldots, \hat{S}_m)$ is a 4-approximate solution to problem (3.1) under a space constraint.

3.5.1 Approximation at a Particular Point

We consider constructing an approximation $f^R(\cdot)$ to $f(\cdot)$ that satisfies $2\, f^R(z) \geq f(z)$ for all $z \in \Re_+$. Similar to our development under the cardinality constraint, the construction of our approximation to $f(\cdot)$ builds on a result that is due to [9]. In addition to a separate cardinality constraint on the assortment offered in each nest, the authors study the problem of maximizing the expected revenue obtained from each customer subject to a separate space constraint on the assortment offered in each nest. Within this setting, for any $z \in \Re_+$ and $b_i \in \Re_+$, [9] focus on the

problem

$$\max_{\substack{S_i \,:\, C_i(S_i) \leq b_i, \\ S_i \subset N}} \left\{ V_i(S_i)^{\gamma_i} \left(R_i(S_i) - z \right) \right\} \quad = \quad \max_{\substack{S_i \,:\, \sum_{j \in S_i} w_{ij} \leq b_i, \\ S_i \subset N}} \left\{ V_i(S_i)^{\gamma_i} \left(R_i(S_i) - z \right) \right\}. \tag{3.9}$$

The authors construct $O(n^2)$ different orderings between the products such that for any $z \in \Re_+$ and $b_i \in \Re_+$, a 2-approximate solution to problem (3.9) can be obtained by sorting the products according to one of these orderings, dropping the products whose space consumption exceeds b_i from consideration and using an assortment that includes some number of earliest products in this ordering. In other words, we use $\{\sigma^g : g \in \mathcal{G}_i\}$ with $|\mathcal{G}_i| = O(n^2)$ to denote the orderings constructed by [9] and $S_i(\sigma^g, k, b_i)$ to denote the assortment that includes the first k products when the products in nest i are sorted according to the ordering σ^g and the products whose space consumption exceeds b_i are dropped from consideration. In this case, [9] show that for any $z \in \Re_+$ and $b_i \in \Re_+$, a 2-approximate solution to problem (3.9) can always be found in the collection of assortments $\{S_i(\sigma^g, k, b_i) : \sigma^g \in \mathcal{G}_i,\ k = 0, \ldots, n,\ b_i \in \Re_+\}$. A critical observation is that we can consider only $b_i \in \{w_{i1}, \ldots, w_{in}\}$, rather than $b_i \in \Re_+$, without changing the collection of assortments $\{S_i(\sigma^g, k, b_i) : \sigma^g \in \mathcal{G}_i,\ k = 0, \ldots, n,\ b_i \in \Re_+\}$, since if b_i takes a value other than $\{w_{i1}, \ldots, w_{in}\}$, then we can decrease the value of b_i to the closest element in $\{w_{i1}, \ldots, w_{in}\}$ without changing the set of products whose space consumptions exceed b_i. Therefore, noting that $|\mathcal{G}_i| = O(n^2)$ and we can consider only $b_i \in \{w_{i1}, \ldots, w_{in}\}$, there are $O(n^4)$ assortments in the collection $\{S_i(\sigma^g, k, b_i) : \sigma^g \in \mathcal{G}_i,\ k = 0, \ldots, n,\ b_i \in \Re_+\}$. For notational brevity, we use $\mathcal{A}_i = \{S_{it} : t \in \mathcal{T}_i\}$ with $|\mathcal{T}_i| = O(n^4)$ to denote the collection of assortments $\{S_i(\sigma^g, k, b_i) : g \in \mathcal{G}_i,\ k = 0, \ldots, n,\ b_i \in \Re_+\}$ and obtain the next lemma. This lemma becomes useful when constructing our approximation $f^R(\cdot)$ to $f(\cdot)$.

Lemma 16. *There exists a collection of assortments* $\mathcal{A}_i = \{S_{it} : t \in \mathcal{T}_i\}$ *with* $|\mathcal{T}_i| = O(n^4)$ *such that for any* $z \in \Re_+$ *and* $b_i \in \Re_+$*, a 2-approximate solution to problem* (3.9) *can be found in* $\mathcal{A}_i$*.*

We construct our approximation to $f(\cdot)$ by focusing only on the assortments in the collections $\mathcal{A}_1, \ldots, \mathcal{A}_m$. Using the decision variables $x = \{x_i(S_i) : i \in M,\ S_i \in \mathcal{A}_i\}$, we define $f^R(\cdot)$ as

$$
\begin{aligned}
f^R(z) \quad = \quad \max \quad & \sum_{i \in M} \sum_{S_i \in \mathcal{A}_i} V_i(S_i)^{\gamma_i} (R_i(S_i) - z)\, x_i(S_i) && (3.10)\\
\text{st} \quad & \sum_{i \in M} \sum_{S_i \in \mathcal{A}_i} C_i(S_i)\, x_i(S_i) \leq c && (3.11)\\
& \sum_{S_i \in \mathcal{A}_i} x_i(S_i) = 1 \qquad \forall\, i \in M && (3.12)\\
& x_i(S_i) \geq 0 \qquad \forall\, i \in M,\ S_i \in \mathcal{A}_i, && (3.13)
\end{aligned}
$$

which corresponds to the optimal objective value of a linear program with $\sum_{i \in M} O(|\mathcal{A}_i|) = O(mn^4)$ decision variables and $O(m)$ constraints. In problem (3.2), each assortment S_i offered in nest i provides a contribution of $V_i(S_i)^{\gamma_i}(R_i(S_i) - z)$. This problem finds one assortment $S_i \subset N$ to offer in each nest i such that the total contribution over all nests is maximized and the total capacity consumption over all nests does not exceed c. Similarly, each assortment S_i offered in nest i provides a contribution of $V_i(S_i)^{\gamma_i}(R_i(S_i) - z)$ in problem (3.10)-(3.13). If we impose integrality constraints on the decision variables in problem (3.10)-(3.13), then noting the second set of constraints, this problem finds one assortment $S_i \in \mathcal{A}_i$ to offer in each nest i such that the total contribution over all nests in maximized and the total capacity consumption over all nests does not exceed c. We note that $f^R(z)$ is decreasing in z. Also, we assume that $C_i(S_i) \leq c$ for all $i \in M$ and $S_i \in \mathcal{A}_i$. If $C_i(S_i) > c$ for some $i \in M$ and $S_i \in \mathcal{A}_i$, then we can drop this assortment from $\mathcal{A}_i$ since using this assortment in problem (3.1) would yield an infeasible solution.

It is possible to use Lemma 16 to show that our approximation $f^R(\cdot)$ to $f(\cdot)$ satisfies $2\, f^R(z) \geq f(z)$ for all $z \in \Re_+$. To see this result, we let $(\hat{S}_1, \ldots, \hat{S}_m)$ be an optimal solution to problem (3.2) and $\hat{b}_i = C_i(\hat{S}_i)$. Since $(\hat{S}_1, \ldots, \hat{S}_m)$ is a feasible solution to problem (3.2), we have $\sum_{i \in M} \hat{b}_i = \sum_{i \in M} C_i(\hat{S}_i) \leq c$. Lemma 16 indicates that the collection of assortments $\mathcal{A}_i$ includes a 2-approximate solution to problem (3.9) for any $z \in \Re_+$ and $b_i \in \Re_+$. Using this result with $b_i = \hat{b}_i$ and noting the fact that $\hat{S}_i$ is a feasible solution to problem (3.9) when this problem is solved with $b_i = \hat{b}_i$, it follows that there exists $\tilde{S}_i \in \mathcal{A}_i$ such that $2\, V_i(\tilde{S}_i)^{\gamma_i}(R_i(\tilde{S}_i) - z) \geq V_i(\hat{S}_i)^{\gamma_i}(R_i(\hat{S}_i) - z)$ and $C_i(\tilde{S}_i) \leq \hat{b}_i$. Adding the last two inequalities over all $i \in M$, we have $2 \sum_{i \in M} V_i(\tilde{S}_i)^{\gamma_i}(R_i(\tilde{S}_i) - z) \geq \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i}(R_i(\hat{S}_i) - z)$ and $\sum_{i \in M} C_i(\tilde{S}_i) \leq \sum_{i \in M} \hat{b}_i \leq c$.

To obtain the desired result, we define the solution $\tilde{x}$ to problem (3.10)-(3.13) as $\tilde{x}_i(\tilde{S}_i) = 1$ for all $i \in M$ and $\tilde{x}_i(S_i) = 0$ for all $i \in M$ and $S_i \in \mathcal{A}_i \setminus \{\tilde{S}_i\}$. The solution $\tilde{x}$ is feasible to problem (3.10)-(3.13) since the definition of $\tilde{x}$ implies that $\sum_{i \in M} \sum_{S_i \in \mathcal{A}_i} C_i(S_i)\, \tilde{x}_i(S_i) = \sum_{i \in M} C_i(\tilde{S}_i) \leq c$ and $\sum_{S_i \in \mathcal{A}_i} \tilde{x}_i(S_i) = \tilde{x}_i(\tilde{S}_i) = 1$. Furthermore, the objective value provided by the solution $\tilde{x}$ for problem (3.10)-(3.13) satisfies $\sum_{i \in M} \sum_{S_i \in \mathcal{A}_i} V_i(S_i)^{\gamma_i}(R_i(S_i) - z)\, \tilde{x}_i(S_i) = \sum_{i \in M} V_i(\tilde{S}_i)^{\gamma_i}(R_i(\tilde{S}_i) - z) \geq \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i}(R_i(\hat{S}_i) - z)/2 = f(z)/2$, where the first equality follows from the definition of $\tilde{x}$, the inequality uses the fact that $2 \sum_{i \in M} V_i(\tilde{S}_i)^{\gamma_i}(R_i(\tilde{S}_i) - z) \geq \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i}(R_i(\hat{S}_i) - z)$ shown in the previous paragraph and the second equality is by the fact that $(\hat{S}_1, \ldots, \hat{S}_m)$ is an optimal solution to problem (3.2). Thus, there exists a feasible solution to problem (3.10)-(3.13) providing an objective value for this problem that is at least $f(z)/2$, which implies that the optimal objective value $f^R(z)$ of problem (3.10)-(3.13) satisfies $f^R(z) \geq f(z)/2$, establishing the desired result.

3.5.2 Finding the Fixed Point

We consider the problem of finding the value of $\hat{z}$ that satisfies $f^R(\hat{z}) = v_0\,\hat{z}$. Noting that $f^R(z)$ is given by the optimal objective value of the linear program in (3.10)-(3.13), we use the dual of this problem to find the value of $\hat{z}$ that satisfies $f^R(\hat{z}) = v_0\,\hat{z}$. In particular, associating the dual variables Δ and $y = \{y_i : i \in M\}$ respectively with the two sets of constraints in problem (3.10)-(3.13), we propose solving the linear program

$$\begin{aligned}
\min \quad & c\,\Delta + \sum_{i \in M} y_i && && (3.14)\\
\text{st} \quad & C_i(S_i)\,\Delta + y_i \geq V_i(S_i)^{\gamma_i}(R_i(S_i) - z) && \forall\, i \in M,\ S_i \in \mathcal{A}_i && (3.15)\\
& c\,\Delta + \sum_{i \in M} y_i = v_0\,z && && (3.16)\\
& \Delta \geq 0,\ y_i \text{ is free},\ z \text{ is free} && \forall\, i \in M && (3.17)
\end{aligned}$$

to find $\hat{z}$ satisfying $f^R(\hat{z}) = v_0\,\hat{z}$. The decision variables are Δ, y and z in the problem above. If we drop the second constraint in problem (3.14)-(3.17) and minimize the objective function subject to the first set of constraints for a fixed value of z, then this problem corresponds to the dual of problem (3.10)-(3.13). With the second constraint added, problem (3.14)-(3.17) allows us to find $\hat{z}$ that satisfies $f^R(\hat{z}) = v_0\,\hat{z}$, as shown in the next theorem.

Theorem 17. *Letting* $(\hat{\Delta}, \hat{y}, \hat{z})$ *be an optimal solution to problem* (3.14)-(3.17)*,* $\hat{z}$ *satisfies* $f^R(\hat{z}) = v_0\,\hat{z}$*.*

Proof. Associating the dual variables Δ and $y = \{y_i : i \in M\}$ with the two sets of

constraints in problem (3.10)-(3.13), the dual of this problem is

$$f^R(z) \;=\; \min \quad c\,\Delta + \sum_{i\in M} y_i \tag{3.18}$$

$$\text{st} \quad C_i(S_i)\,\Delta + y_i \geq V_i(S_i)^{\gamma_i}(R_i(S_i) - z) \qquad \forall\, i \in M,\ S_i \in \mathcal{A}_i \tag{3.19}$$

$$\Delta \geq 0,\ y_i \text{ is free} \qquad \forall\, i \in M. \tag{3.20}$$

Therefore, the solution $(\hat{\Delta}, \hat{y})$ is feasible to problem (3.18)-(3.20) when we solve this problem with $z = \hat{z}$, which implies that $f^R(\hat{z}) \leq c\,\hat{\Delta} + \sum_{i\in M} \hat{y}_i = v_0\,\hat{z}$, where the equality follows from the fact that $(\hat{\Delta}, \hat{y}, \hat{z})$ is a feasible solution to problem (3.14)-(3.17). To get a contradiction, assume that the last inequality is strict so that $f^R(\hat{z}) < v_0\,\hat{z}$. We let $\tilde{z}$ be such that $f^R(\tilde{z}) = v_0\,\tilde{z}$ and $(\tilde{\Delta}, \tilde{y})$ be an optimal solution to problem (3.18)-(3.20) when we solve this problem with $z = \tilde{z}$. Thus, we get $v_0\,\tilde{z} = f^R(\tilde{z}) = c\,\tilde{\Delta} + \sum_{i\in M} \tilde{y}_i$, which indicates that $(\tilde{\Delta}, \tilde{y}, \tilde{z})$ is a feasible solution to problem (3.14)-(3.17). In this case, it follows that $v_0\,\tilde{z} = f^R(\tilde{z}) = c\,\tilde{\Delta} + \sum_{i\in M} \tilde{y}_i \geq c\,\hat{\Delta} + \sum_{i\in M} \hat{y}_i = v_0\,\hat{z} > f^R(\hat{z})$, where the first inequality is by the fact that $(\tilde{\Delta}, \tilde{y}, \tilde{z})$ is a feasible, but not necessarily an optimal solution to problem (3.14)-(3.17). The last chain of inequalities yields $v_0\,\tilde{z} \geq v_0\,\hat{z}$ and $f^R(\tilde{z}) > f^R(\hat{z})$, which contradict the fact that $f^R(\cdot)$ is a decreasing function. $\square$

3.5.3 Construction of an Approximate Assortment

By the earlier discussion in this section, our approximation $f^R(\cdot)$ to $f(\cdot)$ satisfies $2\,f^R(z) \geq f(z)$ for all $z \in \Re_+$. Furthermore, we can find the value of $\hat{z}$ that satisfies $f^R(\hat{z}) = v_0\,\hat{z}$ by solving the linear program in (3.14)-(3.17). In the remainder of this section, we consider the problem of finding an assortment $(\hat{S}_1, \ldots, \hat{S}_m)$ such that $2\sum_{i\in M} V_i(\hat{S}_i)^{\gamma_i}(R_i(\hat{S}_i) - \hat{z}) \geq f^R(\hat{z})$ and $\sum_{i\in M} C_i(\hat{S}_i) \leq c$. In this case, Corollary

13 implies that we have $4\,\Pi(\hat{S}_1,\ldots,\hat{S}_m) \geq z^*$ and $(\hat{S}_1,\ldots,\hat{S}_m)$ is a feasible solution to problem (3.1). So, $(\hat{S}_1,\ldots,\hat{S}_m)$ is a 4-approximate solution to problem (3.1).

It is a simple exercise in linear programming duality to show that any basic optimal solution to problem (3.10)-(3.13) includes at most two fractional components; see [38]. We let $\hat{x}$ be a basic optimal solution to problem (3.10)-(3.13) when we solve this problem with $z = \hat{z}$. We make two observations. First, if $\hat{x}_{i'}(P_{i'}) \in (0,1]$ for some nest $i' \in M$ and assortment $P_{i'} \in \mathcal{A}_{i'}$, then noting the second set of constraints in problem (3.10)-(3.13), there must be some other assortment $Q_{i'} \in \mathcal{A}_{i'}$ such that $\hat{x}_{i'}(Q_{i'}) \in [0,1)$ as well. Second, since $\hat{x}$ has as most two fractional components, there can be no other fractional component of $\hat{x}$. In this case, noting the second set of constraints in problem (3.10)-(3.13) once more, it follows that for each nest $i \in M \setminus \{i'\}$, there exists a single assortment $\tilde{S}_i$ such that $\hat{x}_i(\tilde{S}_i) = 1$. Therefore $\{\hat{x}_i(\tilde{S}_i) : i \in M \setminus \{i'\}\} \cup \{\hat{x}_{i'}(P_{i'})\} \cup \{\hat{x}_{i'}(Q_{i'})\}$ includes all components of $\hat{x}$ taking strictly positive values.

In the rest of the discussion, we assume that the basic optimal solution $\hat{x}$ has two fractional components. In particular, noting the second set of constraints in problem (3.10)-(3.13), $\hat{x}$ cannot have one fractional component and the result holds in a straightforward fashion when $\hat{x}$ has no fractional components. As described in the previous paragraph, if the basic optimal solution $\hat{x}$ to problem (3.10)-(3.13) has two fractional components, then there exist some nest $i' \in M$ and assortments $P_{i'}, Q_{i'} \in \mathcal{A}_{i'}$ such that $\hat{x}_{i'}(P_{i'}), \hat{x}_{i'}(Q_{i'}) \in (0,1)$ and there is no other fractional component of $\hat{x}$. Without loss of generality, we assume that $C_{i'}(P_{i'}) \leq C_{i'}(Q_{i'})$. Furthermore, for each nest $i \in M \setminus \{i'\}$, there exists a single assortment $\tilde{S}_i$ such that $\hat{x}_i(\tilde{S}_i) = 1$. Using the solution $\hat{x}$, we construct two assortments $(\hat{S}^1_1,\ldots,\hat{S}^1_m)$ and $(\hat{S}^2_1,\ldots,\hat{S}^2_m)$ as follows. The first one of these as-

sortments is constructed as $(\hat{S}_1^1, \ldots, \hat{S}_m^1) = (\tilde{S}_1, \ldots \tilde{S}_{i'-1}, P_{i'}, \tilde{S}_{i'+1}, \ldots, \tilde{S}_m)$. In other words, the assortment $(\hat{S}_1^1, \ldots, \hat{S}_m^1)$ uses the components of the solution $\hat{x}$ that take value one, along with the fractional component of the solution $\hat{x}$ with the smaller capacity consumption. We construct the second assortment as $(\hat{S}_1^2, \ldots, \hat{S}_m^2) = (\emptyset, \ldots, \emptyset, Q_{i'}, \emptyset, \ldots, \emptyset)$, offering the subset $Q_{i'}$ in nest i', but offering empty subsets in all of the other nests. A crucial observation is that the two assortments $(\hat{S}_1^1, \ldots, \hat{S}_m^1)$ and $(\hat{S}_1^2, \ldots, \hat{S}_m^2)$ as defined above collectively include all components of the solution $\hat{x} = \{\hat{x}_i(S_i) : i \in M,\ S_i \in \mathcal{A}_i\}$ taking a strictly positive value. In this case, we get

$$\begin{aligned} f^R(\hat{z}) &= \sum_{i \in M} \sum_{S_i \in \mathcal{A}_i} V_i(S_i)^{\gamma_i} (R_i(S_i) - \hat{z})\, \hat{x}_i(S_i) \leq \sum_{i \in M} V_i(\hat{S}_i^1)^{\gamma_i} (R_i(\hat{S}_i^1) - \hat{z}) \\ &\quad + \sum_{i \in M} V_i(\hat{S}_i^2)^{\gamma_i} (R_i(\hat{S}_i^2) - \hat{z}) \\ &\leq 2 \max \left\{ \sum_{i \in M} V_i(\hat{S}_i^1)^{\gamma_i} (R_i(\hat{S}_i^1) - \hat{z}), \sum_{i \in M} V_i(\hat{S}_i^2)^{\gamma_i} (R_i(\hat{S}_i^2) - \hat{z}) \right\}, \end{aligned}$$

where the first inequality is by the fact that if $\hat{x}_i(S_i) > 0$ for some $i \in M$ and $S_i \in \mathcal{A}_i$, then we have $S_i = \hat{S}_i^1$ or $\hat{S}_i = \hat{S}_i^2$. Therefore, the chain of inequalities above shows that if we choose $(\hat{S}_1, \ldots, \hat{S}_m)$ as one of the assortments $(\hat{S}_1^1, \ldots, \hat{S}_m^1)$ and $(\hat{S}_1^2, \ldots, \hat{S}_m^2)$, then $(\hat{S}_1, \ldots, \hat{S}_m)$ satisfies $2 \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i} (R_i(\hat{S}_i) - \hat{z}) \geq f^R(\hat{z})$. Furthermore, we note that both of the solutions $(\hat{S}_1^1, \ldots, \hat{S}_m^1)$ and $(\hat{S}_1^2, \ldots, \hat{S}_m^2)$ are feasible to problem (3.1). The solution $(\hat{S}_1^2, \ldots, \hat{S}_m^2)$ is feasible since this solution only offers $Q_{i'}$ in nest i' and we have $\sum_{i \in M} C_i(\hat{S}_i^2) = C_{i'}(Q_{i'}) \leq c$, where the last inequality uses the assumption that $C_i(S_i) \leq c$ for all $i \in M$ and $S_i \in \mathcal{A}_i$. To see the feasibility of the solution $(\hat{S}_1^1, \ldots, \hat{S}_m^1)$ to problem (3.1), we observe that $\sum_{i \in M} C_i(\hat{S}_i^1) = \sum_{i \in M \setminus \{i'\}} C_i(\tilde{S}_i) + C_{i'}(P_{i'}) \leq \sum_{i \in M \setminus \{i'\}} C_i(\tilde{S}_i) + C_{i'}(P_{i'})\, \hat{x}_{i'}(P_{i'}) + C_{i'}(Q_{i'})\, \hat{x}_{i'}(Q_{i'}) = \sum_{i \in M} \sum_{S_i \in \mathcal{A}_i} C_i(S_i)\, \hat{x}_i(S_i) \leq c$, where the first inequality uses the fact that $C_{i'}(P_{i'}) \leq C_{i'}(Q_{i'})$ and $\hat{x}_{i'}(P_{i'}) + \hat{x}_{i'}(Q_{i'}) = 1$ by the second set of constraints in problem (3.10)-(3.13) and the second equality uses the fact that

$\{\hat{x}_i(\tilde{S}_i) : i \in M \setminus \{i'\}\} \cup \{\hat{x}_{i'}(P_{i'})\} \cup \{\hat{x}_{i'}(Q_{i'})\}$ correspond to all components of $\hat{x}$ taking strictly positive values.

To sum up, we can solve the linear program in (3.14)-(3.17) to find $\hat{z}$ satisfying $f^R(\hat{z}) = v_0\,\hat{z}$. This linear program has $O(m)$ decision variables and $\sum_{i\in M} O(|\mathcal{A}_i|) = O(mn^4)$ constraints. Once we have $\hat{z}$, we can solve problem (3.10)-(3.13) with $z = \hat{z}$ to obtain an optimal solution $\hat{x}$ and construct the assortments $(\hat{S}_1^1, \ldots, \hat{S}_m^1)$ and $(\hat{S}_1^2, \ldots, \hat{S}_m^2)$ as in the previous paragraph. If we choose $(\hat{S}_1, \ldots, \hat{S}_m)$ as one of the assortments $(\hat{S}_1^1, \ldots, \hat{S}_m^1)$ and $(\hat{S}_1^2, \ldots, \hat{S}_m^2)$, then $(\hat{S}_1, \ldots, \hat{S}_m)$ satisfies $2\sum_{i\in M} V_i(\hat{S}_i)^{\gamma_i}(R_i(\hat{S}_i) - \hat{z}) \geq f^R(\hat{z})$ and $(\hat{S}_1, \ldots, \hat{S}_m)$ is a feasible solution to problem (3.1). Since our approximation $f^R(\cdot)$ to $f(\cdot)$ satisfies $2\,f^R(z) \geq f(z)$ for all $z \in \Re_+$, by Corollary 13, we obtain $4\,\Pi(\hat{S}_1, \ldots, \hat{S}_m) \geq z^*$, so that $(\hat{S}_1, \ldots, \hat{S}_m)$ is a 4-approximate solution to problem (3.1). Thus, if we check the expected revenue provided by each one of the assortments $(\hat{S}_1^1, \ldots, \hat{S}_m^1)$ and $(\hat{S}_1^2, \ldots, \hat{S}_m^2)$ for problem (3.1) and pick the best one, then we obtain a 4-approximate solution to problem (3.1).

3.6 Numerical Experiments

In this section, our goal is to numerically test the performance of the 4-approximation algorithm described in Section 4.8.1. Since we can obtain the optimal solution to problem (3.1) when we have a cardinality constraint, we do not provide numerical experiments under a cardinality constraint.

3.6.1 Numerical Setup

In our numerical experiments, we randomly generate a large number of problem instances. We generate each problem instance by using the following procedure. We set the number of nests as $m = 3$ or $m = 5$ and the number of products in each nest as $n = 15$ or $n = 30$. To come up with the revenues and preference weights of the products, we sample C_{ij} from the uniform distribution over $[0, 1]$. Similarly, we sample X_{ij} and Y_{ij} from the uniform distribution over $[0.75, 1.25]$. We set the revenue and preference weight of product j in nest i respectively as $r_{ij} = 10 \times C_{ij}^2 \times X_{ij}$ and $v_{ij} = 10 \times (1 - C_{ij}) \times Y_{ij}$. Through C_{ij}, we ensure that the products having larger revenues generally tend to have smaller preference weights, indicating more expensive products tend to be less attractive. Squaring C_{ij} in the expression for r_{ij} skews the distribution of the revenues so that we have a small number of products with large revenues. Through X_{ij} and Y_{ij}, we incorporate idiosyncratic noise into the revenues and preference weights so that not all products with large revenues have small preference weights. We sample the dissimilarity parameter γ_i for each nest i from the uniform distribution over $[0.25, 0.75]$. We set the preference weight v_0 of the no purchase option such that the probability of no purchase is 0.4 even when we offer all products in all nests. We sample the space requirement w_{ij} of product j in nest i from the uniform distribution over $[1, 10]$. We set the capacity availability as $c = \kappa \sum_{i \in M} \sum_{j \in N} w_{ij}$, corresponding to a κ fraction of the total space consumption of all products in all nests. We use $\kappa = 0.1$, $\kappa = 0.15$ or $\kappa = 0.2$.

We vary (m, n, κ) over $\{3, 5\} \times \{15, 30\} \times \{0.1, 0.15, 0.2\}$ to get 12 parameter combinations. In each parameter combination, we generate 5,000 individual problem instances by using the approach in the previous paragraph. For each problem

instance, we use the approach in Section 4.8.1 to obtain a 4-approximate solution. In Online Appendix B.2, we also give a linear program that provides an upper bound on the optimal expected revenue for a particular problem instance. To assess the quality of the 4-approximate solution, we check the gap between the expected revenue from the 4-approximate solution and the upper bound on the optimal expected revenue.

3.6.2 Numerical Results

Our numerical results are given in Table 4.1. The first column in this table shows the parameter combinations by using (m, n, κ). We recall that we generate 5,000 individual problem instances in each parameter combination. For each problem instance, we use the approach in Section 4.8.1 to obtain a 4-approximate solution to problem (3.1). We use Rev^p to denote the expected revenue obtained by the 4-approximate solution for problem instance p. Using Bnd^p to denote the upper bound on the optimal total expected revenue for problem instance p, the second column in Table 4.1 gives the average percent gap between Rev^p and Bnd^p, where average is taken over all 5,000 problem instances in a parameter combination. The third and fourth columns respectively give the 95th percentile and maximum of the percent gaps between Rev^p and Bnd^p over all 5,000 problem instances in a particular parameter combination. Thus, the second, third and fourth columns respectively give the average, 95th percentile and maximum of the data $\{100 \times (\mathsf{Bnd}^p - \mathsf{Rev}^p)/\mathsf{Bnd}^p : p = 1, \ldots, 5{,}000\}$. In this way, these columns give an indication of the optimality gaps of the 4-approximate solutions. The fifth column gives the number of products in the 4-approximate solution, averaged over all 5,000 problem instances. The sixth column gives the number of problem instances

for which the percent gap between Rev^p and Bnd^p is less than 1%. Similarly, the seventh, eighth and ninth columns give the number of problem instances where the percent gap between Rev^p and Bnd^p is respectively less than 2.5%, 5% and 10%. The tenth and eleventh columns attempt to give a feel for the tightness of the space constraint. The tenth column shows the average number of products in the optimal assortment when there are no space constraints. The eleventh column shows the number of problem instances for which this unconstrained solution violates the space constraint. For comparison, we also use a greedy algorithm to obtain a heuristic solution to problem (3.1). The greedy algorithm starts with the empty assortment, finds the product that provides the largest improvement in the expected revenue per unit of space consumption and adds this product to the assortment, until there is no available space or no improvement in the expected revenue. The twelfth, thirteenth and fourteenth columns respectively show the average, 95th percentile and maximum of the percent gaps between the expected revenue obtained by the greedy algorithm and the upper bound on the optimal expected revenue. So, using Gre^p to denote the expected revenue obtained by the greedy algorithm for problem instance p, these three columns show the average, 95th percentile and maximum of the data $\{100 \times (\mathsf{Bnd}^p - \mathsf{Gre}^p)/\mathsf{Bnd}^p : p = 1, \ldots, 5{,}000\}$.

Our results indicate that our 4-approximation algorithm performs quite well. Over all problem instances, the average optimality gap of this algorithm is no larger than 1.56%. In 58,034 out of all 60,000 problem instances, the optimality gaps of the 4-approximate solutions are less than 5%. As a general trend, the optimality gaps tend to get smaller as κ gets larger. As κ gets larger, the capacity availability gets larger and each product occupies a smaller fraction of the available capacity. So, problem instances where each product occupies a smaller fraction of the available capacity appear to be easier to approximate. This observation is aligned

with the intuition that the linear programming relaxation of a knapsack problem becomes tighter as each item occupies a smaller fraction of the knapsack capacity. In particular, it is known that if each item occupies no larger than a fraction ϵ of the knapsack capacity, then the optimal objective value of the linear programming relaxation exceeds the optimal objective value of a knapsack problem by at most a factor of $1/(1-\epsilon)$. The most problematic parameter combination in Table 4.1 is $(3, 15, 0.1)$, corresponding to a small value of κ with $\kappa = 0.1$. Even for this parameter combination, in more than 75% of the problem instances, the optimality gap of the 4-approximate solution is no larger than 5%. Also, we note that the reported optimality gaps are pessimistic estimates, since these optimality gaps are obtained by comparing the expected revenue from an assortment with an upper bound on the optimal expected revenue, rather than the optimal expected revenue itself. The greedy algorithm performs noticeably worse than the 4-approximation. There are parameter combinations such as $(3, 15, 0.20)$, where the 95th percentile of the optimality gaps from the 4-approximation algorithm is 3.86%, but the 95th percentile of the optimality gaps from the greedy algorithm is 8.16%. The running times for the 4-approximation algorithm are reasonable. We use Java 1.6.033 on an Intel Xeon 2.00 GHz CPU and Gurobi 5.1.0 as the linear programming solver. For the largest problem instances with $m = 5$ and $n = 30$, the average running time for the 4-approximation algorithm is 3.56 seconds.

3.7 Conclusions

We gave tractable methods to solve assortment problems under the NL model when there is a cardinality or space constraint on the assortment offered over all nests. As

a direction for future research, the 4-approximation algorithm does not provide any guidance as to how we can obtain better solutions if we are willing to increase the computational effort. [9] show how to generate candidate assortments that tradeoff running time with solution quality. Furthermore, [7] develop approximations to multiple choice knapsack problems that tradeoff running time with solution quality. It is interesting to see whether we can join these two approaches to develop an approximation algorithm for the assortment problem under a space constraint that tradeoff running time with solution quality.

Param. Combin. (m, n, κ)	% Gap btw. Rev^p, Bnd^p Avg.	95th	Max.	Avg. Assr. Size	No. Prob. with Certain % Gap btw. Rev^p, Bnd^p 1%	2.5%	5%	10%	Uncn. Assr. Size	No. Capac. Prbs.	% Gap btw. Gre^p, Bnd^p Avg.	95th	Max.
$(3, 15, 0.20)$	1.48	3.86	8.68	9.89	2,196	4,090	4,939	5,000	17.02	4,899	2.78	8.16	28.11
$(3, 15, 0.15)$	2.21	5.62	14.36	8.06	1,423	3,258	4,610	4,996	17.02	4,985	3.46	9.61	31.87
$(3, 15, 0.10)$	3.35	8.38	21.53	6.13	861	2,322	3,875	4,898	17.02	4,999	4.51	11.57	36.78
$(3, 30, 0.20)$	0.82	1.95	4.95	20.09	3,430	4,924	5,000	5,000	33.45	4,941	1.43	4.35	13.25
$(3, 30, 0.15)$	1.19	2.75	7.78	16.36	2,464	4,622	4,996	5,000	33.45	4,996	1.82	5.35	15.07
$(3, 30, 0.10)$	1.79	4.06	7.36	12.45	1,551	3,740	4,926	5,000	33.45	5,000	2.39	6.55	17.50
$(5, 15, 0.20)$	1.05	2.34	5.18	16.88	2,776	4,828	4,999	5,000	28.47	4,982	2.31	6.03	13.70
$(5, 15, 0.15)$	1.54	3.41	6.05	13.78	1,798	4,163	4,987	5,000	28.47	4,999	2.83	7.00	15.10
$(5, 15, 0.10)$	2.26	5.16	10.33	10.53	1,086	3,196	4,704	4,999	28.47	5,000	3.44	8.13	17.30
$(5, 30, 0.20)$	0.68	1.37	4.52	33.82	4,097	4,992	5,000	5,000	56.06	4,993	1.30	3.27	7.75
$(5, 30, 0.15)$	0.96	1.88	3.96	27.57	3,010	4,972	5,000	5,000	56.06	5,000	1.63	3.99	9.70
$(5, 30, 0.10)$	1.35	2.73	5.09	21.04	1,907	4,606	4,998	5,000	56.06	5,000	2.03	4.87	13.53
Avg./Total	1.56	3.63	8.32		26,599	49,713	58,034	59,893			2.49	6.57	18.30

Table 3.1: Performance of the 4-approximate solutions and the greedy algorithm on 5,000 randomly generated problem instances.

Chapter 4

Bounding Optimal Expected Revenues for Assortment Optimization under a Mixture of Multinomial Logit Choice Models

4.1 Introduction

In this chapter, we study the assortment problem when customer purchase patterns are governed by a MMNL choice model. In our problem setting, a firm wants to find a set of products to offer to its customers. There is a fixed revenue associated with each product. An arriving customer may be one of multiple

customer types. The firm does not known the type of an arriving customer, but it has access to the probability that an arriving customer is of a particular type. Customers choose among the offered products according to the multinomial logit model and customers of different types choose according to different multinomial logit models whose parameters depend on the type of the customer. This choice model is known as the MMNL model. The goal of the firm is to find a set or an assortment of products to offer to its customers so as to maximize the expected revenue obtained from each customer. [2] show that this assortment problem is NP-complete, give a mixed integer programming formulation to obtain the optimal solution and provide computational experiments that demonstrate that a greedy heuristic performs quite well when compared with the optimal solutions obtained through the mixed integer programming formulation. One shortcoming of using a heuristic is that we use a heuristic simply due to the fact that we cannot obtain the optimal solution and there is no immediate way of being confident that the solution provided by a heuristic is actually a good one. In this chapter, motivated by the difficulty of obtaining optimal solutions and evaluating the quality of the solutions provided by a heuristic, we develop a method to obtain upper bounds on the optimal expected revenue in our assortment problem. Thus, we can check the gap between the expected revenue from the solution provided by a heuristic and the upper bound on the optimal expected revenue to assess the optimality gap of the heuristic.

Our method for obtaining an upper bound on the optimal expected revenue has two crucial pieces. First, a natural approach for obtaining an upper bound on the optimal expected revenue is to assume that the firm knows the type of an arriving customer. In this case, we can focus on each customer type one by one and separately find an assortment that maximizes the expected revenue from each

customer type. This approach essentially allows us to offer different assortments of products to customers of different types, whereas our assortment problem requires that we find a single assortment to offer to all customer types. [41] show that if we focus on one customer type at a time, then the assortment that maximizes the expected revenue from a single customer type can be obtained efficiently. This idea provides an efficient approach for obtaining an upper bound on the optimal expected revenue, but the upper bound provided by this idea can be quite loose since the assortments that maximize the expected revenues from different customer types can be drastically different from each other. To overcome this shortcoming, we still focus on each customer type one by one, but use penalty parameters to penalize a product that appears in the assortment offered to one customer type but does not appear in the assortment offered to another customer type. In this way, our goal is to synchronize the assortments offered to different customer types. We choose the penalty parameters from a certain set that ensures that we continue obtaining an upper bound on the optimal expected revenue even if we penalize the presence or absence of the products in the assortments offered to different customer types. We show that we can choose a good set of penalty parameters by solving a convex program.

Second, as we focus on each customer type one by one and use penalty parameters to penalize the presence or absence of the products, we obtain assortment problems with a single customer type, but the penalty parameters play the role of a fixed cost for offering a product. [18] show that if customers choose according to the multinomial logit model, then the assortment problem with a fixed cost for offering a product is NP-complete, even when there is a single customer type. To deal with this difficulty, we develop a new approximation to the assortment problem with a single customer type and a fixed cost for offering a product. Our

approximation is based on the assumption that the probability that a customer leaves without making a purchase can take on values over a prespecified grid. We design the grid so that we continue obtaining an upper bound on the optimal expected revenue. Denser grid points provide a tighter upper bound at the expense of larger computational effort. We give guidelines for choosing a good set of grid points to balance the tightness of the upper bound with the computational effort.

To our knowledge, our approach is a unique practical method to check the quality of solutions in assortment problems under a MMNL model. Computational experiments indicate that our approach can obtain quite tight upper bounds on the optimal expected revenues. We consider a large set of problem instances with large numbers of products so that we cannot obtain the optimal solutions in a reasonable amount of run time. In more than 98% of the problem instances, the upper bounds from our approach are within 0.5% of the optimal expected revenues. On average, the upper bounds from our approach deviate from the optimal expected revenues by 0.15%. In the process, we support the findings of [2] on large problem instances for which we cannot compute the optimal solutions and demonstrate that the optimality gaps of the greedy heuristic are within a fraction of a percent. Without tight upper bounds on the optimal expected revenues, it would not be possible to obtain such an accurate characterization of the optimality gaps of the greedy heuristic.

There are three papers that particularly motivated us to construct upper bounds when customers choose according to a MMNL model. First, [26] show that a MMNL model can approximate any random utility choice model, where a customer associates random utilities with the products, choosing the product with the largest utility. This result holds irrespective of the joint distribution of the

random utilities. So, a MMNL model is a powerful choice model and solving assortment problems under this choice model can have direct implications on solving assortment problems under arbitrary random utility choice models. Second, [40] considers assortment problems under a MMNL model, but he focuses on a network revenue management setting. The author computes an upper bound on the optimal expected revenue by preallocating the available capacity to different customer types and his approach turns out to be equivalent to assuming that the firm knows the type of an arriving customer, so that the firm can offer different assortments to customers of different types. He does not use any penalty parameters to harmonize the assortments offered to different customer types. In our assortment problems, this approach can yield quite poor upper bounds and we see a need to improve this approach. The gap between the upper bounds provided by this approach and the optimal expected revenues can exceed 14%.

Finally, as mentioned above, [2] show that the assortment problem under a MMNL model can be formulated as a mixed integer program. They demonstrate that a greedy heuristic performs quite well when compared with the optimal solutions obtained by the mixed integer program. It is difficult to evaluate the optimality gap of the greedy heuristic for large problem instances and a good upper bound on the optimal expected revenue becomes useful in this regard. Furthermore, a tempting approach to obtain an upper bound on the optimal expected revenue is to solve the linear programming relaxation of their mixed integer program, but we establish that the upper bound from this linear programming relaxation can be as poor as focusing on each customer type one by one without using any penalty parameters. In other words, the upper bound from the linear programming relaxation can correspond to the upper bound from the approach in [40].

To sum up, we make the following contributions in this chapter. 1) We develop a new approach to obtain an upper bound on the optimal expected revenue in assortment problems under a MMNL model. Our approach finds an assortment that maximizes the expected revenue from each customer type, but we use penalty parameters to synchronize the assortments offered to different customer types. This strategy requires solving assortment problems with a single customer type, but with a fixed cost for offering a product. We show how to approximate such assortment problems by assuming that the probability that a customer leaves without making a purchase lies on a prespecified grid. 2) We show how to choose a good set of penalty parameters by solving a convex program. 3) We show how to choose a good set of grid points. Denser grid points yield tighter upper bounds at the expense of larger computational effort, but we show that if we simply use exponential grid points of the form $\{(1+\rho)^{-k+1} : k = 1, 2, \ldots\}$ for some $\rho > 0$, then no other set of grid points, no matter how dense it is, can improve the upper bound by more than a factor of $1+\rho$. 4) We show that the linear programming relaxation of the mixed integer program given by [2] can be as loose as the upper bound obtained under the assumption that the firm knows the type of an arriving customer. 5) Our approach for obtaining an upper bound on the optimal expected revenue is flexible enough that we can extend it to the case where there is a constraint on the total space consumption of the offered products or where customers choose according to a mixture of NL models. We show how to make such extensions.

The chapter is organized as follows. In Section 4.2, we review the related literature. In Section 4.3, we formulate the assortment problem under a MMNL model and we present our approach for obtaining an upper bound, which is based on offering different assortments to different customer types, but uses penalty parameters to synchronize the assortments offered to different customer types. In

this way, we obtain assortment problems with a single customer type but with a fixed cost for offering a product. In Section 4.4, we show how to approximate such assortment problems by assuming that the probability of not making a purchase lies on a prespecified grid. In Section 4.5, we show how to choose a good set of penalty parameters. In Section 4.6, we show how to choose a good set of grid points. In Section 4.7, we relate our approach for obtaining upper bounds to a Lagrangian relaxation strategy on an appropriate formulation of our assortment problem. This development requires more notational overhead than the path we follow. So, we defer this development towards the end of the chapter. Since our approach can be cast as a Lagrangian relaxation strategy for a nonconvex program, it is difficult to get theoretical tightness guarantees for our upper bounds and there are pathological problem instances that suffer from a large duality gap. In Section 4.8, we make extensions to the case where there is a constraint on the total space consumption of the offered products or where customers choose according to a mixture of NL models. In Section 4.9, we give computational experiments on both large problem instances and small problem instances with a special structure. In Section 4.10, we conclude.

4.2 Literature Review

Our work is related to assortment problems under the multinomial logit model. [8] and [41] consider assortment problems under the multinomial logit model with a single customer type and show that the optimal assortment can be obtained efficiently by focusing on assortments that include a certain number of products with the largest revenues. [2] and [30] consider the assortment problem under

a MMNL model. They show that the problem is NP-complete, give a mixed integer programming formulation of the problem, present valid cuts to tighten this formulation and experiment with a greedy heuristic. [34] consider the assortment problem when there is a constraint on the number of products that can be offered and show that the optimal assortment can be found efficiently when there is a single customer type. [16] consider simple heuristics for assortment problems and show that these heuristics obtain the optimal assortment under the multinomial logit model with a single customer type. [11] and [47] study assortment problems under the multinomial logit model, where customers become more likely to leave without a purchase when the offered assortment lacks variety. [4] give linear programming formulations for assortment problems with constraints on the offered assortment, when customers choose according to the multinomial logit model with a single customer type. [35] consider the assortment problem under a MMNL model, show that the problem is NP-complete even when there are two customer types and give performance guarantees for a certain class of assortments. [5] give approximation schemes for various assortment problems. These approximation schemes can get cumbersome when the number of customer types is large.

There is assortment optimization work under other choice models. [4], [21], [9] and [21] consider assortment problems when customers choose according a NL model with a single customer type and show that the problem is tractable. [6] consider a choice model where each customer arrives with a particular ordering of products in mind and purchases the first product in the ordering that is offered. They focus on estimating the parameters of the choice model in a way consistent with observed sales data. [1] consider a choice model, where if a customer finds that the product he is interested in is not available, then he makes a transition to another product according to a Markov chain and considers purchasing the other

product, until he reaches a product that is available or reaches the option of leaving without purchasing anything. The authors show that the assortment problem is tractable under this choice model.

There is related literature on network revenue management models incorporating customer choice behavior. In this setting, an airline sells itinerary products over a flight network. Customers arriving into the system choose among the offered itineraries and the goal is to dynamically adjust the set of available itineraries over time so as to maximize the expected revenue obtained over the selling horizon. A common approach for such network revenue management problems is to formulate deterministic linear programming approximations. Examples of such approximations can be found in [8], [24], [49], [19], [29], [20] and [44]. Usually, the decision variables in these approximations correspond to the number of time periods during which a particular subset of itineraries is made available. Since there is one decision variable for each subset of itineraries, the number of decision variables can be large and it is common to solve the approximations by using column generation. The column generation subproblems in this setting precisely correspond to the assortment problem that we consider in this chapter when customers choose according to a MMNL model.

Although [26] do not focus on solving assortment problems, their work demonstrates that a MMNL model is a powerful choice model as it can accurately approximate any choice model that is based on random utility maximization. [45] consider the problem of estimating the parameters of the multinomial logit model with a single customer type from sales data. [17] estimate the parameters of a MMNL model from sales data and they focus on the case where the sets of products offered to customers are not observable.

4.3 Problem Formulation and Decomposition Approach

We use N to denote the set of possible products that we can offer to customers. The revenue associated with product j is r_j. We use G to denote the set of customer types. The probability that a customer of type g arrives into the system is α^g, where we have $\sum_{g \in G} \alpha^g = 1$. We use the vector $x = \{x_j : i \in N\} \in \{0,1\}^{|N|}$ to capture the set of products that we offer to the customers, where we have $x_j = 1$ if product j is offered, otherwise we have $x_j = 0$. A customer of a certain type makes a choice among the offered products according to the multinomial logit model whose parameters depend on the type of the customer. In particular, a customer of type g associates the preference weight v_j^g with product j. For all customer types, we normalize the preference weight of the no purchase option to one. In this case, if the set of products that we offer to the customers is captured by the vector x, then a customer of type g purchases product j with probability $P_j^g(x) = v_j^g \, x_j / (1 + \sum_{i \in N} v_i^g \, x_i)$. Thus, if the set of products that we offer to the customers is captured by the vector x, then the expected revenue obtained from a customer is given by $\sum_{g \in G} \alpha^g \, \sum_{j \in N} r_j \, P_j^g(x)$. Noting the definition of $P_j^g(x)$, we can find the set of products that maximizes the expected revenue obtained from a customer by solving the problem

$$Z^* = \max_{x \in \{0,1\}^{|N|}} \left\{ \sum_{g \in G} \alpha^g \frac{\sum_{j \in N} r_j \, v_j^g \, x_j}{1 + \sum_{j \in N} v_j^g \, x_j} \right\}. \tag{4.1}$$

In the problem above, the fraction computes the expected revenue obtained from a customer of type g as a function of the set of products that we offer, whereas the outer sum computes the expected revenue over all customer types. It is likely

that obtaining exact solutions to problem (4.1) is difficult. In particular, [2] show that the problem above is NP-complete. Motivated by this complexity result, we focus on obtaining an upper bound on the optimal expected revenue Z^*, given by the optimal objective value of problem (4.1).

A natural approach for obtaining an upper bound on the optimal expected revenue Z^* is to proceed under the assumption that we can offer different sets of products to different customer types, but use penalty parameters to penalize the absence or presence of the products in the assortments offered to different customer types. To pursue this reasoning, we use $\lambda = \{\lambda_j^g : j \in N,\ g \in G\} \in \Re^{|N|\times|G|}$ to denote a vector of penalty parameters. As a function of the penalty parameters, we define $\Pi^g(\lambda)$ as the optimal objective value of the problem

$$\Pi^g(\lambda) \ = \max_{x\in\{0,1\}^{|N|}} \left\{ \frac{\sum_{j\in N} r_j\, v_j^g\, x_j}{1+\sum_{j\in N} v_j^g\, x_j} - \sum_{j\in N} \lambda_j^g\, x_j \right\}. \tag{4.2}$$

The problem above finds a set of products to offer so as to maximize the expected profit obtained from a customer of type g, where we generate a revenue of r_j when we sell product j and incur a cost of λ_j^g when we offer product j. The next lemma shows that $\sum_{g\in G} \alpha^g\, \Pi^g(\lambda)$ provides an upper bound on the optimal expected revenue Z^*, as long as the penalty parameters take values in the set $\Lambda = \{\lambda \in \Re^{|N|\times|G|} : \sum_{g\in G} \alpha^g\, \lambda_j^g = 0 \ \ \forall\, j \in N\}$. The proof is rather simple, but we include the proof to explicitly show the necessity of imposing the condition $\lambda \in \Lambda$.

Lemma 18. *For any $\lambda \in \Lambda$, we have $\sum_{g\in G} \alpha^g\, \Pi^g(\lambda) \geq Z^*$.*

Proof. Letting x^* be an optimal solution to problem (4.1), we observe that x^* is a feasible, but not necessarily an optimal solution to problem (4.2), in which case,

we obtain

$$\sum_{g \in G} \alpha^g \, \Pi^g(\lambda) \geq \sum_{g \in G} \alpha^g \left\{ \frac{\sum_{j \in N} r_j \, v_j^g \, x_j^*}{1 + \sum_{j \in N} v_j^g \, x_j^*} - \sum_{j \in N} \lambda_j^g \, x_j^* \right\}$$
$$= \sum_{g \in G} \alpha^g \frac{\sum_{j \in N} r_j \, v_j^g \, x_j^*}{1 + \sum_{j \in N} v_j^g \, x_j^*} - \sum_{j \in N} \left\{ \sum_{g \in G} \alpha^g \, \lambda_j^g \right\} x_j^* = Z^*,$$

where the last equality follows from the definition of x^* and the fact that the penalty parameters satisfy $\lambda \in \Lambda$ so that we have $\sum_{g \in G} \alpha^g \, \lambda_j^g = 0$ for all $j \in N$. $\square$

The penalty parameters can be positive or negative, where a positive value for λ_j^g discourages offering product j to a customer of type g, whereas a negative value for λ_j^g encourages offering product j to a customer of type g. Since the zero vector $\bar{0} \in \Re^{|N| \times |G|}$ is in Λ, Lemma 18 implies that $\sum_{g \in G} \alpha^g \, \Pi^g(\bar{0})$ provides an upper bound on the optimal expected revenue Z^* and this upper bound corresponds to the one obtained by offering different sets of products to different customer types without using any penalty parameters. In general, using penalty parameters other than zero can potentially yield tighter upper bounds and our computational experiments indicate that the benefits from using penalty parameters other than zero can be substantial.

Noting that we can use $\sum_{g \in G} \alpha^g \, \Pi^g(\lambda)$ for any $\lambda \in \Lambda$ as an upper bound on the optimal expected revenue Z^*, we can try to solve the problem $\min_{\lambda \in \Lambda} \sum_{g \in G} \alpha^g \, \Pi^g(\lambda)$ to obtain the tightest possible upper bound, but solving the last optimization problem is intractable. In particular, computing $\Pi^g(\lambda)$ at any λ requires solving problem (4.2). Problem (4.2) maximizes the expected profit from a customer of type g, where we generate a revenue from each product we sell and incur a cost for each product we offer. [18] show that such an assortment optimization problem that involves costs for offering the products is NP-complete. To overcome this difficulty, we develop an approximation $\Pi^g(\lambda)$, while maintaining the upper bound

provided by Lemma 18.

4.4 Upper Bound on Optimal Expected Revenue

At the end of the previous section, we propose solving the problem $\min_{\lambda \in \Lambda} \sum_{g \in G} \alpha^g \, \Pi^g(\lambda)$ to obtain the tightest possible upper bound on the optimal expected revenue Z^*, but solving this optimization problem turns out to be intractable. In this section, we develop an approximation $\tilde{\Pi}^g(\cdot)$ to $\Pi^g(\cdot)$. This approximation is tractable to compute and it satisfies $\tilde{\Pi}^g(\lambda) \geq \Pi^g(\lambda)$ for all $\lambda \in \Lambda$. In this case, by Lemma 18, we have $\sum_{g \in G} \alpha^g \, \tilde{\Pi}^g(\lambda) \geq \sum_{g \in G} \alpha^g \, \Pi^g(\lambda) \geq Z^*$ for any $\lambda \in \Lambda$, implying that we can use $\sum_{g \in G} \alpha^g \, \tilde{\Pi}^g(\lambda)$ for any $\lambda \in \Lambda$ as an upper bound on the optimal expected revenue Z^*. In this case, we can solve the problem $\min_{\lambda \in \Lambda} \sum_{g \in G} \alpha^g \, \tilde{\Pi}^g(\lambda)$ to obtain the tightest possible upper bound on the optimal expected revenue provided by the approximations $\{\tilde{\Pi}^g(\cdot) : g \in G\}$. Solving problem $\min_{\lambda \in \Lambda} \sum_{g \in G} \alpha^g \, \tilde{\Pi}^g(\lambda)$ turns out to be tractable. To develop an approximation to $\Pi^g(\lambda)$, we note that $1/(1 + \sum_{j \in N} v_j^g \, x_j)$ in problem (4.2) is the probability that a customer of type g does not purchase anything when the set of offered products is captured by the vector x. We fix the value of this no purchase probability at p and solve the problem

$$\max_{x \in \{0,1\}^{|N|}} \left\{ \sum_{j \in N} p \; r_j \, v_j^g \, x_j - \sum_{j \in N} \lambda_j^g \, x_j \;\; : \;\; \frac{1}{1 + \sum_{j \in N} v_j^g \, x_j} = p \right\}$$

for a fixed value of p. In this case, it follows that if we solve the problem above for all values of p in the interval $[0, 1]$ and pick the largest optimal objective value over all values of p, then we obtain the optimal objective value $\Pi^g(\lambda)$ of problem (4.2). We make a few refinements in this approach. Since the smallest possible value of the no purchase probability for any customer type is $p_{\min} = \min_{g \in G}\{1/(1 + \sum_{j \in N} v_j^g)\}$,

we can consider all possible values of p in the interval $[p_{\min}, 1]$, rather than $[0, 1]$. Furthermore, we can replace the equality constraint in the problem above with the corresponding greater than or equal to constraint $1/(1 + \sum_{j \in N} v_j^g \, x_j) \geq p$, since after replacing the equality constraint with the greater than or equal to constraint, if the constraint ends up being loose for any value of p, then we can increase the value of p until we make the constraint tight, which would only increase the objective value of the problem. So, since we want to find the value of p that makes the objective value of the problem above as large as possible, the values of p that render the constraint loose are not relevant to us. Thus, writing the objective function of the problem above as $\sum_{j \in N} (p \; r_j \, v_j^g - \lambda_j^g) \, x_j$ and noting that the constraint $1/(1 + \sum_{j \in N} v_j^g \, x_j) \geq p$ is equivalent to $\sum_{j \in N} v_j^g \, x_j \leq 1/p - 1$, the discussion above implies that if we solve the problem

$$\max_{x \in \{0,1\}^{|N|}} \left\{ \sum_{j \in N} (p \; r_j \, v_j^g - \lambda_j^g) \, x_j \;\; : \;\; \sum_{j \in N} v_j^g \, x_j \leq \frac{1}{p} - 1 \right\} \tag{4.3}$$

for all values of p in the interval $[p_{\min}, 1]$ and pick the largest optimal objective value over all values of p, then we obtain the optimal objective value $\Pi^g(\lambda)$ of problem (4.2). To develop an approximation to $\Pi^g(\lambda)$, we focus on the values of p over a set of grid points, while ensuring that our approximation is an upper bound on $\Pi^g(\lambda)$ even though we focus only on the grid points.

To develop an approximation to $\Pi^g(\lambda)$, consider problem (4.3) for some $p \in [p_{\min}, 1]$. If we replace the value of p in the objective function with a larger value and the value of p in the constraint with a smaller value, then the optimal objective value of problem (4.3) gets larger. For any $p_L, p_U \in [p_{\min}, 1]$ with $p_L \leq p_U$, we define $\Pi^g(\lambda, p_L, p_U)$ as the optimal objective value of the problem

$$\Pi^g(\lambda, p_L, p_U) = \max_{x \in [0,1]^{|N|}} \left\{ \sum_{j \in N} (p_U \, r_j \, v_j^g - \lambda_j^g) \, x_j \;\; : \;\; \sum_{j \in N} v_j^g \, x_j \leq \frac{1}{p_L} - 1 \right\}. \tag{4.4}$$

We observe that problem (4.4) is a continuous knapsack problem, where the capacity of the knapsack is $1/p_L - 1$, the utility of item j is $p_U\, r_j\, v_j^g - \lambda_j^g$ and the space consumption of item j is v_j^g. As mentioned above, for any $p \in [p_L, p_U]$, comparing problems (4.3) and (4.4), we observe that the objective function coefficients and the right side of the constraint in problem (4.4) are larger than those in problem (4.3). Furthermore, problem (4.4) does not impose integrality constraints on the decision variables. Thus, the optimal objective value of problem (4.4) is larger than that of problem (4.3). To develop an approximation on $\Pi^g(\lambda)$ while making sure that our approximation is an upper bound on $\Pi^g(\lambda)$, we consider an arbitrary set of grid points $\{p^k : k = 1, \ldots, K+1\}$ that satisfy $p_{\min} = p^1 \leq p^2 \leq \ldots \leq p^K \leq p^{K+1} = 1$. Focusing only on this set of grid points, we solve problem (4.4) for all values of p_L, p_U with $p_L = p^k$ and $p_U = p^{k+1}$ for all $k = 1, \ldots, K$. The next proposition shows that picking the largest optimal objective value of problem (4.4) over all values of p_L, p_U provides an upper bound on $\Pi^g(\lambda)$.

Proposition 19. *For any $\lambda \in \Lambda$, we have*

$$\max_{k \in \{1,\ldots,K\}} \left\{ \Pi^g(\lambda, p^k, p^{k+1}) \right\} \geq \Pi^g(\lambda).$$

Proof. We fix some $\lambda \in \Lambda$. We show that there exists $k \in \{1, \ldots, K\}$ such that $\Pi^g(\lambda, p^k, p^{k+1}) \geq \Pi^g(\lambda)$ and this inequality establishes the desired result. Letting x^* be an optimal solution to problem (4.2), we define p^* as $p^* = 1/(1 + \sum_{j \in N} v_j^g\, x_j^*)$ and choose k such that $p^* \in [p^k, p^{k+1}]$. Since $p^* \geq p^k$, we have $\sum_{j \in N} v_j^g\, x_j^* = 1/p^* - 1 \leq 1/p^k - 1$, which implies that the solution x^* is feasible to problem (4.4), when this problem is solved with $p_L = p^k$ and $p_U = p^{k+1}$. Thus, using the fact that $p^* \leq p^{k+1}$, we obtain $\Pi^g(\lambda) = \sum_{j \in N} p^*\, r_j\, v_j^g\, x_j^* - \sum_{j \in N} \lambda_j^g\, x_j^* \leq \sum_{j \in N} (p^{k+1}\, r_j\, v_j^g - \lambda_j^g)\, x_j^* \leq \Pi^g(\lambda, p^k, p^{k+1})$, where the first inequality is by $p^* \leq p^{k+1}$

and the second inequality is by the fact that the solution x^* is feasible to problem (4.4) when solved with $p_L = p^k$ and $p_U = p^{k+1}$. $\square$

Proposition 19 implies that if we let $\tilde{\Pi}^g(\lambda) = \max_{k\in\{1,\ldots,K\}}\{\Pi^g(\lambda, p^k, p^{k+1})\}$ and use $\tilde{\Pi}^g(\lambda)$ as an approximation to $\Pi^g(\lambda)$, then this approximation is an upper bound on $\Pi^g(\lambda)$. We note that Proposition 19 holds for any set of grid points $\{p^k : k = 1, \ldots, K+1\}$. In other words, we have $\max_{k\in\{1,\ldots,K\}}\{\Pi^g(\lambda, p^k, p^{k+1})\} \geq \Pi^g(\lambda)$ irrespective of the placement and number of grid points. Also, we observe that computing $\tilde{\Pi}^g(\lambda)$ for any $\lambda \in \Lambda$ requires solving K continuous knapsack problems. Each knapsack problem can be solved by ordering the items according to their utility to space consumption ratios and filling the knapsack starting from the item with the largest utility to space consumption ratio. Therefore, we can compute $\max_{k\in\{1,\ldots,K\}}\{\Pi^g(\lambda, p^k, p^{k+1})\}$ for any $\lambda \in \Lambda$ quickly as long as the number of grid points is not too large. In Section 4.6, we dwell on the question of how to choose a reasonable set of grid points.

There are two sources of error when we use $\tilde{\Pi}^g(\lambda) = \max_{k\in\{1,\ldots,K\}}\{\Pi^g(\lambda, p^k, p^{k+1})\}$ as an approximation to $\Pi^g(\lambda)$. First, the approximation $\tilde{\Pi}^g(\lambda)$ is obtained by solving problem (4.4) by using the set of grid points $\{p^k : k = 1, \ldots, K+1\}$, whereas $\Pi^g(\lambda)$ is obtained by solving problem (4.3) for all $p \in [p_{\min}, 1]$. Intuitively speaking, if the set of grid points is dense, then we expect the discrepancy due to focusing only on the grid points not to be large. This observation also indicates that by choosing a denser set of grid points, we can obtain better approximations to $\Pi^g(\lambda)$. Second, problem (4.3) imposes integrality constraints on the decision variables, whereas problem (4.4) does not. Our expectation is that the continuous relaxation of a knapsack problem provides good approximations to the original one and the discrepancy due to relaxing the integrality constraints is not large.

It is indeed possible to formulate a continuous knapsack problem whose optimal objective value deviates from the original binary one at most by a factor of two, but the deviation tends to be much less in practice; see [48].

4.5 Choosing Penalty Parameters

At the end of Section 4.3, we propose solving the problem $\min_{\lambda \in \Lambda} \sum_{g \in G} \alpha^g \Pi^g(\lambda)$ to obtain an upper bound on the optimal expected revenue Z^*, but solving this optimization problem is intractable. To overcome this difficulty, we propose using $\max_{k \in \{1,\ldots,K\}} \Pi^g(\lambda, p^k, p^{k+1})$ as an approximation to $\Pi^g(\lambda)$ and solving the problem

$$\min_{\lambda \in \Lambda} \left\{ \sum_{g \in G} \alpha^g \max_{k \in \{1,\ldots,K\}} \left\{ \Pi^g(\lambda, p^k, p^{k+1}) \right\} \right\}. \tag{4.5}$$

Noting that $\max_{k \in \{1,\ldots,K\}} \{\Pi^g(\lambda, p^k, p^{k+1})\} \geq \Pi^g(\lambda)$ for any $\lambda \in \Lambda$ by Proposition 19 and $\min_{\lambda \in \Lambda} \sum_{g \in G} \alpha^g \Pi^g(\lambda) \geq Z^*$ by Lemma 18, it follows that the optimal objective value of problem (4.5) provides an upper bound on the optimal expected revenue Z^*. Also, it is worthwhile to note that our notation in problem (4.5) suggests that the sets of grid points $\{p^k : k = 1, \ldots, K+1\}$ that we use for different customer types are the same, but it does not have to be the case and we can use different sets of grid points for different customer types. In this section, we show that $\max_{k \in \{1,\ldots,K\}} \Pi^g(\lambda, p^k, p^{k+1})$ is a convex function of λ, in which case, the objective function of the minimization problem in (4.5) is convex. Furthermore, we show how to obtain subgradients of $\max_{k \in \{1,\ldots,K\}} \Pi^g(\cdot, p^k, p^{k+1})$. Since the condition $\lambda \in \Lambda$ enforces a set of linear constraints on the penalty parameters, these results indicate that we can solve problem (4.5) by using subgradient search for minimizing a convex function subject to linear constraints; see [36].

It is not difficult to see that $\max_{k\in\{1,\ldots,K\}} \Pi^g(\lambda, p^k, p^{k+1})$ is convex in λ. When we view $\Pi^g(\lambda, p^k, p^{k+1})$ as a function of λ, it corresponds to the optimal objective value of the linear program in (4.4) as a function of its objective function coefficients. Thus, it follows from linear programming theory that $\Pi^g(\lambda, p^k, p^{k+1})$ is convex in λ. Since the pointwise maximum of convex functions is also convex, it follows that $\max_{k\in\{1,\ldots,K\}} \Pi^g(\lambda, p^k, p^{k+1})$ is convex in λ, as desired.

To show how to obtain subgradients of $\max_{k\in\{1,\ldots,K\}} \Pi^g(\cdot, p^k, p^{k+1})$, we let $\tilde{\Pi}^g(\lambda) = \max_{k\in\{1,\ldots,K\}} \Pi^g(\lambda, p^k, p^{k+1})$. To compute a subgradient of $\tilde{\Pi}^g(\cdot)$ at some $\hat{\lambda} \in \Re^{|N|\times|G|}$, we solve problem (4.4) with $\lambda = \hat{\lambda}$ and $p_L = p^k$, $p_U = p^{k+1}$ for all $k = 1, \ldots, K$. We let $k^* \in \{1, \ldots, K\}$ be such that we obtain the largest optimal objective value for problem (4.4) when we solve this problem with $p_L = p^{k^*}$ and $p_U = p^{k^*+1}$. In other words, we have $\tilde{\Pi}^g(\hat{\lambda}) = \Pi^g(\hat{\lambda}, p^{k^*}, p^{k^*+1})$. Furthermore, we let x^* be an optimal solution to problem (4.4) when we solve this problem with $\lambda = \hat{\lambda}$, $p_L = p^{k^*}$ and $p_U = p^{k^*+1}$, in which case, we get $\sum_{j\in N}(p^{k^*+1}\, r_j\, v_j^g - \hat{\lambda}_j^g)\, x_j^* = \Pi^g(\hat{\lambda}, p^{k^*}, p^{k^*+1}) = \tilde{\Pi}^g(\hat{\lambda})$ as well. On the other hand, at any arbitrary λ, we have $\tilde{\Pi}^g(\lambda) \geq \Pi^g(\lambda, p^{k^*}, p^{k^*+1})$ by the definition of $\tilde{\Pi}^g(\cdot)$. Also, when we solve problem (4.4) with an arbitrary value of λ but with $p_L = p^{k^*}$ and $p_U = p^{k^*+1}$, the solution x^* is feasible but not necessarily optimal to problem (4.4) and we obtain $\sum_{j\in N}(p^{k^*+1}\, r_j\, v_j^g - \lambda_j^g)\, x_j^* \leq \Pi^g(\lambda, p^{k^*}, p^{k^*+1}) \leq \tilde{\Pi}^g(\lambda)$. If we subtract this chain of inequalities from the equality $\sum_{j\in N}(p^{k^*+1}\, r_j\, v_j^g - \hat{\lambda}_j^g)\, x_j^* = \tilde{\Pi}^g(\hat{\lambda})$ obtained above, then we get

$$\tilde{\Pi}^g(\lambda) \geq \tilde{\Pi}^g(\hat{\lambda}) - \sum_{j\in N} x_j^*\, (\lambda_j^g - \hat{\lambda}_j^g).$$

The expression above indicates that $\tilde{\Pi}^g(\cdot)$ satisfies the subgradient inequality at the point $\hat{\lambda}$ with the subgradient $D(\hat{\lambda}) = \{D_j^c(\hat{\lambda}) : j \in N,\ c \in G\} \in \Re^{|N|\times|G|}$ given by $D_j^c(\hat{\lambda}) = -x_j^*$ if $c = g$ and $D_j^c(\hat{\lambda}) = 0$ if $c \in G \setminus \{g\}$. To sum up, if we want to

compute a subgradient of $\tilde{\Pi}^g(\cdot)$ at the point $\hat{\lambda}$, then we solve problem (4.4) with $\lambda = \hat{\lambda}$ and $p_L = p^k$, $p_U = p^{k+1}$ for all $k = 1, \ldots, K$. We let $k^* \in \{1, \ldots, K\}$ be such that we obtain the largest optimal objective value for problem (4.4) when we solve this problem with $p_L = p^{k^*}$ and $p_U = p^{k^*+1}$. Finally, using x^* to denote an optimal solution to problem (4.4) when this problem is solved with $\lambda = \hat{\lambda}$, $p_L = p^{k^*}$ and $p_U = p^{k^*+1}$, $D(\hat{\lambda})$ as defined above provides a subgradient of $\tilde{\Pi}^g(\cdot)$ at $\hat{\lambda}$.

4.6 Effective Grid Points

The optimal objective value of problem (4.5) provides an upper bound on the optimal expected revenue Z^* for any choice of the grid points $\{p^k : k = 1, \ldots, K+1\}$. By the discussion that follows Proposition 19, we can obtain tighter upper bounds by using a denser set of grid points, but the computational effort to solve problem (4.5) increases with a denser set of grid points. To provide some guideline into the choice of the grid points, in this section, we explore the performance of exponential grid points. In particular, for fixed $\rho > 0$, we focus on the set of exponential grid points $\{(1+\rho)^{-k+1} : k = 1, \ldots, K+1\}$, where K is large enough that $(1+\rho)^{-K} \leq p_{\min} < (1+\rho)^{-K+1}$, in which case, these grid points cover the interval $[p_{\min}, 1]$.

In this section, we show that if we compute an upper bound on the optimal expected revenue Z^* by using the set of exponential grid points $\{(1+\rho)^{-k+1} : k = 1, \ldots, K+1\}$ in problem (4.5), then no other set of grid points, irrespective of how dense the set of grid points is, can improve this upper bound by more than a factor of $1+\rho$. In other words, if we use $\bar{Z}_\rho^{\mathsf{exp}}$ to denote the optimal objective value of problem (4.5) when we use the set of exponential grid points

$\{(1+\rho)^{-k+1} : k = 1, \ldots, K+1\}$ in this problem and $\bar{Z}^{\mathsf{arb}}$ to denote the optimal objective value of problem (4.5) with any arbitrary set of grid points, then it always holds that $\bar{Z}^{\mathsf{exp}}_{\rho} \leq (1+\rho)\, \bar{Z}^{\mathsf{arb}}$. Therefore, when we use the set of exponential grid points to obtain an upper bound, we can a priori be sure that it is not possible to improve this upper bound by more than a factor of $1+\rho$ by using a denser set of grid points. This result, in a sense, gives a performance guarantee for the set of exponential grid points. Furthermore, since the set of exponential grid points is denser to the left side of the interval $[p_{\min}, 1]$ and less dense to the right, this result builds the intuition that it is beneficial to use denser grid points when approximating smaller values of the no purchase probability. The next proposition becomes useful when showing the effectiveness of exponential grid points.

Proposition 20. *Let $\{(1+\rho)^{-k+1} : k = 1, \ldots, K+1\}$ be a set of exponential grid points for some $\rho > 0$ and $\{p^l : l = 1, \ldots, L+1\}$ be an arbitrary set of grid points with $p^1 \leq p^2 \leq \ldots \leq p^{L+1}$, both covering the interval $[p_{\min}, 1]$. For any $g \in G$ and $\lambda \in \Lambda$, we have*

$$\max_{k \in \{1,\ldots,K\}} \left\{ \Pi^g((1+\rho)\,\lambda, (1+\rho)^{-k}, (1+\rho)^{-k+1}) \right\} \leq (1+\rho) \max_{l \in \{1,\ldots,L\}} \left\{ \Pi^g(\lambda, p^l, p^{l+1}) \right\}. \tag{4.6}$$

Proof. We let $k^* \in \{1, \ldots, K\}$ be the value of k that attains the maximum on the left side of the inequality in (4.6). Also, we let $l^* \in \{1, \ldots, L\}$ be such that $p^{l^*} \leq (1+\rho)^{-k^*} < p^{l^*+1}$. Finally, we let x^* be an optimal solution to problem (4.4) when we solve this problem after replacing λ with $(1+\rho)\,\lambda$ and with $p_L = (1+\rho)^{-k^*}$, $p_U = (1+\rho)^{-k^*+1}$. Since $p^{l^*} \leq (1+\rho)^{-k^*}$, we have $1/p^{l^*} - 1 \geq 1/(1+\rho)^{-k^*} - 1$, implying that x^* is a feasible solution to problem (4.4) when we solve this problem

with $p_L = p^{l^*}$, $p_U = p^{l^*+1}$. Using the definition of x^*, we have

$$\Pi^g((1+\rho)\,\lambda, (1+\rho)^{-k^*}, (1+\rho)^{-k^*+1}) = \sum_{j\in N}((1+\rho)^{-k^*+1}\, r_j\, v_j^g - (1+\rho)\,\lambda_j^g)\, x_j^*$$

$$\leq (1+\rho)\sum_{j\in N}(p^{l^*+1}\, r_j\, v_j^g - \lambda_j^g)\, x_j^* \leq (1+\rho)\,\Pi^g(\lambda, p^{l^*}, p^{l^*+1})$$

$$\leq (1+\rho)\max_{l\in\{1,\ldots,L\}}\Big\{\Pi^g(\lambda, p^l, p^{l+1})\Big\},$$

where the first inequality follows by $(1+\rho)^{-k^*} < p^{l^*+1}$ and the second inequality holds since x^* is a feasible, but not necessarily an optimal solution to problem (4.4) when this problem is solved with $p_L = p^{l^*}$, $p_U = p^{l^*+1}$. By the definition of k^*, the first expression in the chain of inequalities above is equal to the expression on the left side of (4.6) and the desired result follows. □

The inequality in (4.6) holds for any $g \in G$ and $\lambda \in \Lambda$, in which case, multiplying this inequality by α^g, adding over all $g \in G$ and taking the minimum of both sides over all $\lambda \in \Lambda$, we get

$$\min_{\lambda\in\Lambda}\left\{\sum_{g\in G}\alpha^g \max_{k\in\{1,\ldots,K\}}\Big\{\Pi^g((1+\rho)\,\lambda, (1+\rho)^{-k}, (1+\rho)^{-k+1})\Big\}\right\} \leq$$

$$(1+\rho)\min_{\lambda\in\Lambda}\left\{\sum_{g\in G}\alpha^g \max_{l\in\{1,\ldots,L\}}\Big\{\Pi^g(\lambda, p^l, p^{l+1})\Big\}\right\}.$$

By the definition of Λ, we have $\lambda \in \Lambda$ if and only if $(1+\rho)\,\lambda \in \Lambda$. So, the constraint in the minimization problem on the left side above can be written as $(1+\rho)\,\lambda \in \Lambda$. Thus, replacing all occurrences of $(1+\rho)\,\lambda$ with λ through change of variables, we write the inequality above as

$$\min_{\lambda\in\Lambda}\left\{\sum_{g\in G}\alpha^g \max_{k\in\{1,\ldots,K\}}\Big\{\Pi^g(\lambda, (1+\rho)^{-k}, (1+\rho)^{-k+1})\Big\}\right\} \leq$$

$$(1+\rho)\min_{\lambda\in\Lambda}\left\{\sum_{g\in G}\alpha^g \max_{l\in\{1,\ldots,L\}}\Big\{\Pi^g(\lambda, p^l, p^{l+1})\Big\}\right\}.$$

We observe that the expression on the left side of the inequality above is the optimal objective value of problem (4.5) when we use the set of exponential grid points $\{(1+\rho)^{-k+1} : k = 1, \ldots, K+1\}$ in this problem, whereas the expression on the right side is the optimal objective value of problem (4.5) when we use an arbitrary set of grid points $\{p^l : l = 1, \ldots, L+1\}$. Therefore, the inequality above shows that the upper bound on the optimal expected revenue obtained by using an arbitrary set of grid points $\{p^l : l = 1, \ldots, L+1\}$ in problem (4.5) cannot improve the upper bound obtained by using the set of exponential grid points $\{(1+\rho)^{-k+1} : k = 1, \ldots, K+1\}$ by more than a factor of $1+\rho$, which is the desired result.

Since $p_{\min} < (1+\rho)^{-K+1}$, we have $K = O(\log(p_{\min})/\log(1+\rho))$. For example, if the no purchase probability of a customer type is no smaller than 0.01 when we offer all of the products, then we can set $p_{\min} = 0.01$. If we want a performance guarantee of 0.1% by using a set of exponential grid points, then we can choose $\rho = 0.001$, in which case, K comes out to be about 4600. When we use this set of exponential grid points, no other set of grid points can improve the upper bound provided by the optimal objective value of problem (4.5) by more than 0.1%.

4.7 Connection to Lagrangian Relaxation

Noting that $\Pi^g(\lambda)$ is the optimal objective value of problem (4.2), Lemma 18 indicates that if we have $\lambda \in \Lambda$, then $\sum_{g \in G} \alpha^g \Pi^g(\lambda)$ provides an upper bound on the optimal expected revenue Z^*. In this section, our goal is to show that this result can be motivated by using Lagrangian relaxation on an appropriate reformulation of problem (4.1). For this purpose, we define the decision variable x_j^g such that

$x_j^g = 1$ if we offer product j to a customer of type g, otherwise we have $x_j^g = 0$. In this case, we choose an arbitrary customer type ϕ and write problem (4.1) equivalently as

$$\begin{aligned} Z^* = \max \quad & \sum_{g \in G} \alpha^g \frac{\sum_{j \in N} r_j \, v_j^g \, x_j^g}{1 + \sum_{j \in N} v_j^g \, x_j^g} \\ \text{subject to} \quad & x_j^g = x_j^\phi \qquad \forall \, j \in N, \; g \in G \setminus \{\phi\} \\ & x_j^g \in \{0, 1\} \qquad \forall \, j \in N, \; g \in G. \end{aligned} \tag{4.7}$$

By the constraints above, we can replace the decision variables $\{x_j^g : g \in G\}$ with a single decision variable x_j^ϕ, in which case, the problem above becomes equivalent to problem (4.1). Relaxing the constraints in problem (4.7) by associating the Lagrange multipliers $\{\alpha^g \lambda_j^g : j \in N, \; g \in G \setminus \{\phi\}\}$ with them, the objective function of the problem above can be written as

$$\begin{aligned} & \sum_{g \in G \setminus \{\phi\}} \alpha^g \left\{ \frac{\sum_{j \in N} r_j \, v_j^g \, x_j^g}{1 + \sum_{j \in N} v_j^g \, x_j^g} - \sum_{j \in N} \lambda_j^g \, x_j^g \right\} \\ & + \alpha^\phi \left\{ \frac{\sum_{j \in N} r_j \, v_j^\phi \, x_j^\phi}{1 + \sum_{j \in N} v_j^\phi \, x_j^\phi} + \sum_{j \in N} \Big[\sum_{g \in G \setminus \{\phi\}} \frac{\alpha^g \, \lambda_j^g}{\alpha^\phi} \Big] \, x_j^\phi \right\}. \end{aligned} \tag{4.8}$$

We use $\{\alpha^g : g \in G \setminus \{\phi\}\}$ to scale the Lagrange multipliers $\{\alpha^g \lambda_j^g : j \in N, \; g \in G \setminus \{\phi\}\}$, as this scaling ultimately allows us to draw parallels with our earlier development more easily. This scaling is not a concern since if $\alpha^g = 0$ for some customer type g, then we can drop this customer type from consideration. If we define the additional Lagrange multipliers $\{\lambda_j^\phi : j \in N\}$ for the customer type ϕ as $\lambda_j^\phi = -\sum_{g \in G \setminus \{\phi\}} \alpha^g \, \lambda_j^g / \alpha^\phi$ for all $j \in N$, then the coefficient of the decision variable x_j^ϕ in the last sum in (4.8) is $-\lambda_j^\phi$. Also, noting that $\alpha^\phi \, \lambda_j^\phi = -\sum_{g \in G \setminus \{\phi\}} \alpha^g \, \lambda_j^g$, we have $\sum_{g \in G} \alpha^g \, \lambda_j^g = 0$ for all $j \in N$, which implies that $\lambda = \{\lambda_j^g : j \in N, \; g \in G\}$ satisfies $\lambda \in \Lambda$. In this case, noting that the coefficient of the decision variable x_j^ϕ

in the last sum in (4.8) is $-\lambda_j^{\phi}$, we can write (4.8) as

$$\sum_{g \in G} \alpha^g \left\{ \frac{\sum_{j \in N} r_j \, v_j^g \, x_j^g}{1 + \sum_{j \in N} v_j^g \, x_j^g} - \sum_{j \in N} \lambda_j^g \, x_j^g \right\}.$$

The discussion so far shows that relaxing the constraints in problem (4.7) by associating the Lagrange multipliers $\{\alpha^g \, \lambda_j^g : j \in N, \ g \in G \setminus \{\phi\}\}$ with them is equivalent to solving the problem

$$\begin{aligned} \max \quad & \sum_{g \in G} \alpha^g \left\{ \frac{\sum_{j \in N} r_j \, v_j^g \, x_j^g}{1 + \sum_{j \in N} v_j^g \, x_j^g} - \sum_{j \in N} \lambda_j^g \, x_j^g \right\} \\ \text{subject to} \quad & x_j^g \in \{0, 1\} \qquad \forall \, j \in N, \ g \in G, \end{aligned} \tag{4.9}$$

as long as $\lambda \in \Lambda$. Noting that problem (4.9) is obtained by relaxing the constraints in problem (4.7) by associating the Lagrange multipliers $\{\alpha^g \, \lambda_j^g : j \in N, \ g \in G \setminus \{\phi\}\}$ with them, it is straightforward to show that the optimal objective value of the problem above provides an upper bound on the optimal objective value of problem (4.7), which is Z^*. We observe that problem (4.9) decomposes by customer types and noting the definition of $\Pi^g(\lambda)$ in (4.2), the optimal objective value of problem (4.9) is $\sum_{g \in G} \alpha^g \, \Pi^g(\lambda)$. Therefore, it follows that $\sum_{g \in G} \alpha^g \, \Pi^g(\lambda)$ provides an upper bound on the optimal expected revenue Z^* as long as λ satisfies $\lambda \in \Lambda$. This result corresponds to the one given in Lemma 18, but as we show in this section, it is possible to reach this result by using Lagrangian relaxation on an appropriate reformulation of problem (4.1).

4.8 Extensions to Constrained Problems and Other Choice Models

In this section, we discuss two extensions of our approach. First, we consider the case where each product occupies a certain amount of space and the total space consumption of the offered products cannot exceed a certain space limit. Second, we consider the case where customers choose according to a mixture of NL models, rather than a MMNL model.

4.8.1 Space Constraint

To extend our approach to the case where there is a space constraint, we use w_j to denote the space consumption of product j. Letting c be the total amount of space available for the offered products, we want to solve the problem

$$Z^* = \max_{x \in \{0,1\}^{|N|}} \left\{ \sum_{g \in G} \alpha^g \frac{\sum_{j \in N} r_j \, v_j^g \, x_j}{1 + \sum_{j \in N} v_j^g \, x_j} \;:\; \sum_{j \in N} w_j \, x_j \leq c \right\}, \tag{4.10}$$

which maximizes the expected revenue obtained from a customer while making sure that the total space consumption of the offered products does not exceed c. Thus, this problem is the analogue of problem (4.1) under a space constraint. If we have $w_j = 1$ for all $j \in N$, then problem (4.10) simply limits the total number of offered products to c. Following the same argument in Section 4.3, we use the penalty parameters $\lambda = \{\lambda_j^g : j \in N,\ g \in G\} \in \Re^{|N| \times |G|}$ to penalize the absence or presence of the products in the assortments offered to different customer types

and define $\Pi^g(\lambda)$ as the optimal objective value of the problem

$$\Pi^g(\lambda) = \max_{x \in \{0,1\}^{|N|}} \left\{ \frac{\sum_{j \in N} r_j \, v_j^g \, x_j}{1 + \sum_{j \in N} v_j^g \, x_j} - \sum_{j \in N} \lambda_j^g \, x_j \; : \; \sum_{j \in N} w_j \, x_j \leq c \right\}. \tag{4.11}$$

The problem above is the analogue of problem (4.2). With the definitions of Z^* as in (4.10) and $\Pi^g(\lambda)$ as in (4.11), Lemma 18 continues to hold and we have $\sum_{g \in G} \alpha^g \, \Pi^g(\lambda) \geq Z^*$ for any $\lambda \in \Lambda$, where Λ is as defined in Section 4.3. As mentioned in Section 4.3, computing $\Pi^g(\lambda)$ at any λ requires solving an NP-complete problem. To approximate $\Pi^g(\lambda)$, we define $\Pi^g(\lambda, p_L, p_U)$ as

$$\Pi^g(\lambda, p_L, p_U) = \max_{x \in [0,1]^{|N|}} \left\{ \sum_{j \in N} (p_U \, r_j \, v_j^g - \lambda_j^g) \, x_j \; : \; \sum_{j \in N} v_j^g \, x_j \leq \frac{1}{p_L} - 1, \; \sum_{j \in N} w_j \, x_j \leq c \right\}, \tag{4.12}$$

which is the analogue of problem (4.4) under a space constraint. The problem above is a continuous knapsack problem with two dimensions and [38] discuss efficient solution approaches for such knapsack problems. Letting $p_{\min} = \min_{g \in G} \{1/(1 + \sum_{j \in N} v_j^g)\}$ and using $\{p^k : k = 1, \ldots, K+1\}$ to denote a set of grid points that satisfy $p_{\min} = p^1 \leq p^2 \leq \ldots \leq p^K \leq p^{K+1} = 1$, Proposition 19 continues to hold with the definitions of $\Pi^g(\lambda)$ as in (4.11) and $\Pi^g(\lambda, p_L, p_U)$ as in (4.12). In this case, we can solve problem (4.5) to obtain the tightest possible upper bound on the optimal expected revenue. Using the same approach in Section 4.5, we can show that the objective function of problem (4.5) is convex in λ with the definition of $\Pi^g(\lambda, p_L, p_U)$ as in (4.12) and we can efficiently obtain subgradients of the objective function of this problem, which indicates that problem (4.5) continues to be tractable under a space constraint. Therefore, the approach that we propose to compute upper bounds on the optimal expected revenue remains applicable when we have a constraint that limits the total space consumption of the offered products. The main difference is that we need to work with the con-

tinuous knapsack problem with two dimensions given in (4.12), instead of working with the continuous knapsack problem given in (4.4).

4.8.2 Mixture of Nested Logit Models

Under the NL model, the products are grouped into nests so that the products in the same nest are closer substitutes of each other when compared with the products in different nests. Given the nest structure, the choice process of a customer under the NL model proceeds in two stages. First, the customer either chooses one of the nests or decides to leave without a purchase. Second, if the customer chooses a nest, then the customer purchases one of the products offered in this nest. In this section, we extend our approach to the case where customers choose according to a mixture of NL models, as long as the number of products in each nest is reasonably small, but the number of nests can be large. To formulate the NL model, we use M to denote the set of nests and N to denote the set of products in each nest. Therefore, the total number of products is $|M| \times |N|$. The set of customer types is G and a customer of type g arrives with probability α^g. We use $S_i \subset N$ to denote the set of products that we offer in nest i. Therefore, the set of products offered in all nests are given by $\{S_i : i \in M\}$. A customer of type g associates the preference weight v_{ij}^g with product j in nest i. As a function of S_i, we use $V_i^g(S_i) = \sum_{j \in S_i} v_{ij}^g$ to denote the total preference weight of the products offered in nest i for a customer of type g. Under the NL model, if a customer of type g has already decided to make a purchase in nest i and the set S_i of products are offered in this nest, then the customer purchases product $j \in S_i$ with probability $v_{ij}^g / V_i^g(S_i)$. Thus, using r_{ij} to denote the revenue associated with product j in nest i, if a customer of type g has already decided to make a purchase in nest i and the

set S_i of products are offered in this nest, then we obtain an expected revenue of

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

from this customer. A customer of type g associates the dissimilarity parameter γ_i^g with nest i. In particular, the parameter γ_i^g measures how well the products in nest i substitute for each other for a customer of type g; see [25] and [42]. Under the NL model, if the set of products offered in all nests are given by $\{S_i : i \in M\}$, then a customer of type g decides to make a purchase in nest i with probability

$$\frac{(V_i^g(S_i))^{\gamma_i^g}}{1 + \sum_{l \in M} (V_l^g(S_l))^{\gamma_l^g}},$$

where we normalize the preference weight of the no purchase option to one. Thus, since $R_i^g(S_i)$ is the expected revenue from a customer of type g that has already decided to make a purchase in nest i, if the set of products offered over all nests are given by $\{S_i : i \in M\}$, then the expected revenue obtained from a customer of type g is $\sum_{i \in M} R_i^g(S_i) \, (V_i^g(S_i))^{\gamma_i^g} / (1 + \sum_{i \in M} (V_i^g(S_i))^{\gamma_i^g})$. In this case, we can solve the problem

$$Z^* = \max_{\substack{\{S_i \, : \, i \in M\} \, : \\ S_i \subset N \; \forall \, i \in M}} \left\{ \sum_{g \in G} \alpha^g \, \frac{\sum_{i \in M} R_i^g(S_i) \, (V_i^g(S_i))^{\gamma_i^g}}{1 + \sum_{i \in M} (V_i^g(S_i))^{\gamma_i^g}} \right\} \tag{4.13}$$

to find the set of products to offer over all nests so as to maximize the expected revenue obtained from a customer. In the problem above, the fraction computes the expected revenue from a customer of type g, whereas the outer sum computes the expected revenue over all customer types. In our formulation, we assume that there are $|N|$ products in each nest, but it is straightforward to extend our formulation to the case where different nests have different numbers of products.

If we have $\gamma_i^g = 1$ for all $i \in M$, $g \in G$, then the NL model becomes equivalent to the multinomial logit model; see [42]. Thus, solving problem (4.13) is at least as

difficult as finding a set of products to offer that maximizes the expected revenue under a MMNL model. We focus on obtaining an upper bound on the optimal expected revenue Z^* in problem (4.13). Our approach for obtaining such an upper bound exploits the assumption that the number of products in each nest is reasonably small. The starting point for our approach is an appropriate reformulation of problem (4.13). To give this reformulation, we define the decision variable $x_i(S_i)$ such that $x_i(S_i) = 1$ is we offer the set S_i of products in nest i, otherwise we have $x_i(S_i) = 0$. In this case, using $x = \{x_i(S_i) : i \in M,\ S_i \subset N\} \in \{0,1\}^{|M|\times 2^{|N|}}$ to capture the sets of products offered in different nests and letting $\nu_i^g(S_i) = (V_i(S_i))^{\gamma_i^g}$ for notational brevity, we observe that problem (4.13) is equivalent to the problem

$$Z^* = \max_{x\in\{0,1\}^{|M|\times 2^{|N|}}} \left\{ \sum_{g\in G} \alpha^g \frac{\sum_{i\in M}\sum_{S_i\subset N} R_i^g(S_i)\,\nu_i^g(S_i)\,x_i(S_i)}{1+\sum_{i\in M}\sum_{S_i\subset N}\nu_i^g(S_i)\,x_i(S_i)} \ : \right.$$
$$\left. \sum_{S_i\subset N} x_i(S_i) = 1 \ \forall\, i \in M \right\}, \tag{4.14}$$

where the decision variables $\{x_i(S_i) : S_i \subset N\}$ describe which set of products we offer in nest i and the constraints ensure that we offer exactly one set of products in each nest, but this set can be the empty set. In the problem above, we have one decision variable $x_i(S_i)$ for each nest i and for each set $S_i \subset N$. Thus, the number of decision variables is manageable when the number of products in each nest $|N|$ is reasonably small. The number of decision variables is manageable even when the number of nests $|M|$ is large. Redefining the set of products appropriately, it is possible to see that the objective function of problem (4.14) is similar to the expected revenue function when customers choose according to a MMNL model. In particular, we index the products by $\{(i, S_i) : i \in M,\ S_i \subset N\}$. If a customer of type g purchases product (i, S_i), then we generate a revenue of $R_i^g(S_i)$. Furthermore, a customer of type g associates a preference weight of $\nu_i^g(S_i)$ with product (i, S_i). For all $i \in M$, $S_i \subset N$ and $g \in G$, we can compute and store $R_i^g(S_i)$ and $\nu_i^g(S_i)$

so that $\{R_i^g(S_i) : i \in M,\ S_i \subset N,\ g \in G\}$ and $\{\nu_i^g(S_i) : i \in M,\ S_i \subset N,\ g \in G\}$ become constant parameters in problem (4.14). Thus, comparing the objective function of problem (4.14) with that of problem (4.1), the objective function of problem (4.14) is similar to the expected revenue function when customers choose according to a MMNL model.

Building on this similarity, we can use the approach in Section 4.3 to obtain an upper bound on the optimal expected revenue Z^* in problem (4.14). In particular, we use the penalty parameters $\lambda = \{\lambda_i^g(S_i) : i \in M,\ S_i \subset N,\ g \in G\} \in \Re^{|M|\times 2^{|N|}\times|G|}$, where $\lambda_i^g(S_i)$ penalizes offering or not offering product (i, S_i) to a customer of type g. As a function of the penalty parameters, we define $\Pi^g(\lambda)$ as the optimal objective value of the problem

$$\Pi^g(\lambda) = \max_{x\in\{0,1\}^{|M|\times 2^{|N|}}} \left\{ \frac{\sum_{i\in M}\sum_{S_i\subset N} R_i^g(S_i)\,\nu_i^g(S_i)\,x_i(S_i)}{1+\sum_{i\in M}\sum_{S_i\subset N}\nu_i^g(S_i)\,x_i(S_i)} - \sum_{i\in M}\sum_{S_i\subset N}\lambda_i^g(S_i)\,x_i(S_i) : \sum_{S_i\subset N} x_i(S_i) = 1\ \forall\, i\in M \right\}, \tag{4.15}$$

which is the analogue of problem (4.2) under a mixture of NL models. We define the set of penalty parameters $\Lambda = \{\lambda \in \Re^{|M|\times 2^{|N|}\times|G|} : \sum_{g\in G}\alpha^g\,\lambda_i^g(S_i) = 0\ \forall\, i \in M,\ S_i \subset N\}$. With this definition of Λ and the definitions of Z^* as in (4.14) and $\Pi^g(\lambda)$ as in (4.15), it is possible to check that Lemma 18 continues to hold and we have $\sum_{g\in G}\alpha^g\,\Pi^g(\lambda) \geq Z^*$ for any $\lambda \in \Lambda$. Due to the binary decision variables and the nonlinear objective function in problem (4.15), computing $\Pi^g(\lambda)$ for a particular value of λ can be difficult. We get around this difficulty by using an approximation to $\Pi^g(\lambda)$. For our approximation of $\Pi^g(\lambda)$, we define $\Pi^g(\lambda, p_L, p_U)$

as

$$\Pi^g(\lambda, p_L, p_U) = \max_{x \in [0,1]^{|M| \times 2^{|N|}}} \left\{ \sum_{i \in M} \sum_{S_i \subset N} (p_U R_i^g(S_i)\, \nu_i^g(S_i) - \lambda_i^g(S_i))\, x_i(S_i) : \right.$$
$$\left. \sum_{i \in M} \sum_{S_i \subset N} \nu_i^g(S_i)\, x_i(S_i) \leq \frac{1}{p_L} - 1, \ \sum_{S_i \subset N} x_i(S_i) = 1 \ \forall\, i \in M \right\}, \tag{4.16}$$

which is the analogue of problem (4.4) under a mixture of NL models. The problem above is a continuous multiple choice knapsack problem; see [38]. In the objective function of problem (4.15), the smallest value of $1/(1+\sum_{i \in M} \sum_{S_i \subset N} \nu_i^g(S_i)\, x_i(S_i))$ in any feasible solution to this problem is $1/(1+\sum_{i \in M} \nu_i^g(N))$. Thus, letting $p_{\min} = \min_{g \in G}\{1/(1+\sum_{i \in M} \nu_i^g(N))\}$ and using $\{p^k : k = 1, \ldots, K+1\}$ to denote a set of grid points that satisfy $p_{\min} = p^1 \leq p^2 \leq \ldots \leq p^K \leq p^{K+1} = 1$, Proposition 19 continues to hold with the definitions of $\Pi^g(\lambda)$ as in (4.15) and $\Pi^g(\lambda, p_L, p_U)$ as in (4.16). In this case, we can solve problem (4.5) to obtain the tightest possible upper bound on the optimal expected revenue. We can use the same approach in Section 4.5 to show that the objective function of problem (4.5) is convex in λ with the definition of $\Pi^g(\lambda, p_L, p_u)$ as in (4.16). Also, we can use the same approach in Section 4.5 to obtain subgradients of the objective function of problem (4.5), which implies that problem (4.5) continues to be tractable under a mixture of NL models. These observations indicate that the approach that we propose to obtain upper bounds on the optimal expected revenue remains applicable under a mixture of NL models. The main difference is that we need to work with problem (4.16), instead of problem (4.4).

4.9 Computational Experiments

In this section, we provide computational experiments that test the quality of the upper bounds on the optimal expected revenue that we obtain by solving problem (4.5).

4.9.1 Benchmark Strategies

We compare the upper bounds provided by the following three benchmark strategies.

Penalty Multipliers (PM). This benchmark strategy corresponds to the upper bound provided by the optimal objective value of problem (4.5). The set of grid points that we use is of the form $\{(1+\rho)^{-k+1} : k = 1, \ldots, K+1\}$ with $K = O(\log(p_{\min})/\log(1+\rho))$. We use $\rho = 0.001$. To ensure that $\lambda \in \Lambda$, we choose an arbitrary customer type ϕ and assume that only the penalty multipliers $\{\lambda_j^g : j \in N,\ g \in G \setminus \{\phi\}\}$ are decision variables in problem (4.5). We solve the penalty parameters corresponding to customer type ϕ in terms of the other penalty parameters to obtain $\lambda_j^\phi = -\sum_{g \in G \setminus \{\phi\}} \alpha^g \lambda_j^g / \alpha^\phi$ for all $j \in N$. In this way, we ensure that $\lambda \in \Lambda$ without explicitly imposing this constraint. When implementing PM, we solve problem (4.5) by using subgradient search with the initial solution $\lambda = \bar{0}$. We use subgradient search for 100 iterations, where the step size at iteration t is of the form $1/t$. Our hope is that these 100 iterations get us into the vicinity of a reasonable solution. After these 100 iterations, we switch to another form for the step size, where we increase the step size by a factor of two after each iteration that yields an improvement in the objective value of problem (4.5), whereas we

decrease the step size by a factor of two after each iteration that does not yield an improvement. This form for the step size may not ensure convergence to an optimal solution to problem (4.5), but it provides consistently good performance in our experience. Since the optimal objective value of problem (4.5) provides an upper bound on the optimal expected revenue Z^* and this problem is a minimization problem, any feasible solution to problem (4.5) also provides an upper bound on the optimal expected revenue.

Customer Type Decomposition (CD). This benchmark strategy corresponds to the upper bound obtained under the assumption that we know the type of an arriving customer so that we can offer different sets of products to customers of different types. In particular, we can solve the problem $\max_{x \in \{0,1\}^{|N|}} \sum_{j \in N} P_j^g(x)\, r_j = \max_{x \in \{0,1\}^{|N|}} (\sum_{j \in N} r_j\, v_j^g\, x_j)/(1 + \sum_{j \in N} v_j^g\, x_j)$ to find a set of products that maximizes the expected revenue obtained from a customer of type g. [41] show that this problem, which involves a single customer type, can be solved efficiently. Thus, letting $\hat{Z}^g$ be the optimal objective value of this problem, the largest expected revenue that we can obtain from a customer of type g is given by $\hat{Z}^g$. The upper bound provided by CD is $\sum_{g \in G} \alpha^g\, \hat{Z}^g$, which corresponds to the optimal expected revenue that can be obtained under the assumption that we can offer different sets of products to customers of different types. Since problem (4.1) requires that we offer a single set of products to customers of all types, $\sum_{g \in G} \alpha^g\, \hat{Z}^g$ is an upper bound on the optimal objective value of problem (4.1). CD builds on [40], where the author shows that allowing a retailer to offer different sets to different customer types can provide good approximations in network revenue management problems.

Branch and Bound (BB). [2] give a mixed integer programming formulation for problem (4.1), but solving this mixed integer programming formulation to op-

timality can be time consuming for large problem instances. We apply branch and bound on the mixed integer programming formulation for a fixed amount of run time and check the best upper bound that branch and bound achieves on the optimal objective value of the mixed integer program. Therefore, the upper bound provided by BB corresponds to the best upper bound that we obtain by using branch and bound for a fixed amount of run time. We choose the run time for branch and bound as twice the run time for PM, so that we can compare the upper bounds obtained by PM and BB within comparable amounts of run time.

4.9.2 Experimental Setup

In our computational experiments, we generate a large number of problem instances and compare the upper bounds provided by PM, CD and BB for each problem instance. We use the following approach for generating our problem instances. Throughout our computational experiments, the number of products $|N|$ is 100. We vary the number of customer types $|G|$. To come up with the revenues, we simply sample r_j from the uniform distribution over $[0, 2000]$ for all $j \in N$. To come up with the probabilities $\{\alpha^g : g \in G\}$ of observing customers of different types, we sample β^g from the uniform distribution over $[0, 1]$ for all $g \in G$ and set $\alpha^g = \beta^g / \sum_{c \in G} \beta^c$.

To come up with the preference weights, we choose a set $S \subset N$ of products and designate them as specialty products. We refer to the remaining set of products as staple products. Throughout our computational experiments, the number of staple products $|S|$ is 40. Customers of different types can associate significantly different preference weights with a specialty product, indicating that the evaluations of a specialty product by customers of different types can be quite different. Customers

of different types evaluate a staple product more or less in the same fashion. To generate preference weights with these characteristics, for all $j \in N$, $g \in G$, we sample X_j^g as follows. If product j is a specialty product, then we sample X_j^g from the uniform distribution over $[0.1, 0.3] \cup [0.7, 0.9]$, whereas if product j is a staple product, then we sample X_j^g from the uniform distribution over $[0.3, 0.7]$. Thus, the variance of X_j^g is larger when product j is a specialty product. For all $j \in N$, we also sample κ_j from the uniform distribution over $[1, \bar{K}]$, where $\bar{K}$ is a parameter that we vary in our computational experiments. In this case, we set the preference weight v_j^g that a customer of type g associates with product j as a quantity that is proportional to $\kappa_j X_j^g$. In this setup, the value of κ_j determines an overall magnitude for the preference weights $\{v_j^g : g \in G\}$ associated with product j. Furthermore, if product j is a specialty product, then the variance of X_j^g is relatively large, in which case, the variance of $\kappa_j X_j^g$ is relatively large as well. So, if product j is a specialty product, then the preference weights $\{v_j^g : g \in G\}$ that customers of different types associate with product j display relatively large variability among themselves, which agrees with our expectation from a specialty product. Similarly, if product j is a staple product, then the variance of X_j^g is relatively small so that the preference weights $\{v_j^g : g \in G\}$ that customers of different types associate with product j display relatively small variability among themselves.

As mentioned above, we set the preference weight v_j^g that a customer of type g associates with product j as a quantity that is proportional to $\kappa_j X_j^g$. To come up with the values of the preference weights, we sample P_0^g from the uniform distribution over $[0, \bar{P}_0]$ for all $g \in G$, where $\bar{P}_0$ is a parameter that we vary in our computational experiments. In this case, we set the value of the preference weight v_j^g as $v_j^g = \kappa_j X_j^g (1 - P_0^g)/(P_0^g \sum_{i \in N} \kappa_i X_i^g)$. Noting that $\sum_{j \in N} v_j^g =$

$\sum_{j \in N} \kappa_j X_j^g (1 - P_0^g)/(P_0^g \sum_{i \in N} \kappa_i X_i^g) = (1 - P_0^g)/P_0^g$ in this setup, even if we offer all of the products to the customers, a customer of type g leaves without making a purchase with probability $1/(1 + \sum_{j \in N} v_j^g) = 1/(1 + (1 - P_0^g)/P_0^g) = P_0^g$. Therefore, if we use a larger value for $\bar{P}_0$, then customers are more likely to leave without making a purchase. Also, if we use a larger value for $\bar{P}_0$, then the variance of P_0^g gets larger and customers of different types tend to become more heterogeneous in terms of their tendency to leave without making a purchase.

In our computational experiments, we vary $|G|$, $\bar{K}$ and $\bar{P}_0$ over $|G| \in \{25, 50, 75\}$, $\bar{K} \in \{5, 10, 20\}$ and $\bar{P}_0 \in \{0.6, 0.8, 1.0\}$. This setup provides 27 parameter combinations. In each parameter combination, we generate 1000 individual problem instances by using the approach described above. For each problem instance, we compute the upper bounds on the optimal expected revenue provided by PM, CD and BB. To put these upper bounds into perspective, we also use a greedy heuristic to find a solution to problem (4.1). In the greedy heuristic, we start with a solution to problem (4.1) that does not include any products. Given the current solution, we try adding or removing each one of the products into or from this current solution. Among all of these options, we update the current solution by using the option that provides the largest improvement in the expected revenue from the current solution. If none of the options provides an improvement, then we stop. The expected revenue from the solution obtained by the greedy heuristic provides a lower bound on the optimal expected revenue. By checking the gap between the upper bound on the optimal expected revenue provided by PM, CD or BB and the expected revenue from the greedy heuristic, we can assess how PM, CD and BB compare with each other in terms of the tightness of their upper bounds and we can get a conservative estimate of how much the upper bounds provided by PM, CD and BB deviate from the optimal expected revenue.

4.9.3 Computational Results

We give our main computational results in Table 4.1. The first column in this table shows the parameter combinations for our test problems by using $(|G|, \bar{K}, \bar{P}_0)$. We recall that we generate 1000 problem instances in each parameter combination. For each problem instance k, we compute the expected revenue from the solution obtained by the greedy heuristic. We let GRR^k be this expected revenue. We use PM, CD and BB to compute upper bounds on the optimal expected revenue. We let PMU^k, CDU^k and BBU^k respectively be the upper bounds provided by PM, CD and BB for problem instance k. The second column in Table 4.1 shows the percent gap between the upper bounds from PM and the expected revenues from the greedy heuristic, averaged over all problem instances in a parameter combination. In other words, this column shows the average of the data points $\{100 \times (\mathsf{PMU}^k - \mathsf{GRR}^k)/\mathsf{PMU}^k : k = 1, \ldots, 1000\}$, which can be used to assess the average optimality gap of the greedy heuristic when we use the upper bounds from PM to check the quality of a solution. The third and the fourth columns respectively show the 95th percentile and maximum of the same data points $\{100 \times (\mathsf{PMU}^k - \mathsf{GRR}^k)/\mathsf{PMU}^k : k = 1, \ldots, 1000\}$. The interpretations of the fifth, sixth and seventh columns are similar to those of the previous three columns, but the fifth, sixth and seventh columns respectively show the average, 95th percentile and maximum of the percent gaps between $\{\mathsf{CDU}^k : k = 1, \ldots, 1000\}$ and $\{\mathsf{GRR}^k : k = 1, \ldots, 1000\}$, giving a feel for the optimality gaps of the greedy heuristic when we only use the upper bounds provided by CD to check the quality of a solution. Finally, the eighth, ninth and tenth columns respectively show the average, 95th percentile and maximum of the percent gaps between $\{\mathsf{BBU}^k : k = 1, \ldots, 1000\}$ and $\{\mathsf{GRR}^k : k = 1, \ldots, 1000\}$, which

Param. Comb. ($\|G\|$, $\bar{K}$, $\bar{P}_0$)	% Gap. of PMU^k with GRR^k			% Gap. of CDU^k with GRR^k			% Gap. of BBU^k with GRR^k		
	Avg.	95th	Max.	Avg.	95th	Max.	Avg.	95th	Max.
(25, 5, 0.6)	0.14	0.29	0.96	5.13	8.38	11.20	4.66	7.79	10.22
(25, 5, 0.8)	0.15	0.33	0.93	5.87	9.74	13.09	4.63	8.10	12.27
(25, 5, 1.0)	0.16	0.35	0.93	6.12	10.35	15.37	3.87	7.35	12.91
(25, 10, 0.6)	0.14	0.29	0.90	5.42	9.13	12.89	5.09	8.58	12.75
(25, 10, 0.8)	0.15	0.42	0.99	5.98	10.48	15.57	5.30	9.59	14.05
(25, 10, 1.0)	0.16	0.38	0.96	6.37	11.19	15.37	5.06	9.30	13.77
(25, 20, 0.6)	0.15	0.35	0.85	5.56	9.26	13.14	5.19	8.87	12.91
(25, 20, 0.8)	0.15	0.34	0.97	6.14	10.52	16.96	5.58	9.89	16.00
(25, 20, 1.0)	0.16	0.39	0.99	6.36	11.23	17.52	5.31	9.64	14.70
(50, 5, 0.6)	0.15	0.32	0.67	5.46	8.10	10.75	5.03	7.64	10.44
(50, 5, 0.8)	0.16	0.34	0.68	6.14	9.28	12.95	5.08	7.87	11.35
(50, 5, 1.0)	0.16	0.32	0.95	6.38	9.80	12.17	4.91	7.84	9.79
(50, 10, 0.6)	0.15	0.29	0.80	5.69	8.62	12.20	5.51	8.35	12.05
(50, 10, 0.8)	0.15	0.32	0.86	6.34	9.60	12.72	5.73	8.82	12.02
(50, 10, 1.0)	0.16	0.37	0.96	6.70	10.14	14.76	5.65	8.62	12.63
(50, 20, 0.6)	0.15	0.29	0.82	5.81	8.68	12.47	5.63	8.49	12.34
(50, 20, 0.8)	0.16	0.33	0.97	6.57	9.83	14.26	6.12	9.27	13.97
(50, 20, 1.0)	0.16	0.36	0.87	6.85	10.53	14.73	5.87	9.41	13.32
(75, 5, 0.6)	0.15	0.27	0.84	5.52	7.91	9.98	5.43	7.76	9.78
(75, 5, 0.8)	0.15	0.28	0.85	6.28	8.89	10.64	5.85	8.43	10.08
(75, 5, 1.0)	0.16	0.32	0.75	6.57	9.70	11.36	5.63	8.57	10.14
(75, 10, 0.6)	0.15	0.26	0.81	5.77	8.39	11.01	5.70	8.29	10.65
(75, 10, 0.8)	0.15	0.29	0.93	6.47	9.60	13.77	6.05	9.19	13.27
(75, 10, 1.0)	0.15	0.30	0.77	6.81	9.88	14.08	6.00	9.02	13.12
(75, 20, 0.6)	0.14	0.27	0.97	5.93	8.56	12.69	5.86	8.46	12.64
(75, 20, 0.8)	0.15	0.29	0.83	6.63	9.61	13.50	6.24	9.18	13.15
(75, 20, 1.0)	0.16	0.31	0.98	7.04	10.11	13.21	6.25	9.26	12.30
Average	0.15			6.15			5.45		

Table 4.1: Comparison of the upper bounds provided by PM, CD and BB.

indicate the optimality gaps of the greedy heuristic when we only use BB to obtain upper bounds on the optimal expected revenues.

The results in Table 4.1 indicate that the upper bounds provided by PM for our problem instances are quite tight. Over all of our problem instances, the average gap between the upper bounds from PM and the expected revenues from the greedy heuristic is 0.15%, whereas the maximum gap between the upper bounds from PM and the expected revenues from the greedy heuristic is 0.99%. The small gaps between the upper bounds from PM and the expected revenues from the

greedy heuristic demonstrate that the upper bounds provided by PM are within a fraction of a percent of the optimal expected revenues. Furthermore, if we use PM to obtain upper bounds on the optimal expected revenues, then we can establish that the greedy heuristic provides optimality gaps no larger than 0.99% for our problem instances. In contrast, the upper bounds provided by CD or BB can be substantially looser. The gap between the upper bound from CD and the expected revenue from the greedy heuristic can be as large as 17.52%. In other words, if we use the upper bounds from CD to evaluate the quality of a solution, then there are problem instances where we are left with the impression that the greedy heuristic may have optimality gaps as large as 17.52%, although we can use the upper bounds from PM to establish that the optimality gaps of the greedy heuristic are actually no larger than 0.99%. The upper bounds provided by BB improve those provided by CD slightly. The average and maximum gaps between the upper bounds from BB and the expected revenues from the greedy heuristic are respectively 5.45% and 16%. The same gaps are respectively 6.15% and 17.52% when we consider CD. Overall, our computational results for PM demonstrate that the upper bounds from PM are within 1% of the optimal expected revenues and the optimality gaps of the greedy heuristic are no larger than 1%. The last observation, together with the fact that the gap between the upper bound provided by BB and the expected revenue from the greedy heuristic can exceed 15%, indicates that the upper bound provided by BB can deviate from the optimal expected revenue by almost 14%. Naturally, the mixed integer programming formulation used by BB would eventually obtain the optimal expected revenue, but it turns out that this formulation is not effective when we want to obtain good upper bounds on the optimal expected revenues within a limited amount of run time. We shortly investigate the reasons why BB does not yield tight upper bounds.

It is useful to point out an interesting trend in Table 4.1. As $\bar{P}_0$ increases, there are larger gaps between the upper bound provided by CD and the expected revenue from the greedy heuristic. As mentioned when describing our experimental setup in Section 4.9.2, as $\bar{P}_0$ increases, customers of different types tend to become more heterogeneous in terms of their tendency to leave without making a purchase. As customers of different types become more heterogeneous, CD, which is based on the assumption that we can offer different sets of products to different customer types, ends up offering significantly different sets to different customer types. In this case, the upper bound from CD can deviate significantly from the optimal objective value of problem (4.1), which does not allow offering different sets of products to different customer types. In contrast, the gaps between the upper bound provided by PM and the expected revenue from the greedy heuristic remain quite stable as $\bar{P}_0$ increases.

The results in Table 4.1 show that the upper bounds provided by BB are not as tight, indicating that the mixed integer program used by BB is ineffective in obtaining good upper bounds within a limited amount of run time. One reason that BB is not able to obtain good upper bounds is that the linear programming relaxation of the mixed integer program used by BB turns out to be loose. In all of our test problems, the linear programming relaxation of the mixed integer program only slightly improves the upper bound from CD. To shed more light into this observation, Proposition 4 in Online Appendix C.1 shows that when we focus on each customer type individually, if customers of each type make a purchase with a probability that exceeds 1/2, then the optimal objective value of the linear programming relaxation of the mixed integer program used by BB precisely corresponds to the upper bound provided by CD. Thus, although it is tempting to try to obtain upper bounds on the optimal expected revenue by solving

the linear programming relaxation of the mixed integer program used by BB, this upper bound does not improve the one provided by CD when customers make a purchase with a reasonably large probability.

In Table 4.2, we give the details on the gaps between the upper bounds obtained by our benchmark strategies and the expected revenues from the greedy heuristic. The first column in this table shows the parameter combinations for our test problems. The second column shows the number of problem instances where the gap between the upper bound obtained by PM and the expected revenue from the greedy heuristic is less than 0.125%. The interpretations of the third, fourth, fifth, sixth and seventh columns are similar to that of the second column, but these columns show the numbers of problem instances where the gap between the upper bound obtained by PM and the expected revenue from the greedy heuristic is respectively less than 0.25%, 0.5%, 2.5%, 5% and 7.5%. The eighth to thirteenth columns have the same interpretations as the second to seventh columns, but they focus on the gap between the upper bound obtained by BB and the expected revenue from the greedy heuristic. The upper bounds provided by BB are slightly better than those from CD. For economy of space, we do not provide the details on CD. The results in Table 4.2 indicate that in more than 24000 out of 27000 problem instances, we can use the upper bounds from PM to conclude that the optimality gap of the greedy heuristic is smaller than 0.25%, which also implies that the upper bounds provided by PM for these problem instances deviate from the optimal expected revenues by at most 0.25%. In contrast, the upper bounds provided by BB deviate from the expected revenues from the greedy heuristic by less than 5% in only about 11500 out of 27000 problem instances. For PM, the gaps between the upper bounds and the expected revenues from the greedy heuristic are almost exclusively less than 0.5%, whereas the gaps between the upper bounds

from BB and the expected revenues from the greedy heuristic almost never falls below 0.5%.

The run times for PM are quite reasonable. Over all of our problem instances, the average run time for PM is 3.95 seconds. Considering the problem instances with 25, 50 and 75 customer types separately, the average run time for PM is respectively 1.91, 3.94 and 6.01 seconds. Overall, our results indicate that PM can obtain quite tight upper bounds on the optimal expected revenues. The small gaps between the upper bounds from PM and the expected revenues from the greedy heuristic do not only demonstrate that the upper bounds provided by PM are close to the optimal expected revenues, but also point out that the greedy heuristic is effective in obtaining near optimal solutions. In this way, the upper bounds provided by PM can be used to check the quality of the solutions provided by not only the greedy heuristic, but also any other heuristic or approximation method that is used to obtain solutions to assortment problems, when customers choose according to a MMNL model.

4.9.4 Specially Structured Problem Instances

The computational results that we present so far indicate that the upper bounds provided by PM can be quite tight. In this section, we work with small and specially structured problem instances to demonstrate that it is possible to come up with problem instances where the upper bounds provided by PM are not quite as tight. In particular, we focus on a class of problem instances where the number of customer types $|G|$ is equal to the number of products $|N|$. For a scalar $\theta > 1$, Table 4.3 lists the parameters of a problem instance for the case with $|G| = |N| = 3$. For example, the revenues associated with the products in this problem instance are

$(r_1, r_2, r_3) = (1, \theta, \theta^2)$. A customer of type two associates the preference weights $(v_1^2, v_2^2, v_3^2) = (\theta^6, \theta^4, 0)$ with the products. The probability of observing a customer of type two is $\alpha^2 = \theta/(1 + \theta + \theta^2)$. Following the pattern in Table 4.3, it is straightforward to generalize this problem instance to a larger value for $|G|$ and $|N|$. For example, for the case with $|G| = |N| = 5$, the revenues associated with the five products are $(r_1, r_2, r_3, r_4, r_5) = (1, \theta, \theta^2, \theta^3, \theta^4)$. A customer of type one associates the preference weights $(v_1^1, v_2^1, v_3^1, v_4^1, v_5^1) = (\theta^{10}, \theta^8, \theta^6, \theta^4, \theta^2)$ with the products, whereas a customer of type two associates the preference weights $(v_1^2, v_2^2, v_3^2, v_4^2, v_5^2) = (\theta^{10}, \theta^8, \theta^6, \theta^4, 0)$. The probability of observing a customer of type two would be $\alpha^2 = \theta/(1 + \theta + \theta^2 + \theta^3 + \theta^4)$.

The motivation behind the problem instance in Table 4.3 is that if the value of θ is large, then CD, which offers different sets of products to different customer types, provides an upper bound that deviates from the optimal expected revenue by a factor that is close to $|G| = |N|$. Thus, for this problem instance, the upper bound obtained under the assumption that we can offer different sets of products to different customer types can be quite poor. To see this result, we observe that if we offer the set $\{3\}$ of products to a customer of type one, then we obtain an expected revenue of $\theta^4/(1 + \theta^2)$ from this customer type. If we offer the set $\{2\}$ of products to a customer of type two, then we obtain an expected revenue of $\theta^5/(1 + \theta^4)$ from this customer type. Lastly, if we offer the set $\{1\}$ of products to a customer of type three, then we obtain an expected revenue of $\theta^6/(1 + \theta^6)$ from this customer type. Thus, noting the probability of observing each customer type in Table 4.3, the upper bound on the optimal expected revenue that we obtain by offering different sets of products to customers of different types is at least

$$\frac{1}{1 + \theta + \theta^2} \left\{ \frac{\theta^4}{1 + \theta^2} + \theta \frac{\theta^5}{1 + \theta^4} + \theta^2 \frac{\theta^6}{1 + \theta^6} \right\}. \tag{4.17}$$

In other words, using CDU to denote the upper bound obtained by CD, CDU is no

smaller than the quantity in (4.17). On the other hand, considering the optimal expected revenue in problem (4.1) for this problem instance, we can check the expected revenue from each set of products, in which case, it is straightforward to establish that the optimal expected revenue Z^* in problem (4.1) for this problem instance is at most $(3+2\,\theta+\theta^2)/(1+\theta+\theta^2)$. We give the details of this computation in Online Appendix C.2. In this case, noting the expression in (4.17), it follows that

$$\frac{\mathsf{CDU}}{Z^*} \geq \frac{\dfrac{1}{1+\theta+\theta^2}\left\{\dfrac{\theta^4}{1+\theta^2}+\theta\,\dfrac{\theta^5}{1+\theta^4}+\theta^2\,\dfrac{\theta^6}{1+\theta^6}\right\}}{\dfrac{3+2\,\theta+\theta^2}{1+\theta+\theta^2}}.$$

As θ approaches to infinity, the expression on the right side above approaches to three. Thus, for large values of θ, the upper bound obtained by offering different sets of products to customers of different types exceeds the expected revenue by a factor that is close to three, indicating that the upper bounds provided by CD can be quite loose for this problem instance.

A natural question is how much we can improve the upper bound provided by CD for this class of problem instances through the use of penalty multipliers, which corresponds to the upper bound provided by PM. It is difficult to compute the upper bound provided by PM in closed form and we carry out numerical experiments. In Table 4.4, we give the optimal expected revenue Z^*, along with the upper bounds on the optimal expected revenue obtained by PM and CD for different problem instances parameterized by θ and $|G|$. All of the problem instances in Table 4.4 are generated by following the pattern in Table 4.3 with values of $\theta \in \{2, 4, 8\}$ and $|G| \in \{3, 4, 5\}$. The first column in this table shows the parameter combinations by using $(\theta, |G|)$. The second column shows the optimal expected revenue Z^*, corresponding to the optimal objective value of problem (4.1). Since

the number of products is reasonably small, we compute the optimal expected revenue by checking the expected revenue provided by every possible assortment. The third column shows the upper bound obtained by PM, which corresponds to the approach that propose in this chapter. The fourth column shows the upper bound obtained by CD, which corresponds to the upper bound obtained by offering different sets to different customer types. Letting PMU and CDU respectively be the upper bounds obtained by PM and CD, the fifth and sixth columns give the ratios PMU/Z^* and CDU/Z^*, characterizing the tightness of the upper bounds provided by PM and CD.

The results in Table 4.4 indicate that the upper bounds provided by CD for the specially structured problem instances can be quite loose. For example, for the problem instance with $\theta = 8$ and $|G| = 5$, the upper bound provided by CD deviates from the optimal expected revenue by a factor of about 4.3. This observation is consistent with the earlier discussion that establishes that the upper bound provided by CD deviates from the optimal expected revenue by a factor that is close to $|G|$ when θ is large. The upper bounds provided by PM can significantly improve those provided by CD. For the problem instance with $\theta = 8$ and $|G| = 5$, the upper bound provided by PM deviates from the optimal expected revenue by a factor of about 2.23, while the upper bound provided by CD deviates by a factor of about 4.3. Nevertheless, the upper bounds provided by PM can still be quite loose. Furthermore, we observe that the upper bounds provided by both CD and PM tend to get looser as $|G|$ increases and there are more customer types. Thus, the results in Table 4.4 indicate that PM can significantly improve the upper bounds from CD even for these specially structured problem instances, but it is possible to construct problem instances where the upper bounds provided by PM can still be quite loose. It is also worthwhile to note that these problem instances,

for which the upper bounds provided by PM can be quite loose, involve products whose revenues and preference weights differ by orders of magnitude. For example, even with the smallest value of two that we use for θ, if $|G| = 5$, then we have $v_1^1 = 1024$, but $v_5^1 = 4$.

There are three sources of error in the approach that PM uses to obtain upper bounds on the optimal expected revenue. As discussed in Section 4.7, PM is equivalent to using Lagrangian relaxation on an appropriate reformulation of problem (4.1). Since problem (4.1) does not have a concave objective function and it involves binary decision variables, we do not necessarily obtain the optimal objective value of this problem through Lagrangian relaxation. Thus, the first source of error is due to the fact that we use Lagrangian relaxation on a nonconvex optimization problem, potentially resulting in a duality gap. The other two sources of error, as discussed at the end of Section 4.4, is related to the approximation $\max_{k\in\{1,\ldots,K\}} \Pi^g(\lambda, p^k, p^{k+1})$ that we use for $\Pi^g(\lambda)$. In particular, the second source of error is due to the fact that we use a finite number of grid points in $\{p^k : k = 1, \ldots, K+1\}$. Proposition 20 and the following discussion imply that if we use grid points of the form $\{(1+\rho)^{-k+1} : k = 1, \ldots, K+1\}$, then this source of error is no more than a multiplicative factor of $1 + \rho$. Thus, the second source of error can be alleviated by using a small value for ρ. The third source of error is due to the fact that we do not impose binary constraints on the decision variables in problem (4.4). The intuitive motivation is that the linear programming relaxations of knapsack problems can give good approximations to the version with binary constraints. For the large problem instances used in the previous section, none of the three sources of error appears to be problematic, as the results in Table 4.1 indicate that the upper bounds obtained by PM are quite tight. In the remainder of this section, we use the specially structured problem instances to investigate

the three sources of error. For these problem instances, we compute the optimal expected revenue Z^* in problem (4.1). In addition to computing the optimal expected revenue, we use PM with $\rho \in \{0.5, 0.25, 0.1, 0.01, 0.001\}$, corresponding to five different numbers of grid points. Furthermore, we also use PM with $\rho = 0.001$, but impose binary constraints on the decision variables in problem (4.4).

Our results are summarized in Table 4.5. The first column in this table shows the parameter combinations by using $(\theta, |G|)$. The second column shows the optimal expected revenue Z^* in problem (4.1). The third to seventh columns show the upper bounds obtained by PM when we respectively use the values of 0.5, 0.25, 0.1, 0.01 and 0.001 for ρ. Finally, the eighth column shows the upper bound obtained by PM when we use the value of 0.001 for ρ and impose binary constraints on the decision variables in problem (4.4). As expected, when we use a smaller value for ρ and the set of grid points are denser, the upper bounds provided by PM become tighter. When we decrease ρ from 0.5 to 0.01, the upper bound quickly tightens, but decreasing ρ further from 0.01 to 0.001 yields a marginal improvement in the upper bound. These observations are consistent with the fact that error caused by a finite number of grid points is at most a multiplicative factor of $1 + \rho$. If we impose binary constraints on the decision variables in problem (4.4), then the upper bound provided by PM only marginally improves. Over all of our problem instances, the improvement was no larger than 8×10^{-4}, corresponding to an improvement of about 0.07%. This improvement occurs for problem instance $(\theta, |G|) = (2, 3)$. Nevertheless, the upper bounds provided by PM can still be loose when compared with the optimal expected revenue. For these specially structured problem instances, our results indicate that by using a smaller value for ρ, we can quickly overcome the source of error due to the fact that we use a finite number of grid points. In addition, the error caused by the fact that we do not impose

binary constraints in problem (4.4) does not appear to be problematic, as imposing binary constraints only marginally improves the upper bound. Thus, the most significant error appears to be due to the fact that we use Lagrangian relaxation on a nonconvex optimization problem and we have a duality gap. Thus, by specially structuring pathological problem instances, it is possible to come up with cases where the duality gap is large, but the duality gaps do not appear to be a problem in any of the large problem instances that we work with. Also, these pathological instances are relatively unlikely to appear in practice, since as mentioned above, they involve products whose revenues and preference weights differ from each other by orders of magnitude.

4.9.5 Problem Instances with a Space Constraint

In Section 4.8.1, we describe how to obtain upper bounds on the optimal expected revenue when there is a space constraint on the set of offered products. In this section, we provide computational experiments under a space constraint. We generate our test problems by using the approach described in Section 4.9.2. The only difference is that we need to generate the space consumption of each product and the total amount of space available. To come up with the space consumption of product j, we simply set $w_j = 1$. This setup corresponds to the case where we limit the total number of products in the offered set. We also carried out computational experiments where we sample the space consumption of each product from the uniform distribution over $[0, 1]$ and the performance of the upper bounds obtained by PM qualitatively remained the same. For economy of space, we report the results for the case where $w_j = 1$ for all $j \in N$. One advantage of working with the case where $w_j = 1$ for all $j \in N$ is that if there is a single customer

type and we have a limit on the number of products that can be offered, then [34] show that the optimal set of products to offer can efficiently be computed. Thus, it is tractable to compute the upper bound provided by CD when we have $w_j = 1$ for all $j \in N$, but this is not the case when different products have different space consumptions. To come up with the total amount of space available, after generating all of the other problem parameters as described in Section 4.9.2, we use the greedy heuristic at the end of Section 4.9.2 to compute a reasonably good solution without a space constraint. Using $\hat{x} = \{\hat{x}_j : j \in N\} \in \{0,1\}^{|N|}$ to capture the set of products offered in this solution, we set the total amount of space available as $c = \gamma \sum_{j \in N} w_j \hat{x}_j$, where γ is a parameter that we vary. Thus, the total amount of space available is a γ fraction of the total amount of space consumed by an unconstrained reasonably good assortment. [2] extend their mixed integer programming formulation to the case where there is a space constraint. Building on this formulation, we can continue using BB when there is a space constraint.

In our computational experiments, we vary $|G|$, $\bar{K}$, $\bar{P}_0$ and γ over $|G| \in \{20, 40\}$, $\bar{K} \in \{5, 10\}$, $\bar{P}_0 \in \{0.8, 1.0\}$ and $\gamma \in \{0.6, 0.8\}$, where $\bar{P}_0$ and $\bar{K}$ are as described in Section 4.9.2. This setup provides 16 parameter combinations. In each parameter combination, we generate 100 individual problem instances. Table 4.6 gives our computational results. The format of this table is similar to that of Table 4.1. The results in Table 4.6 indicate that PM continues to provide quite tight upper bounds on the optimal expected revenues when we have a space constraint and these upper bounds are significantly tighter than the ones from CD and BB. Over all of our problem instances, the average gap between the upper bounds obtained by PM and the expected revenues from the greedy heuristic is 0.31%. This observation implies that the average gap between the upper bounds obtained by PM and the optimal expected revenues is no larger than 0.31% as well. In the worst case, the

gap between the upper bound obtained by PM and the optimal expected revenue is 0.87%. The upper bounds provided by CD and BB are significantly looser. The average gap between the upper bounds provided by CD and the expected revenues from the greedy heuristic is 10.16%. This gap can be as large as 19.73% in the worst case. On the other hand, the average gap between the upper bounds from BB and the expected revenues from the greedy heuristic is 5.43%. This gap can reach 11.52% in the worst case.

4.9.6 Problem Instances under a Mixture of NL Models

In Section 4.8.2, we describe how to obtain upper bounds on the optimal expected revenue when customers choose according to a mixture of NL models. In this section, we provide computational experiments under a mixture of NL models. Throughout our computational experiments, the number of nests $|M|$ is 10 and the number of products in each nest $|N|$ is 5, yielding a total of 50 products. The number of customer types $|G|$ is 50. To generate our test problems, for all $i \in M$ and $j \in N$, we sample the revenue r_{ij} associated with product j in nest i from the uniform distribution over $[200, 600]$. For all $g \in G$ and $i \in M$, we sample the dissimilarity parameter γ_i^g from the uniform distribution over $[0, 1]$. To come up with the preference weights, for all $g \in G$, $i \in M$ and $j \in N$, we sample η_{ij}^g from the uniform distribution over $[0, 200]$. For all $g \in G$, we also sample η_0^g from the uniform distribution over $[0, \bar{P}_0]$, where $\bar{P}_0$ is a parameter that we vary. In this case, we set the preference weight v_{ij}^g that a customer of type g associates with product j in nest i as $v_{ij}^g = \eta_{ij}^g/(\eta_0^g)^{1/\gamma_i^g}$. In this setup, if we offer all of the products in all of the nests, then a customer of type g leaves without making a purchase with probability $1/(1+\sum_{i\in M}(V_i^g(N))^{\gamma_i^g}) = \eta_0^g/(\eta_0^g+\sum_{i\in M}(\sum_{j\in N}\eta_{ij})^{\gamma_i^g})$.

So, customers of type g are more likely to leave without making a purchase when η_0^g is larger. Thus, as $\bar{P}_0$ gets larger, customers are more likely to leave without making a purchase. To come up with the customer arrival probabilities, we sample β^g from the uniform distribution over $[0, 1]$ for all $g \in G$ and set $\alpha^g = \beta^g / \sum_{c \in G} \beta^c$.

In our computational experiments, we vary $\bar{P}_0$ over $\bar{P}_0 \in \{25, 50, 100\}$. This setup provides three parameter combinations. In each parameter combination, we generate 100 individual problem instances. Table 4.7 gives our computational results. The first column in this table shows the parameter combination by using $\bar{P}_0$. The interpretations of the second, third and fourth columns in Table 4.7 are similar to those of the second, third and fourth columns in Table 4.1. These columns respectively show the average, 95th percentile and maximum of the percent gaps between the upper bound obtained by our approach and the expected revenue from the greedy heuristic, when we focus on 100 problem instances in a particular parameter combination. The interpretations of the fifth, sixth and seventh columns in Table 4.7 are similar to those of the second, third and fourth columns in Table 4.2. These columns show the number of problem instances for which the percent gap between the upper bound obtained by our approach and the expected revenue from the greedy heuristic are respectively less than 0.5%, 1% and 2%. Over all of our problem instances, the upper bounds obtained by our approach deviate from the expected revenues from the greedy heuristic by 0.58% on average, which implies that the average gap between our upper bounds and the optimal expected revenues is no larger than 0.58%. The maximum gaps in Table 4.7 are larger than those in Table 4.1. This difference is likely due to the fact that our extension to a mixture of NL models works with problem (4.16), which is an $|M| + 1$ dimensional knapsack problem and the linear programming relaxations of such knapsack problems tend to be looser than those of one dimensional knapsack

problems. Nevertheless, we observe that in more than 85% of our problem instances, the upper bounds obtained by our approach are within 1% of the optimal expected revenues.

4.10 Conclusions

We developed a method to obtain an upper bound on the optimal expected revenue in assortment problems under a MMNL model. Our approach focuses on each customer type one by one and finds a separate assortment that maximizes the expected revenue from each customer type, but we use penalty parameters to synchronize the assortments offered to different customer types. This strategy requires solving assortment problems with a single customer type but with a fixed cost for offering a product. We develop tractable approximations to such assortment problems by assuming that the probability of not making a purchase can take values over a prespecified grid. We show how to obtain a set of good penalty parameters and a good set of grid points. We extend our approach to the case where there is a constraint on the total space consumption of the offered products or where the customers choose according to a mixture of NL models. In our computational experiments, our upper bounds for randomly generated problem instances are quite tight. However, there are pathological problem instances, where the revenues and preference weights of the products differ from each other by orders of magnitude, for which our upper bounds are not as tight. Ultimately, our approach will hopefully increase the practical use of MMNL model. Although heuristics tend to provide good assortments, it is generally difficult to check the quality of the solutions obtained by heuristics and our upper bounds allow checking the quality of the

solutions from any heuristic or approximation method.

There are several future research directions. Our approach for obtaining upper bounds has three sources of error. First, as mentioned in Section 4.7, our approach is based on using Lagrangian relaxation on a nonconvex optimization problem and there can be a duality gap. Second, we use a finite number of grid points. Third, we do not impose binary constraints in problem (4.4). In Section 4.6, we show that by using exponential grid points, we can bound the error due to the second source of error by a multiplicative factor of $1 + \rho$ for any $\rho > 0$. It is possible to show that the optimal objective value of a certain linear programming relaxation of a knapsack problem deviates from the optimal objective value of the binary version by at most a factor of two; see [48]. By using these two results, we can bound the error due the second and third sources of error by a multiplicative factor of $2\,(1 + \rho)$. However, it is difficult to bound the error due to the first source of error and it might be possible to find special cases where we can bound this error. Also, our extension to a mixture of NL models is under the assumption that the number of products in each nest is reasonably small, but the number of nests can be large. We can make extensions to the case where the number of nests is reasonably small but the number of products in each nest can be large. Briefly, the main idea for this extension is that if we have a reasonably small number of nests, then we can assume that the total preference weight of the products offered in a particular nest lies on a prespecified grid and we can carry out a search over an $|M|$ dimensional grid. Naturally, carrying out a search over an $|M|$ dimensional grid is tractable when the number of nests does not exceed three or four. A useful research direction is to make extensions to a mixture of NL models with a large number of nests and a large number of products in each nest. In addition to the NL model, it is useful to investigate upper bounds under more general choice models, for which it is difficult

to compute the optimal assortment.

Param. Comb. ($\|G\|$, $\bar{K}$, $\bar{P}_0$)	Number of Problems with a Certain % Gap between PMU^k and GRR^k						Number of Problems with a Certain % Gap between BBU^k and GRR^k					
	0.125%	0.25%	0.5%	2.5%	5%	7.5%	0.125%	0.25%	0.5%	2.5%	5%	7.5%
(25, 5, 0.6)	622	932	990	1000	1000	1000	0	0	0	105	598	939
(25, 5, 0.8)	606	908	976	1000	1000	1000	0	0	0	159	592	908
(25, 5, 1.0)	577	891	973	1000	1000	1000	0	0	1	257	757	956
(25, 10, 0.6)	642	926	984	1000	1000	1000	0	0	0	78	524	876
(25, 10, 0.8)	672	903	965	1000	1000	1000	0	0	0	83	510	828
(25, 10, 1.0)	608	889	973	1000	1000	1000	0	0	0	110	536	856
(25, 20, 0.6)	673	919	979	1000	1000	1000	0	0	0	65	518	857
(25, 20, 0.8)	665	916	984	1000	1000	1000	0	0	0	65	437	816
(25, 20, 1.0)	652	896	973	1000	1000	1000	0	0	1	85	492	824
(50, 5, 0.6)	519	907	989	1000	1000	1000	0	0	0	15	529	937
(50, 5, 0.8)	508	899	986	1000	1000	1000	0	0	0	26	518	924
(50, 5, 1.0)	498	900	980	1000	1000	1000	0	0	0	46	559	938
(50, 10, 0.6)	571	926	993	1000	1000	1000	0	0	0	15	420	878
(50, 10, 0.8)	555	924	986	1000	1000	1000	0	0	0	13	373	837
(50, 10, 1.0)	534	900	977	1000	1000	1000	0	0	0	20	376	854
(50, 20, 0.6)	573	918	987	1000	1000	1000	0	0	0	9	367	873
(50, 20, 0.8)	548	897	984	1000	1000	1000	0	0	0	11	294	755
(50, 20, 1.0)	513	893	984	1000	1000	1000	0	0	0	15	354	820
(75, 5, 0.6)	498	942	993	1000	1000	1000	0	0	0	4	399	934
(75, 5, 0.8)	516	932	990	1000	1000	1000	0	0	0	2	313	862
(75, 5, 1.0)	446	905	986	1000	1000	1000	0	0	0	8	386	874
(75, 10, 0.6)	522	944	995	1000	1000	1000	0	0	0	0	350	874
(75, 10, 0.8)	515	927	984	1000	1000	1000	0	0	0	1	257	832
(75, 10, 1.0)	471	925	992	1000	1000	1000	0	0	0	7	302	824
(75, 20, 0.6)	561	941	996	1000	1000	1000	0	0	0	0	288	873
(75, 20, 0.8)	523	921	989	1000	1000	1000	0	0	0	3	240	784
(75, 20, 1.0)	471	913	988	1000	1000	1000	0	0	0	0	238	777
Total	15059	24694	26576	27000	27000	27000	0	0	2	1202	11527	23310

Table 4.2: Distribution of the upper bounds provided by PM and BB.

Revenues

Product		
1	2	3
θ	θ^2	

Preference Weights

Cus. Typ.	Product		
	1	2	3
1	θ^6	θ^4	θ^2
2	θ^6	θ^4	0
3	θ^6	0	0

Arrival Probs.

Cus. Typ.		
1	2	3
$\frac{1}{1+\theta+\theta^2}$	$\frac{\theta}{1+\theta+\theta^2}$	$\frac{\theta^2}{1+\theta+\theta^2}$

Table 4.3: A specially structured problem instance with $|G| = |N| = 3$.

Param. Comb. $(\theta, \lvert G\rvert)$	Z^*	PMU	CDU	$\frac{\text{PMU}}{Z^*}$	$\frac{\text{CDU}}{Z^*}$
$(2,3)$	1.09	1.09	1.56	1.00	1.43
$(2,4)$	1.12	1.27	1.99	1.13	1.77
$(2,5)$	1.13	1.49	2.44	1.32	2.15
$(4,3)$	1.04	1.24	2.24	1.19	2.14
$(4,4)$	1.05	1.73	2.96	1.65	2.83
$(4,5)$	1.05	2.00	3.71	1.91	3.54
$(8,3)$	1.01	1.37	2.62	1.35	2.58
$(8,4)$	1.01	1.98	3.49	1.96	3.44
$(8,5)$	1.01	2.26	4.36	2.23	4.30

Table 4.4: Upper bounds provided by PM and CD for the specially structured problem instances.

Param. Comb. $(\theta, \lvert G\rvert)$	Z^*	PMU with a Certain Value for ρ 0.5	0.25	0.1	0.01	0.001	PMU Bin. Const.
$(2,3)$	1.09	1.44	1.30	1.18	1.10	1.09	1.09
$(2,4)$	1.12	1.79	1.47	1.34	1.28	1.27	1.27
$(2,5)$	1.13	1.89	1.67	1.60	1.50	1.49	1.49
$(4,3)$	1.04	1.68	1.43	1.31	1.25	1.24	1.24
$(4,4)$	1.05	2.09	1.91	1.78	1.73	1.73	1.73
$(4,5)$	1.05	2.47	2.24	2.07	2.01	2.00	2.00
$(8,3)$	1.01	1.72	1.59	1.46	1.38	1.37	1.37
$(8,4)$	1.01	2.22	2.23	2.10	2.01	1.98	1.98
$(8,5)$	1.01	2.63	2.52	2.38	2.27	2.26	2.26

Table 4.5: Upper bounds provided by PM with different values for ρ and with binary constraints in problem (4.4).

Param. Comb. ($\|G\|$, $\bar{K}$, $\bar{P}_0$, γ)	% Gap. of PMU^k with GRR^k			% Gap. of CDU^k with GRR^k			% Gap. of BBU^k with GRR^k		
	Avg.	95th	Max.	Avg.	95th	Max.	Avg.	95th	Max.
(20, 5, 0.8, 0.6)	0.21	0.31	0.69	10.94	14.09	15.81	4.78	6.92	8.90
(20, 5, 0.8, 0.8)	0.20	0.27	0.39	7.15	11.35	12.28	4.11	7.61	10.53
(20, 5, 1.0, 0.6)	0.21	0.27	0.45	11.20	14.36	19.73	4.27	6.97	10.01
(20, 5, 1.0, 0.8)	0.21	0.33	0.61	6.76	11.49	13.30	3.33	6.59	9.32
(20, 10, 0.8, 0.6)	0.21	0.32	0.70	11.42	15.84	17.61	5.40	8.86	9.67
(20, 10, 0.8, 0.8)	0.21	0.31	0.40	7.00	11.85	14.72	4.75	8.83	11.55
(20, 10, 1.0, 0.6)	0.20	0.22	0.34	11.02	14.14	15.18	4.96	7.49	8.53
(20, 10, 1.0, 0.8)	0.20	0.29	0.34	7.74	12.49	16.49	4.61	8.26	10.97
(40, 5, 0.8, 0.6)	0.22	0.33	0.66	11.35	14.74	15.92	5.69	8.03	11.52
(40, 5, 0.8, 0.8)	0.21	0.34	0.38	7.56	11.28	12.34	5.03	7.92	9.70
(40, 5, 1.0, 0.6)	0.21	0.25	0.55	11.47	14.44	17.31	5.05	7.42	10.44
(40, 5, 1.0, 0.8)	0.21	0.27	0.87	7.32	10.26	11.76	4.36	6.59	8.30
(40, 10, 0.8, 0.6)	0.20	0.23	0.47	11.29	13.94	15.96	5.89	7.83	9.49
(40, 10, 0.8, 0.8)	0.21	0.34	0.77	7.75	11.48	12.71	5.27	8.28	10.00
(40, 10, 1.0, 0.6)	0.21	0.32	0.49	11.92	15.17	17.01	5.77	8.70	10.32
(40, 10, 1.0, 0.8)	0.21	0.29	0.70	8.16	11.49	12.05	5.02	7.85	9.13
Average	0.31			10.16			5.43		

Table 4.6: Comparison of the upper bounds provided by PM, CD and BB under a space constraint.

Param. Comb. $\bar{P}_0$	% Gap of PMU^k with GRR^k			No. of Problems with a Certain % Gap between PMU^k and GRR^k		
	Avg.	95th	Max	0.5%	1%	2%
25	0.48	1.47	3.24	72	90	98
50	0.64	2.20	3.46	63	83	91
100	0.60	1.90	2.67	60	84	96
Average	0.58					

Table 4.7: Upper bounds provided by our approach under a mixture of NL models.

Chapter 5

Finishing Remarks

This thesis contributes a variety of algorithms and approaches to solve assortment and revenue management problems under the Markov chain, nested logit, and mixtures of multinomial logit choice models. Particularly noteworthy, is the fact that each of these choice models subsumes the MNL choice model, which has perhaps become the most well known choice for its simplicity and modeling prowess. Consequently, these choice models that we focus on represent potential ways to get a better grasp on modeling customer choice behavior. Further, developing efficient solutions to the corresponding assortment and revenue management problems is an important step towards improving the types of managerial decisions that result when retailers of all types ask themselves which sets of products to make available to their customers.

Appendix A

Appendix for Chapter 2

A.1 Appendix: Poor Performance of Nested by Revenue Subsets

In this section, we show that nested by revenue subsets can perform arbitrarily poorly for the 3.1 problem when customers choose according to the Markov chain choice model. In particular, we consider a problem instance with $2n$ products and show that the expected revenue provided by the best nested by revenue subset can deviate from the optimal expected revenue by a factor arbitrarily close to n. In our problem instance, we index the products by $N = \{1, \ldots, 2n\}$. For some $\epsilon \in (0, 1)$, the revenue of product j is given by ϵ^j. The probability that a customer arrives into the system to purchase product j is given by $\lambda_j = \epsilon^{2n-j+1}$. We assume that ϵ is small enough that $\sum_{j \in N} \lambda_j = \sum_{j \in N} \epsilon^{2n-j+1} \leq 1$. With probability $1 - \sum_{j \in N} \lambda_j$, there is no customer arrival. The assumption that we may not have a customer arrival is without loss of generality. In particular, if we want to ensure that there is a customer arrival with probability one, then we can scale the probabilities $(\lambda_1, \ldots, \lambda_{2n})$ with the same constant to ensure that $\sum_{j \in N} \lambda_j = 1$ without changing the optimal solution to the 3.1 problem. The transition probabilities are given by $\rho_{2k,2k-1} = 1 - \epsilon$ for all $k = 1, \ldots, n$. The other transition probabilities in $\{\rho_{j,i} : j, i \in N\}$ are zero. Figure A.1 shows the transition probabilities. Thus, if a customer visits product $2k$ for some $k = 1, \ldots, n$ and product $2k$ is not available, then she transitions to product $2k - 1$ with probability $1 - \epsilon$. With probability ϵ, she transitions to the no purchase option. If a customer visits product $2k - 1$ for some $k = 1, \ldots, n$ and product $2k - 1$ is not available, then she transitions to the no purchase option with probability one.

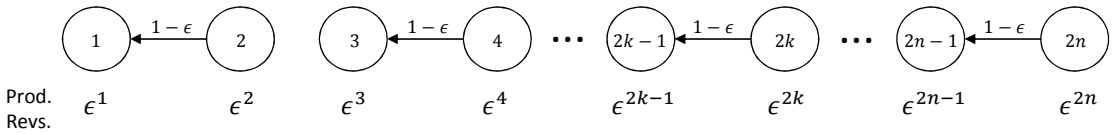


Figure A.1: A problem instance where the optimal expected revenue exceeds the expected revenue from the best nested by revenue subset by a factor arbitrarily close to n.

Since $\epsilon \in (0,1)$ and $r_j = \epsilon^j$, the revenues of the products satify $r_1 \geq \ldots \geq r_{2n}$. Thus, a nested by revenue subset for this problem instance is of the form $\{1,\ldots,j\}$ for some $j \in N$. The first row in Table A.1 shows the purchase probabilities $(P_{1,S},\ldots,P_{2n,S})$ when we offer the nested by revenue subset $S = \{1,\ldots,2k\}$ for some $k = 1,\ldots,n$. Similarly, the second row of Table A.1 shows the purchase probabilities $(P_{1,S},\ldots,P_{2n,S})$ when we offer the nested by revenue subset $S = \{1,\ldots,2k-1\}$ for some $k = 1,\ldots,n$. Finally, the third row of Table A.1 shows the purchase probabilities $(P_{1,S},\ldots,P_{2n,S})$ when we offer the subset $S = \{1,3,5,\ldots,2k-1\}$ for some $k = 1,\ldots,n$, which is not a nested by revenue subset. All of these purchase probabilities are obtained by solving the Balance equations for (P_S, R_S). To give an idea, we briefly verify some of the purchase probabilities $(P_{1,S},\ldots,P_{2n,S})$ when we offer the subset $S = \{1,3,5,\ldots,2k-1\}$. Since $1 \in S$ and $2 \notin S$, by the Balance equations, we $R_{1,S} = 0$ and $P_{2,S} = 0$. Thus, the Balance equations imply that $P_{1,S} = \lambda_1 + \rho_{2,1}\, R_{2,S}$ and $R_{2,S} = \lambda_2$, in which case, we obtain $R_{2,S} = \epsilon^{2n-1}$ and $P_{1,S} = \epsilon^{2n} + (1-\epsilon)\,\epsilon^{2n-1} = \epsilon^{2n-1}$. Similarly, since $3 \in S$ and $4 \notin S$, by the Balance equations, we $R_{3,S} = 0$ and $P_{4,S} = 0$. Therefore, the Balance equations imply that $P_{3,S} = \lambda_3 + \rho_{4,3}\, R_{4,S}$ and $R_{4,S} = \lambda_4$, in which case, we obtain $R_{4,S} = \epsilon^{2n-3}$ and $P_{3,S} = \epsilon^{2n-2} + (1-\epsilon)\,\epsilon^{2n-3} = \epsilon^{2n-3}$. All of the other purchase probabilities in the bottom portion of Table A.1 can be verified in a similar fashion.

Focusing on the first row of Table A.1, if we offer the nested by revenue subset

S	$P_{1,S}$	$P_{2,S}$	$P_{3,S}$	$\ldots$	$P_{2k-2,S}$	$P_{2k-1,S}$	$P_{2k,S}$	$P_{2k+1,S}$	$\ldots$	$P_{2n,S}$
$\{1,\ldots,2k\}$	ϵ^{2n}	ϵ^{2n-1}	ϵ^{2n-2}	$\ldots$	$\epsilon^{2n-2k+3}$	$\epsilon^{2n-2k+2}$	$\epsilon^{2n-2k+1}$	0	$\ldots$	0
$\{1,\ldots,2k-1\}$	ϵ^{2n}	ϵ^{2n-1}	ϵ^{2n-2}	$\ldots$	$\epsilon^{2n-2k+3}$	$\epsilon^{2n-2k+1}$	0	0	$\ldots$	0
$\{1,3,5,\ldots,2k-1\}$	ϵ^{2n-1}	0	ϵ^{2n-3}	$\ldots$	0	$\epsilon^{2n-2k+1}$	0	0	$\ldots$	0

Table A.1: A problem instance where the optimal expected revenue exceeds the expected revenue from the best nested by revenue subset by a factor arbitrarily close to n.

$S = \{1, \ldots, 2k\}$ for some $k = 1, \ldots, n$, then we obtain an expected revenue of $\sum_{j \in N} P_{j,S}\, r_j = \epsilon^{2n}\, \epsilon + \epsilon^{2n-1}\, \epsilon^2 + \ldots + \epsilon^{2n-2k+2}\, \epsilon^{2k-1} + \epsilon^{2n-2k+1}\, \epsilon^{2k} = 2k\, \epsilon^{2n+1}$. This expected revenue is increasing in k. Thus, if we want to obtain the largest expected revenue by offering a nested by revenue subset of the form $\{1, \ldots, 2k\}$ for some $k = 1, \ldots, n$, then we set $k = n$, yielding an expected revenue of $2n\, \epsilon^{2n+1}$. Focusing on the second row of Table A.1, if we offer the nested by revenue subset $S = \{1, \ldots, 2k-1\}$ for some $k = 1, \ldots, n$, then we obtain an expected revenue of $\sum_{j \in N} P_{j,S}\, r_j = \epsilon^{2n}\, \epsilon + \epsilon^{2n-1}\, \epsilon^2 + \ldots + \epsilon^{2n-2k+3}\, \epsilon^{2k-2} + \epsilon^{2n-2k+1}\, \epsilon^{2k-1} = (2k-2)\, \epsilon^{2n+1} + \epsilon^{2n}$. This expected revenue is also increasing in k. Thus, if we want to obtain the largest expected revenue by offering a nested by revenue subset of the form $\{1, \ldots, 2k-1\}$ for some $k = 1, \ldots, n$, then we set $k = n$, yielding an expected revenue of $(2n-2)\, \epsilon^{2n+1} + \epsilon^{2n}$. Putting the results in this paragraph together, if we offer the nested by revenue subset $\{1, \ldots, 2k\}$ for some $k = 1, \ldots, n$, then the largest expected revenue is $2n\, \epsilon^{2n+1}$. If we offer the nested by revenue subset $\{1, \ldots, 2k-1\}$ for some $k = 1, \ldots, n$, then the largest expected revenue is $(2n-2)\, \epsilon^{2n+1} + \epsilon^{2n}$. Thus, the expected revenue from any nested by revenue subset is no larger than $2n\, \epsilon^{2n+1} + \epsilon^{2n}$.

Focusing on the third row of Table A.1, if we offer the subset $S = \{1, 3, 5, \ldots, 2k-1\}$ for some $k = 1, \ldots, n$, then we obtain an expected revenue of $\sum_{j \in N} P_{j,S}\, r_j = \epsilon^{2n-1}\, \epsilon + \epsilon^{2n-3}\, \epsilon^3 + \ldots + \epsilon^{2n-2k+1}\, \epsilon^{2k-1} = k\, \epsilon^{2n}$. This expected revenue is also increasing in k. Thus, if we want to obtain the largest expected revenue by offering a subset of the form $\{1, 3, 5, \ldots, 2k-1\}$ for some $k = 1, \ldots, n$, then we set $k = n$, yielding an expected revenue of $n\, \epsilon^{2n}$. At the end of the previous paragraph, we observe that the expected revenue from any nested by revenue subset is no larger than $2n\, \epsilon^{2n+1} + \epsilon^{2n}$, but the expected revenue from the subset $\{1, 3, 5, \ldots, 2n-1\}$ is $n\, \epsilon^{2n}$. Therefore, the ratio between the optimal expected

revenue and the expected revenue from the best nested by revenue subset is at least $n\,\epsilon^{2n}/(2n\,\epsilon^{2n+1}+\epsilon^{2n}) = n/(2n\,\epsilon+1)$. Since we have $\lim_{\epsilon\to 0} n/(2n\,\epsilon+1) = n$, by choosing ϵ sufficiently small, we obtain a problem instance, where the optimal expected revenue exceeds the expected revenue from the best nested by revenue subset by a factor arbitrarily close to n.

A.2 Appendix: Optimality of Nested by Revenue Subsets

Below is the proof of Lemma 5.

Proof of Lemma 5. Letting $\hat{v}$ be the optimal solution to the DUAL problem, we define the subset $\hat{S} = \{j \in N : \hat{v}_j = r_j\}$. By Theorem 3, $\hat{S}$ is the optimal solution to the 3.1 problem. To get a contradiction, assume that $\hat{S}$ is not a nested by revenue subset so that $j_1 \notin \hat{S}$ and $j_2 \in \hat{S}$ for some $j_1, j_2 \in N$ with $j_1 < j_2$. Since $\hat{v}$ is a feasible solution to the DUAL problem, we have $v_{j_1} \geq r_{j_1}$ and $v_{j_1} \geq \sum_{i\in N} \rho_{j_1,i}\,\hat{v}_i$. By the same argument in the proof of Theorem 3, we have $\hat{v}_j = r_j$ or $\hat{v}_j = \sum_{i\in N} \rho_{j,i}\,\hat{v}_i$ for all $j \in N$. Thus, since $j_1 \notin \hat{S}$, we have $\hat{v}_{j_1} \neq r_{j_1}$, which implies that $\sum_{i\in N} \rho_{j_1,i}\,\hat{v}_i = \hat{v}_{j_1} > r_{j_1}$. Furthermore, we have $r_{j_2} = v_{j_2} \geq \sum_{i\in N} \rho_{j_2,i}\,\hat{v}_i$, where the equality uses the fact that $j_2 \in \hat{S}$ and the inequality uses the fact that $\hat{v}$ is a feasible solution to the DUAL problem. Noting that $j_1 < j_2$ and the products are indexed such that $r_1 \geq \ldots \geq r_n$, it follows that $\sum_{i\in N} \rho_{j_1,i}\,\hat{v}_i = \hat{v}_{j_1} > r_{j_1} \geq r_{j_2} = v_{j_2} \geq \sum_{i\in N} \rho_{j_2,i}\hat{v}_i$, indicating that $\sum_{i\in N} \rho_{j_1,i}\,\hat{v}_i > \sum_{i\in N} \rho_{j_2,i}\,\hat{v}_i$. However, since $j_1 < j_2$, by the assumption in the lemma, we have $\rho_{j_1,i} \leq \rho_{j_2,i}$ for all $i \in N$. Therefore, we cannot have $\sum_{i\in N} \rho_{j_1,i}\,\hat{v}_i > \sum_{i\in N} \rho_{j_2,i}\,\hat{v}_i$ and we reach a contradiction. □

A.3 Appendix: Nesting Order of Optimal Subsets

Below is the proof of Lemma 7, which follows as a corollary to Lemma 5 and Theorem 6.

Proof of Lemma 7. We observe that $\hat{S}_t(x)$ is given by the optimal solution to the problem $\max_{S \subset N} \sum_{j \in N} P_{j,S} \left(r_j + V_{t+1}(x-1) - V_{t+1}(x) \right)$. This problem has the same form as the 3.1 problem, where the revenue of product j is given by $r_j + V_{t+1}(x-1) - V_{t+1}(x)$. Since the products are indexed such that $r_1 \geq \ldots \geq r_n$, we have $r_1 + V_{t+1}(x-1) - V_{t+1}(x) \geq \ldots \geq r_n + V_{t+1}(x-1) - V_{t+1}(x)$. In this case, Lemma 5 implies that the optimal solution to the problem $\max_{S \subset N} \sum_{j \in N} P_{j,S} \left(r_j + V_{t+1}(x-1) - V_{t+1}(x) \right)$ is a nested by revenue subset of the form $\{1, \ldots, j_t(x)\}$ for some $j_t(x) \in N \cup \{0\}$. Thus, $\hat{S}_t(x) = \{1, \ldots, j_t(x)\}$ for some $j_t(x) \in N \cup \{0\}$. By Theorem 6, we have $\hat{S}_t(x-1) \subset \hat{S}_t(x)$ and $\hat{S}_{t-1}(x) \subset \hat{S}_t(x)$, which implies that $j_t(x)$ is increasing in x and increasing in t. $\square$

A.4 Appendix: Comparison of Purchase Probabilities

For $(\hat{x}, \hat{z}) \in \mathcal{H}$, we define $S_{\hat{x}} = \{ j \in N : \hat{x}_j > 0 \}$. In this section, our objective is to show that we have $\hat{z}_j \geq R_{j,S_{\hat{x}}}$ for all $j \in N$. This result is used in several places throughout Section 2.6. In the next lemma, we give a preliminary result that shows that if we offer a larger subset of products, then the probability that a

customer considers purchasing a product during her choice process and does not purchase this product becomes smaller. This result becomes useful when we show that $\hat{z}_j \geq R_{j,S_{\hat{x}}}$ for all $j \in N$ later in this section.

Lemma 21. *For $\hat{S} \subset S$, we have $R_{j,\hat{s}} \geq R_{j,S}$ for all $j \in N$.*

Proof. We define $q = (q_1, \ldots, q_n)$ as $q_j = 0$ for all $j \in S$ and $q_j = \min\{R_{j,\hat{s}}, R_{j,S}\}$ for all $j \notin S$. By the definition of q, we have $q_j \leq R_{j,S}$ and $q_j \leq R_{j,\hat{s}}$ for all $j \in N$. By the BALANCE equations, noting that $P_{j,S} = 0$ for all $j \notin S$, we have $R_{j,S} = \lambda_j + \sum_{i \in N} \rho_{i,j} R_{i,S}$ for all $j \notin S$. Similarly, the BALANCE equations imply that $R_{j,\hat{s}} = \lambda_j + \sum_{i \in N} \rho_{i,j} R_{i,\hat{s}}$ for all $j \notin \hat{S}$. Since $\hat{S} \subset S$ and the last equality holds for all $j \notin \hat{S}$, the last equality also yields $R_{j,\hat{s}} = \lambda_j + \sum_{i \in N} \rho_{i,j} R_{i,\hat{s}}$ for all $j \notin S$. Therefore, for all $j \notin S$, we can use the definition of q_j to obtain

$$q_j = \min\{R_{j,\hat{s}}, R_{j,S}\} = \min\left\{\lambda_j + \sum_{i \in N} \rho_{i,j} R_{i,\hat{s}},\ \lambda_j + \sum_{i \in N} \rho_{i,j} R_{i,S}\right\} \geq \lambda_j + \sum_{i \in N} \rho_{i,j} q_i,$$

where the inequality follows from the fact that $q_j \leq R_{j,S}$ and $q_j \leq R_{j,\hat{s}}$ for all $j \in N$. Using $\mathbf{1}(\cdot)$ to denote the indicator function and adding the inequality above over all $j \notin S$, it follows that $\sum_{j \in N} q_j = \sum_{j \in N} \mathbf{1}(j \notin S)\, q_j \geq \sum_{j \in N} \mathbf{1}(j \notin S)\, \lambda_j + \sum_{j \in N} \sum_{i \in N} \mathbf{1}(j \notin S)\, \rho_{i,j}\, q_i$, where the first equality is by the fact that $q_j = 0$ for all $j \in S$. Focusing on the first and last expressions in the last chain of inequalities and arranging the terms, we obtain $\sum_{j \in N}(1 - \sum_{i \in N} \mathbf{1}(i \notin S)\, \rho_{j,i})\, q_j \geq \sum_{j \in N} \mathbf{1}(j \notin S)\, \lambda_j$. Earlier in the proof, we observe that we have $R_{j,S} = \lambda_j + \sum_{i \in N} \rho_{i,j} R_{i,S}$ for all $j \notin S$. In this case, if we add these equalities over all $j \notin S$ and arrange the terms, then we obtain $\sum_{j \in N}(1 - \sum_{i \in N} \mathbf{1}(i \notin S)\, \rho_{j,i}) R_{j,S} = \sum_{j \in N} \mathbf{1}(j \notin S)\, \lambda_j$. Therefore, the last equality and the last inequality yield $\sum_{j \in N}(1 - \sum_{i \in N} \mathbf{1}(i \notin S)\, \rho_{j,i})\, q_j \geq \sum_{j \in N}(1 - \sum_{i \in N} \mathbf{1}(i \notin S)\, \rho_{j,i}) R_{j,S}$. Noting that $q_j \leq R_{j,S}$ for all $j \in N$ and using the assumption that $\sum_{i \in N} \rho_{j,i} < 1$, this inequality implies that $q_j = R_{j,S}$ for all $j \in N$. By the definition of q, we have $q_j = 0$ for all $j \in S$, but since $q_j = R_{j,S}$, we

obtain $q_j = R_{j,S} = 0 \leq R_{j,\hat{s}}$ for all $j \in S$. We also have $q_j = \min\{R_{j,\hat{s}}, R_{j,S}\}$ for all $j \notin S$, but since $q_j = R_{j,S}$, it must be the case that $R_{j,S} \leq R_{j,\hat{s}}$ for all $j \notin S$. $\square$

In the next lemma, we give the main result of this section.

Lemma 22. *For $(\hat{x}, \hat{z}) \in \mathcal{H}$, define $S_{\hat{x}} = \{j \in N : \hat{x}_j > 0\}$. Then, we have $\hat{z}_j \geq R_{j,S_{\hat{x}}}$ for all $j \in N$.*

Proof. Since $(\hat{x}, \hat{z}) \in \mathcal{H}$, there exist extreme points $(x^1, z^1), \ldots, (x^K, z^K)$ of $\mathcal{H}$ and positive scalars $\gamma^1, \ldots, \gamma^K$ summing to one such that $\hat{x} = \sum_{k=1}^{K} \gamma^k x^k$ and $\hat{z} = \sum_{k=1}^{K} \gamma^k z^k$. Without loss of generality, we assume that the scalars $\gamma^1, \ldots, \gamma^K$ are strictly positive. Otherwise, we can drop the scalars that are equal to zero. In this case, if we define the subset $S^k = \{j \in N : x^k_j > 0\}$, then Lemma 2 implies that $P_{j,S^k} = x^k_j$ and $R_{j,S^k} = z^k_j$ for all $j \in N$. Thus, we have $\hat{x}_j = \sum_{k=1}^{K} \gamma^k x^k_j = \sum_{k=1}^{K} \gamma^k P_{j,S^k}$ and $\hat{z}_j = \sum_{k=1}^{K} \gamma^k z^k_j = \sum_{k=1}^{K} \gamma^k R_{j,S^k}$ for all $j \in N$. We consider one of the subsets $S^1, \ldots, S^K$. If $j \in S^k$, then we have $P_{j,S^k} > 0$, but since $\hat{x}_j = \sum_{k=1}^{K} \gamma^k P_{j,S^k}$, we obtain $\hat{x}_j > 0$. Therefore, having $j \in S^k$ implies that $\hat{x}_j > 0$, in which case, by the definition of $S_{\hat{x}}$, we obtain $j \in S_{\hat{x}}$. Thus, we have $S^k \subset S_{\hat{x}}$. In this case, since we have $S^k \subset S_{\hat{x}}$ for all $k = 1, \ldots, K$, by Lemma 21, it follows that $R_{j,S^k} \geq R_{j,S_{\hat{x}}}$ for all $j \in N$, $k = 1, \ldots, K$. Using the last inequality, we obtain $\hat{z}_j = \sum_{k=1}^{K} \gamma^k R_{j,S^k} \geq \sum_{k=1}^{K} \gamma^k R_{j,S_{\hat{x}}} = R_{j,S_{\hat{x}}}$ for all $j \in N$. $\square$

Appendix B

Appendix for Chapter 3

B.1 Online Appendix: Proof of Corollary 13

We let $(S_1^*, \ldots, S_m^*)$ be an optimal solution to problem (3.1) so that $z^* = \Pi(S_1^*, \ldots, S_m^*) = \sum_{i \in M} V_i(S_i^*)^{\gamma_i} R_i(S_i^*) / (v_0 + \sum_{i \in M} V_i(S_i^*)^{\gamma_i})$. Focusing on the first and last terms in this chain of equalities and solving for z^*, we obtain $v_0\, z^* = \sum_{i \in M} V_i(S_i^*)^{\gamma_i} (R_i(S_i^*) - z^*)$. Since $(S_1^*, \ldots, S_m^*)$ is a feasible solution to problem (3.2) when we solve this problem with $z = z^*$, we obtain $f(z^*) \geq \sum_{i \in M} V_i(S_i^*)^{\gamma_i} (R_i(S_i^*) - z^*)$, in which case, using the last equality, we have $f(z^*) \geq v_0\, z^*$. We claim that $\alpha\, \hat{z} \geq z^*$. To get a contradiction, assume that $\alpha\, \hat{z} < z^*$. In this case, we obtain $f(z^*) \geq v_0\, z^* > \alpha\, v_0\, \hat{z} = \alpha\, f^R(\hat{z}) \geq f(\hat{z})$, where the equality follows from the definition of $\hat{z}$. Since $f(\cdot)$ is decreasing, having $f(z^*) \geq f(\hat{z})$ implies that $z^* \leq \hat{z} \leq \alpha\, \hat{z}$, which contradicts the assumption that $\alpha\, \hat{z} < z^*$ and the claim follows. To obtain the desired result, we observe that $\sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i} (\alpha\, \beta\, R_i(\hat{S}_i) - z^*) \geq \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i} (\alpha\, \beta\, R_i(\hat{S}_i) - \beta\, z^*) \geq \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i} (\alpha\, \beta\, R_i(\hat{S}_i) - \alpha\, \beta\, \hat{z}) \geq \alpha\, f^R(\hat{z}) = \alpha\, v_0\, \hat{z} \geq v_0\, z^*$, where the first inequality follows from the fact that $\beta \geq 1$, the second inequality is by the fact that $\alpha\, \hat{z} \geq z^*$ and the third inequality follows from the inequality given in the corollary. Focusing on the first and last expressions in the last chain of inequalities and solving for z^*, we obtain $z^* \leq \alpha\, \beta \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i} R_i(\hat{S}_i) / (v_0 + \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i}) = \alpha\, \beta\, \Pi(\hat{S}_1, \ldots, \hat{S}_m)$.

B.2 Online Appendix: An Upper Bound

The approach in Section 4.8.1 obtains a 4-approximate solution under a space constraint, indicating that this approach never performs arbitrarily badly. However, knowing that a solution provides at least a quarter of the optimal expected

revenue may not be thoroughly satisfying from a practical perspective. In this section, we develop a tractable approach for obtaining an upper bound on the optimal expected revenue for an individual instance of problem (3.1) under a space constraint. By comparing this upper bound on the optimal expected revenue with the expected revenue obtained by a particular assortment, we can get a feel for the optimality gap of the assortment on hand.

To construct an upper bound on the optimal expected revenue in problem (3.1), for each nest i, we partition the interval $[0, c]$ into K intervals $\{[b_i^{k-1}, b_i^k] : k = 1, \ldots, K\}$, where we have $0 = b_i^0 \leq b_i^1 \leq \ldots \leq b_i^{K-1} \leq b_i^K = c$. Noting that the total preference weight of the products offered in nest i can at most be $\sum_{j \in N} v_{ij}$, we let $\bar{v}_i = \sum_{j \in N} v_{ij}$ and partition the interval $[0, \bar{v}_i]$ into L intervals $\{[\nu_i^{q-1}, \nu_i^q] : q = 1, \ldots, L\}$ with $0 = \nu_i^0 \leq \nu_i^1 \leq \ldots \leq \nu_i^{L-1} \leq \nu_i^L = \bar{v}_i$. Using the decision variables $x_i = \{x_{ij} : j \in N\} \in [0, 1]^n$, we define $\phi_i^{kq}(z)$ as

$$\begin{aligned}
\phi_i^{kq}(z) \quad = \quad & \max \quad (\nu_i^{q-1})^{\gamma_i} \left\{ \frac{\sum_{j \in N} v_{ij}\, r_{ij}\, x_{ij}}{\nu_i^{q-1}} - z \right\} && \text{(B.1)} \\
& \text{st} \quad \sum_{j \in N} w_{ij}\, x_{ij} \leq b_i^k && \text{(B.2)} \\
& \qquad \sum_{j \in N} v_{ij}\, x_{ij} \leq \nu_i^q && \text{(B.3)} \\
& \qquad 0 \leq x_{ij} \leq \mathbf{1}(w_{ij} \leq b_i^k) \qquad \forall\, j \in N, && \text{(B.4)}
\end{aligned}$$

which is a continuous knapsack problem with two dimensions. The selection of the intervals $\{[b_i^{k-1}, b_i^k] : k = 1, \ldots, K\}$ and $\{[\nu_i^{q-1}, \nu_i^q] : q = 1, \ldots, L\}$ can be completely arbitrary, as long as these intervals respectively cover $[0, c]$ and $[0, \bar{v}_i]$. We observe that $\phi_i^{kq}(z)$ is a linear function of z. If $q = 1$, then $\nu_i^{q-1} = 0$, in which case, we have a zero in the denominator of the fraction above. To deal with this case, if $q = 1$, then we follow the convention that $\phi_i^{kq}(z) = 0$ for all $k = 1, \ldots, K$ and $z \in \Re_+$. Roughly speaking, we can interpret problem (B.1)-(B.4) as a continuous

version of problem (3.9). In the objective function of problem (B.1)-(B.4), the term ν_i^{q-1} corresponds to $V_i(S_i)$ in the objective function of problem (3.9). Noting that $R_i(S_i) = \sum_{j \in S_i} r_{ij}\, v_{ij} / V_i(S_i)$, the term $\sum_{j \in N} v_{ij}\, r_{ij}\, x_{ij} / \nu_i^{q-1}$ in the objective function of problem (B.1)-(B.4) corresponds to $R_i(S_i)$ in the objective function of problem (3.9). The first constraint in problem (B.1)-(B.4) imposes the capacity constraint, whereas the second constraint ensures that the total preference weight of the offered products are computed correctly. We use ν_i^{q-1} in the objective function, but ν_i^q in the constraint to ultimately ensure that we can use $\phi_i^{kq}(z)$ to obtain an upper bound on the optimal expected revenue. Using the decision variables Δ, $y = \{y_i : i \in M\}$ and z, to obtain an upper bound on the optimal expected revenue, we propose solving the problem

$$\begin{aligned}
\min \quad & c\,\Delta + \sum_{i \in M} y_i && && \text{(B.5)}\\
\text{st} \quad & b_i^{k-1}\,\Delta + y_i \geq \phi_i^{kq}(z) && \forall\, i \in M,\ k = 1, \ldots, K,\ q = 1, \ldots, L && \text{(B.6)}\\
& c\,\Delta + \sum_{i \in M} y_i = v_0\, z && && \text{(B.7)}\\
& \Delta \geq 0,\ y_i \text{ is free},\ z \text{ is free} && \forall\, i \in M. && \text{(B.8)}
\end{aligned}$$

Since $\phi_i^{kq}(\cdot)$ is linear, the problem above is a linear program. The next theorem shows that we can use this problem to obtain an upper bound on the optimal expected revenue z^* in problem (3.1).

Theorem 23. *Letting $(\hat{\Delta}, \hat{y}, \hat{z})$ be an optimal solution to problem* (B.5)-(B.8)*, we have $\hat{z} \geq z^*$.*

Proof. We let $(S_1^*, \ldots, S_m^*)$ be an optimal solution to problem (3.1), k_i' be such that $C_i(S_i^*) \in [b_i^{k_i'-1}, b_i^{k_i'}]$ and q_i' be such that $V_i(S_i^*) \in [\nu_i^{q_i'-1}, \nu_i^{q_i'}]$. Since $(\hat{\Delta}, \hat{y}, \hat{z})$ is a feasible solution to problem (B.5)-(B.8), we have $b_i^{k_i'-1}\,\hat{\Delta} + \hat{y}_i \geq \phi_i^{k_i' q_i'}(\hat{z})$ for all $i \in M$. Adding this inequality over all $i \in M$, we obtain $\sum_{i \in M} \phi_i^{k_i' q_i'}(\hat{z}) \leq$

$\sum_{i\in M} b_i^{k_i'-1}\,\hat{\Delta} + \sum_{i\in M}\hat{y}_i \le \sum_{i\in M} C_i(S_i^*)\,\hat{\Delta} + \sum_{i\in M}\hat{y}_i \le c\,\hat{\Delta} + \sum_{i\in M}\hat{y}_i = v_0\,\hat{z}$, where the second inequality uses the fact that $C_i(S_i^*) \ge b_i^{k_i'-1}$, the third inequality uses the fact that $(S_1^*,\ldots,S_m^*)$ is a feasible solution to problem (3.1) and the equality is by the fact that $(\hat{\Delta},\hat{y},\hat{z})$ is a feasible solution to problem (B.5)-(B.8). Thus, the last chain of inequalities implies that $\sum_{i\in M}\phi_i^{k_i'q_i'}(\hat{z}) \le v_0\,\hat{z}$. On the other hand, consider a solution x_i^* to problem (B.1)-(B.4) obtained by letting $x_{ij}^* = 1$ if $j \in S_i^*$ and $x_{ij}^* = 0$ otherwise. Since $\sum_{j\in N} w_{ij}\,x_{ij}^* = C_i(S_i^*) \le b_i^{k_i'}$ and $\sum_{j\in N} v_{ij}\,x_{ij}^* = V_i(S_i^*) \le \nu_i^{q_i'}$, the solution x_i^* is feasible to problem (B.1)-(B.4) when we solve this problem with $k = k_i'$, $q = q_i'$ and $z = \hat{z}$. So, the optimal objective value of problem (B.1)-(B.4) is at least as large as the objective value provided by the feasible solution x_i^* and we obtain

$$\begin{aligned}
\phi_i^{k_i'q_i'}(\hat{z}) &\ge (\nu_i^{q_i'-1})^{\gamma_i}\left\{\frac{\sum_{j\in N} v_{ij}\,r_{ij}\,x_{ij}^*}{\nu_i^{q_i'-1}} - \hat{z}\right\} = \frac{\sum_{j\in N} v_{ij}\,r_{ij}\,x_{ij}^*}{(\nu_i^{q_i'-1})^{1-\gamma_i}} - (\nu_i^{q_i'-1})^{\gamma_i}\,\hat{z} \\
&\ge \frac{\sum_{j\in N} v_{ij}\,r_{ij}\,x_{ij}^*}{V_i(S_i^*)^{1-\gamma_i}} - V_i(S_i^*)^{\gamma_i}\,\hat{z} = V_i(S_i^*)^{\gamma_i}\left\{\frac{\sum_{j\in S_i^*} v_{ij}\,r_{ij}}{V_i(S_i^*)} - \hat{z}\right\} \\
&= V_i(S_i^*)^{\gamma_i}(R_i(S_i^*) - \hat{z}),
\end{aligned}$$

where the second inequality uses the fact that $V_i(S_i^*) \ge \nu_i^{q_i'-1}$ and the second equality uses the definition of x_i^*. Since we have $\sum_{i\in M}\phi_i^{k_i'q_i'}(\hat{z}) \le v_0\hat{z}$ as shown at the beginning of the proof, the chain of inequalities above implies that $v_0\,\hat{z} \ge \sum_{i\in M}\phi_i^{k_i'q_i'}(\hat{z}) \ge \sum_{i\in M} V_i(S_i^*)^{\gamma_i}\,(R_i(S_i^*) - \hat{z})$. If we focus on the first and last expressions in this chain of inequalities and solve for $\hat{z}$, then we obtain $\hat{z} \ge \sum_{i\in M} V_i(S_i^*)^{\gamma_i}\,R_i(S_i^*)/(v_0 + \sum_{i\in M} V_i(S_i^*)^{\gamma_i}) = \Pi(S_1^*,\ldots,S_m^*) = z^*$. □

Appendix C

Appendix for Chapter 4

C.1 Online Appendix: Upper Bounds from a Linear Programming Formulation

[2] show that problem (4.1) can be formulated as a mixed integer program. Thus, a tempting approach to obtain upper bounds on the optimal expected revenue is to solve the linear programming relaxation of this mixed integer program. In this section, we show that when we focus on each customer type individually, if customers of each type make a purchase with a reasonably large probability, then the optimal objective value of the linear programming relaxation of the mixed integer program is equal to the upper bound on the optimal expected revenue that is obtained under the assumption that we can offer different sets of products to different customer types. To state this result, we note that [2] show that we can obtain the optimal objective value of problem (4.1) by solving the mixed integer program

$$\begin{aligned}
\max \quad & \sum_{g\in G}\sum_{j\in N} \alpha^g\, r_j\, v_j^g\, y_j^g && && \text{(C.1)}\\
\text{subject to} \quad & \sum_{j\in N} v_j^g\, y_j^g + w^g = 1 && \forall\, g\in G\\
& y_j^g \le w^g && \forall\, j\in N,\ g\in G\\
& y_j^g \le z_j && \forall j\in N,\ g\in G\\
& w^g - y_j^g \le 1 - z_j && \forall j\in N,\ g\in G\\
& y_j^g \ge 0,\ w^g \ge 0,\ z_j\in\{0,1\} && \forall j\in N,\ g\in G.
\end{aligned}$$

We use the vector $\hat{x}^g = \{\hat{x}_j^g : j\in N\} \in \{0,1\}^{|N|}$ to capture the set of products that maximizes the expected revenue only from customers of type g. In other words, $\hat{x}^g$ is an optimal solution to the problem $\hat{Z}^g = \max_{x\in\{0,1\}^{|N|}} \sum_{j\in N} r_j\, P_j^g(x)$ with

the corresponding optimal objective value $\hat{Z}^g$. Therefore, the expected revenue $\sum_{g \in G} \alpha^g \hat{Z}^g$ provides an upper bound on the optimal objective value of problem (4.1), since the expected revenue $\sum_{g \in G} \alpha^g \hat{Z}^g$ is obtained under the assumption that we can offer different sets of products to different customer types, whereas problem (4.1) requires that we offer a single set of products to customers of all types. In this case, if we offer the set of products captured by the vector $\hat{x}^g$, then a customer of type g makes a purchase within the set of offered products with probability $\hat{P}^g = \sum_{j \in N} P_j^g(\hat{x}^g)$.

The next proposition shows that if we have $\hat{P}^g \geq 1/2$ for all $g \in G$, then the optimal objective value of the linear programming relaxation of problem (C.1) is equal to $\sum_{g \in G} \alpha^g \hat{Z}^g$. So, if customers of each type g make a purchase with a probability of at least $1/2$ when offered the set of products captured by the vector $\hat{x}^g$, then the upper bound on the optimal expected revenue provided by the linear programming relaxation of problem (C.1) does not improve the upper bound obtained under the assumption that we can offer different sets of products to different customer types.

Proposition 24. *Let $\hat{x}^g$ be an optimal solution to the problem $\max_{x \in \{0,1\}^{|N|}} \sum_{j \in N} r_j P_j^g(x)$ with the corresponding optimal objective value $\hat{Z}^g$ and $\hat{P}^g$ be defined as $\hat{P}^g = \sum_{j \in N} P_j^g(\hat{x}^g)$. If we have $\hat{P}^g \geq 1/2$ for all $g \in G$, then the optimal objective value of the linear programming relaxation of problem* (C.1) *is equal to $\sum_{g \in G} \alpha^g \hat{Z}^g$.*

Proof. We let $\hat{\zeta}$ be the optimal objective value of the linear programming relaxation of problem (C.1). First, we show that $\hat{\zeta} \leq \sum_{g \in G} \alpha^g \hat{Z}^g$. We use $\hat{y} = \{\hat{y}_j^g : j \in N,\ g \in G\}$, $\hat{w} = \{\hat{w}^g : g \in G\}$ and $\hat{z} = \{\hat{z}_j : j \in N\}$ to denote an optimal solution to the linear programming relaxation of problem (C.1) and let $\hat{\zeta}^g$ be defined as $\hat{\zeta}^g =$

$\sum_{j \in N} r_j \, v_j^g \, \hat{y}_j^g$, in which case, we have $\hat{\zeta} = \sum_{g \in G} \alpha^g \, \hat{\zeta}^g$. We observe that we must have $\hat{w}^g > 0$ for all $g \in G$, since if $\hat{w}^g = 0$ for some $g \in G$, then the second set of constraints in problem (C.1) imply that $\hat{y}_j^g = 0$ for all $j \in N$ as well, in which case, it is not possible to satisfy the first set of constraints. Now, we claim that $\hat{\zeta}^g \leq \hat{Z}^g$. To get a contradiction, we proceed under the assumption that $\hat{\zeta}^g > \hat{Z}^g$. By the definition of $\hat{Z}^g$, we have $\hat{Z}^g \geq \sum_{j \in N} r_j \, P_j^g(x) = (\sum_{j \in N} r_j \, v_j^g \, x_j^g)/(1 + \sum_{j \in N} v_j^g \, x_j^g)$ for all $x^g \in \{0,1\}^{|N|}$. If we arrange the terms in this inequality, then it follows that $\hat{Z}^g \geq \sum_{j \in N} (r_j - \hat{Z}^g) \, v_j^g \, x_j^g$ for all $x^g \in \{0,1\}^{|N|}$, in which case, we obtain the chain of inequalities

$$\begin{aligned} \zeta^g > \hat{Z}^g \geq \max_{x^g \in \{0,1\}^{|N|}} \left\{ \sum_{j \in N} (r_j - \hat{Z}^g) \, v_j^g \, x_j^g \right\} &\geq \max_{x^g \in \{0,1\}^{|N|}} \left\{ \sum_{j \in N} (r_j - \hat{\zeta}^g) \, v_j^g \, x_j^g \right\} \\ &= \max_{x^g \in [0,1]^{|N|}} \left\{ \sum_{j \in N} (r_j - \hat{\zeta}^g) \, v_j^g \, x_j^g \right\} \geq \sum_{j \in N} (r_j - \hat{\zeta}^g) \, v_j^g \, \frac{\hat{y}_j^g}{\hat{w}^g}. \end{aligned}$$

The first and third inequalities above use the assumption that $\hat{\zeta}^g > \hat{Z}^g$. The equality above follows by noting that the objective function of the second optimization problem above is linear, in which case, the continuous relaxation of this problem has an integer optimal solution. To see that the fourth inequality above holds, we note that since $\hat{w}^g > 0$, the second set of constraints in problem (C.1) yield $\hat{y}_j^g / \hat{w}^g \in [0,1]$ for all $j \in N$ so that $\{\hat{y}_j^g / \hat{w}^g : j \in N\}$ is a feasible solution to the third optimization problem above. From the chain of inequalities above, we obtain $\sum_{j \in N} (r_j - \hat{\zeta}^g) \, v_j^g \, \hat{y}_j^g / \hat{w}^g < \hat{\zeta}^g$, which can equivalently be written as $\sum_{j \in N} r_j \, v_j^g \, \hat{y}_j^g < \hat{\zeta}^g \, (\hat{w}^g + \sum_{j \in N} v_j^g \, \hat{y}_j^g)$. By the definition of $\hat{\zeta}^g$, the left side of the last strict inequality is equal to $\hat{\zeta}^g$, but noting the first set of constraints in problem (C.1), we have $\hat{w}^g + \sum_{j \in N} v_j^g \, \hat{y}_j^g = 1$ and the right of this strict inequality is equal to $\hat{\zeta}^g$ as well, which is a contradiction. Thus, our claim holds and we have $\hat{\zeta}^g \leq \hat{Z}^g$. In this case, we obtain $\hat{\zeta} = \sum_{g \in G} \alpha^g \, \hat{\zeta}^g \leq \sum_{g \in G} \alpha^g \, \hat{Z}^g$.

Second, we show that $\hat{\zeta} \geq \sum_{g \in G} \alpha^g \, \hat{Z}^g$. Letting $\hat{x}^g = \{\hat{x}^g_j : j \in N\}$ be defined as in the statement of the proposition, we define the solution $\hat{y} = \{\hat{y}^g_j : j \in N, \; g \in G\}$, $\hat{w} = \{\hat{w}^g : g \in G\}$ and $\hat{z} = \{\hat{z}_j : j \in N\}$ to the linear programming relaxation of problem (C.1) as

$$\hat{y}^g_j = \frac{\hat{x}^g_j}{1 + \sum_{i \in N} v^g_i \, \hat{x}^g_i} \qquad \hat{w}^g = \frac{1}{1 + \sum_{j \in N} v^g_j \, \hat{x}^g_j} \qquad \hat{z}_j = \max_{g \in G} \left\{ \frac{\hat{x}^g_j}{1 + \sum_{i \in N} v^g_i \, \hat{x}^g_i} \right\}.$$

It is straightforward to see that the solution $(\hat{y}, \hat{w}, \hat{z})$ satisfies the first, second and third sets of constraints in problem (C.1). Since $\hat{P}^g = (\sum_{j \in N} v^g_j \, \hat{x}^g_j)/(1 + \sum_{j \in N} v^g_j \, \hat{x}^g_j) \geq 1/2$, subtracting one from both sides of this inequality, we obtain $1/(1 + \sum_{j \in N} v^g_j \, \hat{x}^g_j) \leq 1/2$, which implies that $\hat{y}^g_j \leq 1/2$, $\hat{w}^g \leq 1/2$ and $\hat{z}_j \leq 1/2$ for all $j \in N$, $g \in G$. Also, by the definition of $\hat{y}^g_j$ and $\hat{w}^g$, we have either $\hat{w}^g - \hat{y}^g_j = 0$ or $\hat{w}^g - \hat{y}^g_j = \hat{w}^g$, which happen respectively when $\hat{x}^g_j = 1$ and $\hat{x}^g_j = 0$. If we have $\hat{w}^g - \hat{y}^g_j = 0$, then the fourth set of constraints for this product j and customer type g is satisfied. If we have $\hat{w}^g - \hat{y}^g_j = \hat{w}^g$, then we obtain $\hat{w}^g - \hat{y}^g_j = \hat{w}^g \leq 1/2 = 1 - 1/2 \leq 1 - \hat{z}_j$, indicating that the fourth set of constraints for this product j and customer type g is satisfied as well. Thus, the solution $(\hat{y}, \hat{w}, \hat{z})$ is feasible to the linear programming relaxation of problem (C.1). So, the optimal objective value of the linear programming relaxation of problem (C.1) satisfies $\hat{\zeta} \geq \sum_{g \in G} \sum_{j \in N} \alpha^g \, r_j \, v^g_j \, \hat{y}^g_j = \sum_{g \in G} \alpha^g \, \sum_{j \in N} r_j \, v^g_j \, \hat{x}^g_j / (1 + \sum_{i \in N} v^g_i \, \hat{x}^g_i) = \sum_{g \in G} \sum_{j \in N} \alpha^g \, r_j \, P^g_j(\hat{x}^g) = \sum_{g \in G} \alpha^g \, \hat{Z}^g$, where the first inequality follows from the fact that $(\hat{y}, \hat{w}, \hat{z})$ is a feasible, but not necessarily an optimal solution to the linear programming relaxation of problem (C.1). $\square$

The first part of the proof of Proposition 24 does not use the assumption that $\hat{P}^g \geq 1/2$ for all $g \in G$. Therefore, the upper bound on the optimal expected revenue provided by the linear programming relaxation of problem (C.1) is always at least as tight as the upper bound that is obtained under the assumption that

we can offer different sets of products to different customer types. However, when we focus on each customer type individually, if customers of each type make a purchase with a probability exceeding 1/2, then the upper bound provided by the linear programming relaxation of problem (C.1) is equal to the upper bound that is obtained under the assumption that we can offer different sets of products to different customer types.

If we do not have $\hat{P}^g \geq 1/2$ for all $g \in G$, then we can give examples where the upper bound provided the linear programming relaxation of problem (C.1) is tighter than the upper bound obtained under the assumption that we can offer different sets of products to different customer types. Consider a problem instance with two products $N = \{1, 2\}$ and two customer types $G = \{1, 2\}$. The revenues of the products are $(r_1, r_2) = (95, 7)$. The preference weights of the two customer types are $(v_1^1, v_2^1) = (0.09, 0.09)$ and $(v_1^2, v_2^2) = (0, 0.01)$. The probabilities of observing the two customer types are $(\alpha^1, \alpha^2) = (0.5, 0.5)$. For this problem instance, the optimal objective value of the linear programming relaxation of problem (C.1) is about 3.92. On the other hand, if we assume that we can offer different sets of products to different customer types, then the upper bound that we obtain is about 3.96. For this problem instance, the solutions $\hat{x}^1 = (1, 0)$ and $\hat{x}^2 = (1, 1)$ maximize the expected revenue from each one of the two customer types when we focus on each one of the two customer types individually. When customers of each type are offered the solutions corresponding to them, they make a purchase respectively with probabilities $0.09/(1 + 0.09) \approx 0.08$ and $0.01/(1 + 0.01) \approx 0.01$. Since these probabilities of making a purchase are less than 1/2, this example violates the assumption of Proposition 24 and the upper bound on the optimal expected revenue provided by the linear programming relaxation of problem (C.1) can be tighter than the upper bound that is obtained under the assumption that we can

offer different sets of products to different customer types.

C.2 Online Appendix: Expected Revenue for Specially Structured Problem Instances

In this section, we show that the optimal expected revenue in the specially structured problem instance given in Table 4.3 is at most $(3+2\,\theta+\theta^2)/(1+\theta+\theta^2)$. We observe that the revenue associated with product three is θ^2, which is the largest one among all product revenues. Thus, it is trivially optimal to offer product three and we only consider the sets of products that include product three without loss of optimality. Table C.1 shows the expected revenue obtained from each customer type when we offer a particular set of products and an upper bound on the expected revenue over all customer types when we offer the particular set. To understand the figures in Table C.1, for example, if we offer the set of products $\{2,3\}$, then we obtain expected revenues of $\sum_{j\in\{2,3\}} r_j\, v_j^1/(1+\sum_{j\in\{2,3\}} v_j^1) = (\theta^4+\theta^5)/(1+\theta^2+\theta^4)$, $\sum_{j\in\{2,3\}} r_j\, v_j^2/(1+\sum_{j\in\{2,3\}} v_j^2) = \theta^5/(1+\theta^4)$ and zero respectively from customers of type one, two and three. These figures correspond to the entries in the second, third and fourth columns in Table C.1. Noting that $\theta \geq 1$, we have $(\theta^4+\theta^5)/(1+\theta^2+\theta^4) \leq 2\,\theta^5/\theta^4 = 2\,\theta$ and $\theta^5/(1+\theta^4) \leq \theta^5/\theta^4 = \theta$, in which case, if we offer the set of products $\{2,3\}$, then the expected revenue obtained over all customer types is at most

$$\frac{1}{1+\theta+\theta^2}\,2\,\theta + \frac{\theta}{1+\theta+\theta^2}\,\theta = \frac{2\,\theta+\theta^2}{1+\theta+\theta^2},$$

which corresponds to the entry in the last column in Table C.1. The other entries in Table C.1 are obtained in a similar fashion. We observe that each one of the

Offered Set of Prods.	Expected Revenue from Cus. Typ. 1	2	3	Upp. Bnd. on Exp. Rev.
{3}	$\frac{\theta^4}{1+\theta^2}$	0	0	$\frac{\theta^2}{1+\theta+\theta^2}$
{1,3}	$\frac{\theta^4+\theta^6}{1+\theta^2+\theta^6}$	$\frac{\theta^6}{1+\theta^6}$	$\frac{\theta^6}{1+\theta^6}$	$\frac{2+\theta+\theta^2}{1+\theta+\theta^2}$
{2,3}	$\frac{\theta^4+\theta^5}{1+\theta^2+\theta^4}$	$\frac{\theta^5}{1+\theta^4}$	0	$\frac{2\,\theta+\theta^2}{1+\theta+\theta^2}$
{1,2,3}	$\frac{\theta^4+\theta^5+\theta^6}{1+\theta^2+\theta^4+\theta^6}$	$\frac{\theta^5+\theta^6}{1+\theta^4+\theta^6}$	$\frac{\theta^6}{1+\theta^6}$	$\frac{3+2\,\theta+\theta^2}{1+\theta+\theta^2}$

Table C.1: Expected revenues obtained from different customer types and an upper bound on expected revenues obtained by offering different sets of products.

entries in the last column of Table C.1 is no larger than $(3+2\,\theta+\theta^2)/(1+\theta+\theta^2)$, which implies that the optimal expected revenue for the problem instance given in Table 4.3 is at most $(3+2\,\theta+\theta^2)/(1+\theta+\theta^2)$, as desired.

Bibliography

[1] J. Blanchet, G. Gallego, and V. Goyal. A Markov chain approximation to choice modeling. Technical report, Columbia University, New York, NY, 2013.

[2] 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.

[3] J. R. Correa, C. G. Fernandes, and Y. Wakabayashi. Approximating a class of combinatorial problems with rational objective function. *Mathematical Programming*, 124(1-2):255–269, 2010.

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

[5] A. Desir and V. Goyal. An FPTAS for capacity constrained assortment optimization. Technical report, Columbia University, School of Industrial Engineering and Operations Research, 2013.

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

[7] 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.

[8] 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.

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

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

[11] 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.

[12] L. Grigolon and F. Verboren. Nested logit or random coefficients logit? A comparison of alternative discrete choice models of product differentiation. *The Review of Economics and Statistics*, 96(5):916–935, 2014.

[13] S. Hashizume, M. Fukushima, N. Katoh, and T. Ibaraki. Approxiation algorithms for combinatorial fractional programming problems. *Mathematical Programming*, 37(3):255–267, 1987.

[14] S. Jagabathula. *Nonparametric Choice Modeling: Applications to Operations Management*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2008.

[15] S. Jagabathula and G. Vulcano. A model to estimate individual preferences using panel data. Technical report, New York University, Stern Business

School, 2015.

Available at http://pages.stern.nyu.edu/∼gvulcano/Research.html.

[16] Srikanth Jagabathula, Vivek Farias, and Devavrat Shah. Assortment optimization under general choice. In *INFORMS Conference, Charlotte, NC*, 2011.

[17] Anton Kleywegt and Xinchang Wang. A stochastic trust region algorithm for mixed logit type problems. In *International Conference on Stochastic Programming, Bergamo, Italy*, 2013.

[18] S. Kunnumkal, P. Rusmevichientong, and H. Topaloglu. Assortment optimization under the multinomial logit model with product costs. Technical report, Cornell University, School of Operations Research and Information Engineering, 2009.

[19] 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.

[20] Sumit Kunnumkal and Kalyan Talluri. A new compact linear programming formulation for choice network revenue management. Technical report, Universitat Pompeu Fabra, Barcelona, Spain, 2012.

[21] G. Li and P. Rusmevichientong. A greedy algorithm for assortment optimization in the two-level nested logit model. *Operations Research Letters*, 42(5):319–324, 2014.

[22] G. Li, P. Rusmevichientong, and H. Topaloglu. The d-level nested logit model: Assortment and price optimization problems. *Operations Research*, 62(2), 2015.

[23] 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.

[24] 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.

[25] Daniel McFadden. Conditional logit analysis of qualitative choice behavior. In P. Zarembka, editor, Frontiers in Economics, pages 105–142. Academic Press, 1974.

[26] Daniel McFadden and Kenneth Train. Mixed MNL models for discrete response. *Journal of Applied Economics*, 15:447–470, 2000.

[27] N. Meggido. Combinatorial optimization with rational objective functions. *Mathematics of Operations Research*, 4(4):414–424, 1979.

[28] 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.

[29] Joern Meissner, Arne Strauss, and Kalyan Talluri. An enhanced concave program relaxation for choice network revenue management. *Production and Operations Management*, 22(1):71–87, 2012.

[30] 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.

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

[32] M. L. Puterman. *Markov Decision Processes*. John Wiley and Sons, Inc., New York, 1994.

[33] 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.

[34] 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.

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

[36] A. Ruszczynski. *Nonlinear Optimization*. Princeton University Press, Princeton, New Jersey, 2006.

[37] R. D. s. *Individual Choice Behavior: A Theoretical Analysis*. Wiley, New York, NY, 1959.

[38] P. Sinha and A. A. Zoltners. The multiple-choice knapsack problem. *Operations Research*, 27(3):503–515, 1979.

[39] M. E. Slade. Merge-simulations of unilateral effects: What can we learn from the UK brewing industry? In B. Lyons, editor, Cases in European

Competition Policy: The Economic Analysis, Cambridge, 2009. Cambridge University Press.

[40] K. Talluri. A randomized concave programming method for choice network revenue management. Technical report, Universitat Pompeu Fabra, Barcelona, Spain, 2011.

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

[42] Kenneth Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, Cambridge, UK, 2003.

[43] 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.

[44] Thomas W. M. Vossen and Dan Zhang. Reductions of approximate linear programs for network revenue management. Technical report, University of Colorado at Boulder, Boulder, CO, 2013.

[45] G. Vulcano, G. J. van Ryzin, and R. Ratliff. Estimating primary demand for substitutable products from sales transaction data. *Operations Research*, 60(2):313–334, 2012.

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

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

[48] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, New York, 2011.

[49] D. Zhang and D. Adelman. An approximate dynamic programming approach to network revenue management with customer choice. *Transportation Science*, 42(3):381–394, 2009.

[50] D. Zhang and W. L. Cooper. Revenue management for parallel flights with customer choice behavior. *Operations Research*, 53(3):415–431, 2005.