Abstract

We study assortment optimization problems where customer choices are governed by the nested logit model and there are constraints on the assortment of products offered in each nest. There is a fixed revenue associated with each product. Each customer chooses a product within the offered assortment according to the nested logit model. The goal is to find an assortment of products to offer so as to maximize the expected revenue per customer subject to a cardinality, space or parent product constraint. Cardinality and space constraints, respectively, limit the number of products and the total space consumption of the products offered in each nest. Under parent product constraints, each product is either a parent product or a child product with a designated parent. A child product cannot be offered unless its parent product is offered. We show that the optimal assortment under cardinality or parent product constraints can be obtained in a tractable fashion by solving a linear program. The problem is NP-hard under space constraints and we give a 2-approximation algorithm for this case. We show that this approximation algorithm also provides a performance guarantee of \(1/(1 - \epsilon)\) when the space requirement for each product is at most a fraction \(\epsilon\) of the space availability in each nest. Furthermore, we develop a linear program to obtain an assortment with an arbitrary good performance guarantee under space constraints. The size of the linear program increases with the performance guarantee, but it allows us to tradeoff computational effort with performance guarantee. Finally, we demonstrate how our work can be used to solve joint assortment offering and pricing problems under the nested logit model.
Discrete choice models have long been used to describe how customers choose among an assortment of offered products that differ in attributes such as price and quality. Specifically, discrete choice models represent the demand for a particular product through the attributes of all offered products, capturing substitution possibilities and complementary relationships between the products. To pursue this thought, different discrete choice models have been proposed in the literature. Some of these models are based on axioms as in Luce (1959), resulting in the basic attraction model, whereas some others are based on random utility theory as in McFadden (1974), resulting in the multinomial logit model. A popular extension to the multinomial logit model is the nested logit model introduced by Williams (1977). Under the nested logit model, the products are organized in nests. The customer first selects a nest, and then a product within the selected nest. The nested logit model was developed primarily to avoid the independence of irrelevant alternatives property suffered by the multinomial logit model, which is exemplified by the red bus blue bus example that can be found in Ben-Akiva and Lerman (1994).

In this paper, we study constrained assortment optimization problems where the choices of the customers are governed by the nested logit model. There is a fixed revenue contribution associated with each product. The objective is to find an assortment of products to offer so as to maximize the expected revenue per customer subject to a constraint on the assortment offered in each nest. We consider three types of constraints, which we refer to as cardinality, space and parent product constraints. Cardinality and space constraints, respectively, limit the number of products and the total space consumption of the products in the assortment offered in each nest. Under parent product constraints, each product is designated either as a parent product or as a child product with a specific parent. A child product cannot be in the offered assortment unless its parent product is offered. Such assortment optimization problems are combinatorial and nonlinear in nature as the number of possible assortments can be large and the inclusion of a product in a nest has nonlinear externalities in the product selection probabilities both within and outside the nest.

We show that the assortment optimization problem under cardinality or parent product constraints can be solved to optimality through a linear programming formulation and the size of this formulation remains tractable as the number of products and the number of nests increase. On the other hand, if we have space constraints on the assortment offered in each nest, then the problem is NP-hard. We give a 2-approximation algorithm under space constraints. Beside the performance guarantee of two, this approximation algorithm also provides a performance guarantee of \(1/(1-\epsilon)\) when the space requirements for all products are at most a fraction \(\epsilon\) of the space availability in a nest. Thus, our approximation algorithm is guaranteed to perform better if each product, by itself, occupies a small fraction of the space availability. Our approximation algorithm is based on a tractable linear program. Furthermore, we show that if we want to obtain an \(\alpha\)-approximate solution under space constraints for an arbitrary value of \(\alpha\), then we can use a linear program with \(1+m\) decision variables and \(1+O(m[\alpha/(\alpha-1)]n^{[\alpha/(\alpha-1)]+2})\) constraints, where \(m\) is the number of nests and \(n\) is the number of products in the problem. This result is akin to polynomial time approximation schemes, where we tradeoff solution quality with computational effort. Finally,
we show how to use our approach to solve joint assortment offering and pricing problems, where
we choose the products to offer and their prices within a finite set of possible price levels. So,
our results have implications not only for assortment optimization, but also for pricing under the
nested logit model.

Assortment optimization is an active area of research, as assortment optimization problems find
important applications in retailing and revenue management. Our literature review covers papers
that model customer choices through attraction based choice models, such as the multinomial or
nested logit model. We refer the reader to Kok et al. (2008), Farias et al. (2011) and Farias et al.
(2012) for examples of assortment optimization problems under other discrete choice models. Davis
et al. (2011) classify the complexity of the assortment problem for nested attraction models along
two dimensions. The first dimension is the magnitude of the nest dissimilarity parameters that
characterize the degree of dissimilarity of the products within a nest. The second dimension is the
presence or absence of the no purchase alternative within a nest. Even if there are no constraints on
the offered assortment, the only polynomially solvable case is when the nest dissimilarity parameters
are all less than one and the no purchase alternative is only available at the time of selecting a
nest. Fortunately, this is the case that arises from a random utility model and corresponds to the
model studied by Williams (1977), McFadden (1980) and Borsch-Supan (1990). For the NP-hard
cases, Davis et al. (2011) provide heuristics with performance guarantees. Davis et al. (2011) do
not consider constraints on the offered assortment. Imposing such constraints is the main focus of
our paper, as the assortments that can be offered in practice are usually limited by a variety of
concerns and constraints can significantly complicate the assortment optimization problem.

We focus on cardinality, space and parent product constraints in this paper. There has been
some work on assortment optimization under cardinality and space constraints. Rusmeevichientong
et al. (2009) consider assortment optimization problems under the nested logit model with space
constraints. The crucial difference between our space constraints and theirs is that the space
constraints in Rusmeevichientong et al. (2009) limit the total amount of space consumed by all
offered products in all nests, whereas our space constraints separately limit the amount of space
consumed by the products offered in each nest. Both type of space constraints can have important
applications. In practice, nests represent different categories of products, different sales channels or
different retail stores. When nests represent different categories of products, it may be sensible
to limit the space consumption of all products in all categories, corresponding to the space
constraint in Rusmeevichientong et al. (2009). When nests represent different sales channels or
different retail stores, it may be sensible to limit the space consumption of the products offered
in each sales channel or in each retail store separately, corresponding to the space constraint in
this paper. Rusmeevichientong et al. (2009) give a polynomial time approximation scheme for the
assortment optimization problem under a space constraint covering all nests. The running time of
their algorithm increases exponentially with the number of nests and it would be difficult to apply
this algorithm when the number of nests exceeds two or three. In contrast, our 2-approximation
algorithm for space constraints is insensitive to the number of nests and can be applied when the

number of nests is large. On the other hand, under the assumption that the choices of the customers are governed by the multinomial logit model, Rusmevichientong, Shen and Shmoys (2010) consider assortment optimization problems with cardinality constraints on the offered assortment. They show that this assortment optimization problem is solvable in polynomial time. In this paper, we show that we can formulate a tractable linear program to solve the assortment optimization problem with cardinality constraints under the nested logit model. Since the multinomial logit model is a special case of the nested logit model with a single nest, our results in this paper are more general than the assortment optimization results in Rusmevichientong, Shen and Shmoys (2010).

There is also assortment work without constraints on the offered assortment. If the customers choose according to the multinomial logit model and there are no constraints on the offered assortment, then the assortment optimization problem can be solved efficiently, as it can be shown that the optimal assortment includes a certain number of products with the largest revenues. The problem can get complicated for other choice models. Rusmevichientong, Shmoys and Topaloglu (2010) study the assortment problem when there are multiple customer types and customers of different types choose according to different multinomial logit models. They show that the problem is NP-hard even with two customer classes. Mendez-Diaz et al. (2010) give a branch and cut algorithm to find the optimal assortment for the same problem and Meissner et al. (2012) give valid cuts. Rusmevichientong and Topaloglu (2011) study a robust assortment problem under the multinomial logit model when some of the parameters of the choice model are not known.

Capturing customer choice behavior in revenue management decisions has recently received attention. Talluri and van Ryzin (2004) consider a revenue management problem with a single flight leg. The customers choose among the fare classes available for purchase. The goal is to dynamically adjust the assortment of available fare classes to maximize the expected revenue. Gallego et al. (2004), Liu and van Ryzin (2008), Kunnumkal and Topaloglu (2008), Zhang and Adelman (2009), Bront et al. (2009) and Talluri (2011) extend the model in Talluri and van Ryzin (2004) to a flight network. The main idea in these papers is to develop a variety of linear programming approximations. The decision variables in these linear programs correspond to the number of time periods during which a particular subset of itinerary fare class combination is offered to customers. Noting that there is one decision variable for every possible subset of itinerary fare class combination, the number of decision variables is quite large. Therefore, the linear programs are usually solved by using column generation. The column generation subproblem for these linear programs exactly corresponds to the assortment optimization problem in this paper as long as the customer choices are dictated by the nested logit model.

The papers reviewed so far assume that the prices of the products are fixed and we are interested in choosing a set of products to offer so as to maximize the expected revenue per customer. Beside these models with fixed prices, research on pricing in the context of the multinomial and nested logit models has been active. In the pricing setting, the decision maker sets the prices of the products, where the prices of all products jointly determine the probability that a customer purchases a
particular product. The goal is to maximize the expected revenue per customer. For the pricing problem, Hanson and Martin (1996) observe that the expected revenue function is not concave in prices under the multinomial logit model, but Song and Xue (2007) and Dong et al. (2009) note that the expected revenue function remains concave in the market shares of the products. Li and Huh (2011) extend the concavity result to the nested logit model under the assumption that the price sensitivities of the products are constant within each nest and the nest dissimilarity parameters are all less than one. Gallego and Wang (2011) relax both assumptions in Li and Huh (2011). They show that the optimal prices add two terms to the unit costs, where the first term is the inverse of the price sensitivities of the products and the second term is a nest dependent constant. They find that optimal prices do not depend directly on quality, implying that products of different quality in a nest have the same markup as long as they have the same price sensitivity. Interestingly, we demonstrate that our results in this paper can be used to solve a joint assortment offering and pricing problem under the nested logit model. In this problem, we simultaneously decide which products to offer and choose the prices of the offered products within a discrete set. Chen and Hausman (2000) study a related product selection and pricing problem, but their results only apply to the multinomial logit model.

The paper is organized as follows. In Section 1, we formulate the constrained assortment optimization problem under the nested logit model and describe the cardinality, space and parent product constraints. In Section 2, we show how to use a linear program to find the best assortment to offer in each nest out of a given collection of candidate assortments. In Section 3, we give a general result that shows how to find a collection of candidate assortments for each nest such that if we restrict our attention to these candidate assortments, then we obtain an $\alpha$-approximate assortment for the constrained assortment optimization problem. An $\alpha$-approximate assortment means that the ratio of the optimal expected revenue to the expected revenue generated by the assortment is at most $\alpha \geq 1$. In Sections 4, 5 and 6, we leverage our general result in Section 3 respectively for cardinality, parent and space constraints. In Section 4, we show that we can obtain the optimal assortment under cardinality constraints by solving a linear program whose size increases polynomially with the numbers of nests and products. In Section 5, we show that we can solve a similar linear program to obtain the optimal assortment under parent product constraints. Thus, cardinality and parent product constraints constitute the cases where we can obtain the optimal assortment. In Section 6, we solve a similar type of linear program to obtain a 2-approximate assortment under space constraints. This section also provides a data dependent performance guarantee of $1/(1 - \epsilon)$ when the space requirement for each product is at most a fraction $\epsilon$ of the space capacity in its respective nest. In Section 7, we provide an approximation method under space constraints, where we obtain arbitrarily good assortments by increasing the computational effort. This approach is akin to a polynomial time approximation scheme. In Section 8, we give computational results under space constraints. In Section 9, we give directions for future research and describe extensions, including the joint assortment offering and pricing problem that we mention in the paragraph above.
1 Problem Formulation

In this section, we describe the nested logit model that we use to model the choice process of the customers and formulate the assortment optimization problem. There are \( m \) nests indexed by \( M = \{1, \ldots, m\} \). Depending on the application, each nest may correspond to a different category of products, a different sales channel or a different retail store. In each nest, there are \( n \) products that we can choose to offer to customers and we index the products by \( N = \{1, \ldots, n\} \). Under the nested logit model, an arriving customer decides either to make a purchase in one of the nests or to leave the system without purchasing anything. If the customer decides to make a purchase in one of the nests, then the customer must choose one of the products offered in this nest. We let \( v_{ij} \) be the preference weight of product \( j \) in nest \( i \) and use \( V_i(S_i) \) to denote the total preference weight of all offered products in nest \( i \) when we offer the assortment \( S_i \subset N \) of products in this nest. More precisely, \( V_i(S_i) = \sum_{j \in S_i} v_{ij} \). Under the nested logit model, given that a customer decides to make a purchase in nest \( i \), if the assortment \( S_i \) is offered in this nest, then the probability that the customer chooses product \( j \in S_i \) is \( v_{ij}/V_i(S_i) \). We let \( r_{ij} \) be the profit contribution of product \( j \) in nest \( i \). Following the revenue management tradition, we refer to \( r_{ij} \) as the revenue of product \( j \) in nest \( i \), which is sensible when the marginal cost is negligible or the profit contribution \( r_{ij} \) is the unit revenue net of marginal cost. If assortment \( S_i \) is offered in nest \( i \) and a customer decides to make a purchase in this nest, then the expected revenue is given by

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

If \( S_i = \emptyset \), then \( R_i(S_i) = 0 \) and we assume that \( 0/0 = 0 \) in the expression above. Our notation thus far implies that the number of possible products is the same in each nest, but this assumption is only for notational brevity and all of our results in the paper continue to hold when each nest includes a different number of products.

The preference weight of not making a purchase is \( v_0 \). Each nest \( i \) has a dissimilarity parameter \( \gamma_i \) associated with it, characterizing the degree of dissimilarity of the products in this nest. In this case, if we offer the assortment \( (S_1, \ldots, S_m) \) over all nests with \( S_i \subset N \) for all \( i \in M \), then the probability that a customer decides to make a purchase in nest \( i \) is given by

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

Depending on the interpretation of each nest as a product category, a sales channel or a retail store, the expression above determines the probability that a customer decides to make a purchase in a particular sales channel, a product category or a retail store as a function of the assortment offered over all nests. The dissimilarity parameter \( \gamma_i \) plays the role of dampening or magnifying the total preference weight of the products offered in nest \( i \). The standard form of the nested logit model assumes that the dissimilarity parameters satisfy \( \gamma_i \in (0,1] \) for all \( i \in M \), in which case, it is possible to show that the nested logit model arises from a random utility based choice model;
see McFadden (1974). Throughout the paper, we assume that \(\gamma_i \in (0, 1]\) for all \(i \in M\). If we offer the assortment \((S_1, \ldots, S_m)\) over all nests, then the expected revenue obtained from each customer can be written as

\[
\Pi(S_1, \ldots, S_m) = \sum_{i \in M} Q_i(S_1, \ldots, S_m) R_i(S_i) = \frac{1}{v_0 + \sum_{i \in M} V_i(S_i)\gamma_i} \sum_{i \in M} V_i(S_i)\gamma_i R_i(S_i),
\]

where the second equality uses the definition of \(Q_i(S_1, \ldots, S_m)\). Our goal is to find an assortment \((S_1, \ldots, S_m)\) to offer in all nests so as to maximize the expected revenue per customer while making sure that the assortment offered in each nest satisfies a certain feasibility constraint.

We work with three types of feasibility constraints on the offered assortment in each nest. The first type of constraints limits the number of products in the assortment offered in each nest. Using \(c_i\) to denote the limit on the number of products offered in nest \(i\), the feasible assortments in nest \(i\) under the first type of constraints are given by \(C_i = \{S_i \subset N : |S_i| \leq c_i\}\). This type of constraints naturally occurs when each nest corresponds to a product category or a sales channel and we want to limit the number of products offered in each product category or sales channel. We refer to these constraints as the \textit{cardinality constraints}. In the second type of constraints, there is a space requirement for each product and each nest provides a certain amount of available space. The total space requirement of the products offered in a nest cannot exceed the space availability of the nest. Using \(w_{ij}\) to denote the space requirement of product \(j\) in nest \(i\) and \(c_i\) to denote the total space availability in nest \(i\), the feasible assortments in nest \(i\) under the second type of constraints can be written as \(C_i = \{S_i \subset N : \sum_{j \in S_i} w_{ij} \leq c_i\}\). We refer to the second type of constraints as \textit{space constraints}. For example, space constraints occur when each nest corresponds to a retail store and each retail store provides a certain amount of shelf space. We assume that \(w_{ij} \leq c_i\) for all \(i \in M, j \in N\), implying that it is feasible to offer each product by itself.

Finally, the third type of constraints designates certain products as parent products. Each parent product has a set of child products associated with it and if a parent product is not offered, then none of its child products can be offered. We use \(P_i \subset N\) to denote the set of parent products in nest \(i\) and \(C_{ij}\) to denote the set of child products of parent product \(j\) in nest \(i\). We assume that \(C_{ij} \cap C_{ik} = \emptyset\) for all distinct \(j, k \in P_i\) so that a child product has a single parent product. In this case, the feasible assortments in nest \(i\) are given by \(C_i = \{S_i \subset N : j \notin S_i \cap P_i \implies S_i \cap C_{ij} = \emptyset\}\), implying that if a parent product is not offered, then none of its child products can be offered. This type of constraints arises when company policy or law requires offering certain products before offering others. For example, a company may be required to offer the generic version of a drug before it can offer the brand name versions, in which case, the generic version acts as the parent product, the brand name versions are the child products of the generic version and if the generic version is not offered, then none of the brand name versions can be offered. Also, in situations where a company offers different versions of a service, it may make sense to offer an upscale version of the service only if a basic version is offered. For example, for a company offering various accounting services, it may be sensible to offer a basic service such as bookkeeping before offering more advanced services.
such as budgeting and profitability analysis. In this case, bookkeeping plays the role of a parent product to its child products budgeting and profitability analysis. We refer to the third type of constraints as parent product constraints.

Our goal is to find an assortment to offer over all nests so that we maximize the expected revenue obtained from each customer while the assortment offered in each nest satisfies the cardinality, space or parent product constraints. In other words, we want to solve the problem

$$Z^* = \max_{(S_1, ..., S_m) \in C_1 \times ... \times C_m} \Pi(S_1, ..., S_m),$$  \hspace{1cm} (1)$$

where the feasible assortments $(C_1, ..., C_m)$ may correspond to cardinality, space or parent product constraints. If the feasible assortments $(C_1, ..., C_m)$ correspond to cardinality constraints, then we show that problem (1) can be solved by using a linear program with $1 + m$ decision variables and $1 + O(mn^2)$ constraints. Similarly, if the feasible assortments $(C_1, ..., C_m)$ correspond to parent product constraints, then we show that the optimal assortment can be obtained by solving a linear program with $1 + m$ decision variables and $1 + O(mn)$ constraints. However, if the feasible assortments $(C_1, ..., C_m)$ correspond to space constraints, then Lemma 2.1 in Rusmevichientong et al. (2009) shows that problem (1) is NP-hard even when there is only one nest with $\gamma_1 = 1$, indicating that it is difficult to find the optimal assortment under space constraints. Under space constraints, we show that we can solve a linear program with $1 + m$ decision variables and $1 + O(mn)$ constraints to obtain an assortment whose expected revenue deviates from the optimal expected revenue by at most a factor of two. Furthermore, if the products do not consume too much of the capacity available in a nest in the sense that $w_{ij} \leq \epsilon c_i$ for all $i \in M$, $j \in N$ for some $\epsilon \in (0, 1)$, then the expected revenue from this assortment deviates from the optimal expected revenue by at most a factor of $1/(1 - \epsilon)$.

2 Combining Candidate Assortments

In this section, we answer a fundamental question that arises when constructing an algorithm to solve problem (1). Assume that we are given a collection of candidate assortments $\{A^i_t : t \in \mathcal{T}_i\}$ to offer in nest $i$. All of the assortments in the collection are feasible in the sense that $A^i_t \in C_i$ for all $t \in \mathcal{T}_i$ and the set of feasible assortments $C_i$ may correspond to cardinality, space or parent product constraints. For each nest $i$, we want to find an assortment $\hat{S}_i \in \{A^i_t : t \in \mathcal{T}_i\}$ such that the assortment $(\hat{S}_1, ..., \hat{S}_m)$ provides the largest expected revenue among all assortments of the form $(S_1, ..., S_m)$ with $S_i \in \{A^i_t : t \in \mathcal{T}_i\}$. In other words, we want to use the collections of candidate assortments $\{A^i_t : t \in \mathcal{T}_i\}$ for all $i \in M$ to stitch together the best assortment to offer over all nests. Finding an answer to this question by brute force is computationally difficult since there are $|\mathcal{T}_1| \times ... \times |\mathcal{T}_m|$ possible combinations of assortments to choose over all nests.

It turns out that we can solve a linear program to obtain the assortment that provides the largest expected revenue among all assortments of the form $(S_1, ..., S_m)$ with $S_i \in \{A^i_t : t \in \mathcal{T}_i\}$. The numbers of decision variables and constraints in this linear program increase linearly with the
number of nests and the number of assortments in the candidate collection, rendering the linear program practical to use even when the number of nests and the number of candidate assortments are large. To see this result, assume that we find a value of \( z \) that satisfies

\[
v_0 z = \sum_{i \in M} \max_{S_i \in \{A_i^t : t \in T_i\}} \left\{ V_i(S_i)^\gamma_i \left( R_i(S_i) - z \right) \right\}.
\]

(2)

Using \( \hat{z} \) to denote the value of \( z \) that satisfies (2), we claim that \( \hat{z} \) is the largest expected revenue that can be obtained by using assortments of the form \((S_1, \ldots, S_m)\) with \( S_i \in \{A_i^t : t \in T_i\} \). To establish this claim, let \( \hat{S}_i \) be the optimal solution to the problem

\[
\max_{S_i \in \{A_i^t : t \in T_i\}} \left\{ V_i(S_i)^\gamma_i \left( R_i(S_i) - \hat{z} \right) \right\},
\]

(3)

Since \( \hat{z} \) is the value of \( z \) satisfying (2), we get \( v_0 \hat{z} = \sum_{i \in M} V_i(\hat{S}_i)^\gamma_i (R_i(\hat{S}_i) - \hat{z}) \). Solving for \( \hat{z} \) in the last expression yields \( \hat{z} = \sum_{i \in M} V_i(\hat{S}_i)^\gamma_i / (v_0 + \sum_{i \in M} V_i(\hat{S}_i)^\gamma_i) = \Pi(\hat{S}_1, \ldots, \hat{S}_m) \), where the last equality follows from the definition of \( \Pi(S_1, \ldots, S_m) \). Thus, there exists an assortment \((\hat{S}_1, \ldots, \hat{S}_m)\) with \( \hat{S}_i \in \{A_i^t : t \in T_i\} \) such that \( \hat{z} \) corresponds to the expected revenue provided by this assortment. On the other hand, for any assortment \((\hat{S}_1, \ldots, \hat{S}_m)\) with \( \hat{S}_i \in \{A_i^t : t \in T_i\} \), \( \hat{S}_i \) provides a feasible, but not necessarily the optimal, solution to the maximization problem on the right side of (2), in which case, we obtain \( v_0 \hat{z} \geq \sum_{i \in M} V_i(\hat{S}_i)^\gamma_i (R_i(\hat{S}_i) - \hat{z}) \). Solving for \( \hat{z} \) in the last inequality and using the definition of \( \Pi(S_1, \ldots, S_m) \) once more, we get \( \hat{z} \geq \Pi(\hat{S}_1, \ldots, \hat{S}_m) \). Thus, for any assortment \((\hat{S}_1, \ldots, \hat{S}_m)\) with \( \hat{S}_i \in \{A_i^t : t \in T_i\} \), \( \hat{z} \) is at least as large as the expected revenue provided by this assortment. These observations establish our claim and it follows that if the value of \( z \) that satisfies (2) is given by \( \hat{z} \), then \( \hat{z} \) is the largest expected revenue that can be obtained by using assortments of the form \((S_1, \ldots, S_m)\) with \( S_i \in \{A_i^t : t \in T_i\} \).

The discussion in the paragraph above shows that if we use \( \hat{z} \) to denote the value of \( z \) satisfying (2) and let \( \hat{S}_i \) be the optimal solution to problem (3), then the assortment \((\hat{S}_1, \ldots, \hat{S}_m)\) is the best assortment among all assortments of the form \((S_1, \ldots, S_m)\) with \( S_i \in \{A_i^t : t \in T_i\} \). This result provides an answer to the question that we ask at the beginning of this section. One remaining question, however, is how we can find a value of \( z \) satisfying (2). Noting that the left side of (2) is strictly increasing in \( z \) and the right side is decreasing in \( z \), there always exists a unique value of \( z \) satisfying (2). To construct a simple approach for obtaining the value of \( z \) that satisfies (2), we observe that this value of \( z \) corresponds to the optimal objective value of the optimization problem \( \min\{z : v_0 z \geq \sum_{i \in M} \max_{S_i \in \{A_i^t : t \in T_i\}} \{V_i(S_i)^\gamma_i (R_i(S_i) - z)\}\} \). To linearize the constraint in this problem, we define the decision variables \( y = (y_1, \ldots, y_m) \) as \( y_i = \max_{S_i \in \{A_i^t : t \in T_i\}} \{V_i(S_i)^\gamma_i (R_i(S_i) - z)\} \) and write the last optimization problem as

\[
\min \left\{ z : v_0 z \geq \sum_{i \in M} y_i, \quad y_i \geq V_i(S_i)^\gamma_i (R_i(S_i) - z) \quad \forall S_i \in \{A_i^t : t \in T_i\}, \quad i \in M \right\},
\]

(4)

where the decision variables are \( (z, y) \). The problem above is a linear program with \( 1 + m \) decision variables and \( 1 + \sum_{i \in M} |T_i| \) constraints, which is tractable as long as the number of candidate assortments is not too large. The next theorem collects our observations in this section.
Theorem 1 Let \( \hat{z} \) be the optimal objective value of the linear program in (4) and \( \hat{S}_i \) be the optimal solution to problem (3). Then, the assortment \((\hat{S}_1, \ldots, \hat{S}_m)\) provides the largest expected revenue among all assortments of the form \((S_1, \ldots, S_m)\) with \(S_i \in \{A^i_t : t \in T_i\}\).

If we are given a collection of candidate assortments \(\{A^i_t : t \in T_i\}\) for all \(i \in M\), then Theorem 1 provides a systematic approach for stitching together the best assortment over all nests. All we need to do is to solve the linear program in (4) to obtain its optimal objective value. Using \( \hat{z} \) to denote the optimal objective value of this linear program and \( \hat{S}_i \) to denote the optimal solution to problem (3), the assortment \((\hat{S}_1, \ldots, \hat{S}_m)\) is the best assortment we can obtain by using the collection of candidate assortments \(\{A^i_t : t \in T_i\}\) for nest \(i\). In the next section, we consider the question of how we can come up with a good collection of candidate assortments.

3 Obtaining Candidate Assortments

In this section, we show how to obtain a collection of candidate assortments for each nest so that the best assortment that we can stitch together by focusing on these candidate assortments provides a certain performance guarantee. In the next lemma, we begin by giving a characterization of good assortments for each nest.

Lemma 2 Let \( Z^* \) be the optimal objective value of problem (1). If the assortments \( \hat{S}_i, i \in M \) satisfy

\[
\alpha V_i(\hat{S}_i)^\gamma_i (R_i(\hat{S}_i) - Z^*) \geq \max_{\hat{S}_i \in C_i} \left\{ V_i(S_i)^\gamma_i (R_i(S_i) - Z^*) \right\} \tag{5}
\]

for some \( \alpha \geq 1 \), then \( \alpha \Pi(\hat{S}_1, \ldots, \hat{S}_m) \geq Z^* \).

Proof. Letting \((S_1^*, \ldots, S_m^*)\) be the optimal solution to problem (1) providing the expected revenue \( Z^* \), the definition of \( \Pi(S_1, \ldots, S_m) \) yields \( Z^* = \sum_{i \in M} V_i(S_i^*)^{\gamma_i} R_i(S_i^*)/(v_0 + \sum_{i \in M} V_i(S_i^*)^{\gamma_i}) \). If we arrange the terms in the last expression, then we get \( \sum_{i \in M} V_i(S_i^*)^{\gamma_i}(R_i(S_i^*) - Z^*) = v_0 Z^* \). On the other hand, if \( \hat{S}_i \) satisfies the inequality in (5) for all \( i \in M \), then evaluating the objective function on the right side of (5) at the feasible, but not necessarily the optimal, solution \( S_i^* \) and adding over \( i \in M \), we obtain

\[
\alpha \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i} (R_i(\hat{S}_i) - Z^*) \geq \sum_{i \in M} V_i(S_i^*)^{\gamma_i} (R_i(S_i^*) - Z^*) = v_0 Z^*.
\]

Since \( \alpha \geq 1 \) and \( Z^* \geq 0 \), the first and last expressions above yield \( \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i}(\alpha R_i(\hat{S}_i) - Z^*) \geq v_0 Z^* \). Solving for \( Z^* \) in this inequality, we obtain \( \alpha \Pi(\hat{S}_1, \ldots, \hat{S}_m) \geq Z^* \). \( \square \)

Lemma 2 implies that if \( \hat{S}_i \) is an \( \alpha \)-approximate solution to the maximization problem on the right side of (5) for all \( i \in M \), then the assortment \((\hat{S}_1, \ldots, \hat{S}_m)\) is an \( \alpha \)-approximate solution to
problem (1). This lemma is not immediately useful for obtaining good solutions to problem (1) since obtaining an \(\alpha\)-approximate solution to the maximization problem on the right side of (5) requires knowing the value of \(Z^*\). Furthermore, the presence of the exponent \(\gamma_i\) in \(V_i(S_i){}^{\gamma_i}\) may introduce some undesirable nonlinearity when we deal with maximization problems of the form that appear on the right side of (5).

In the next lemma, we build on Lemma 2 to give an alternative and key characterization of good assortments for each nest. The surprising feature of this lemma is that it eliminates the presence of the exponent \(\gamma_i\). The proof of the lemma crucially depends on the fact that \(u^{\gamma_i}\) is a concave function of \(u\) when \(\gamma_i \in (0, 1]\).

**Lemma 3** Let \((S_1^*, \ldots, S_m^*)\) be an optimal solution to problem (1) with the optimal objective value \(Z^*\). If the assortments \(\tilde{S}_i, i \in M\) satisfy

\[
\alpha V_i(\tilde{S}_i) (R_i(\tilde{S}_i) - \gamma_i Z^* - (1 - \gamma_i) R_i(S_i^*)) \geq \max_{\tilde{S}_i \in \mathcal{C}_i} \left\{ V_i(S_i) (R_i(S_i) - \gamma_i Z^* - (1 - \gamma_i) R_i(S_i^*)) \right\}
\]

for some \(\alpha \geq 1\), then there exists an assortment \((\tilde{S}_1^*, \ldots, \tilde{S}_m^*)\) with \(\tilde{S}_i = \tilde{S}_i\) or \(\tilde{S}_i = \emptyset\) such that \(\alpha \Pi(\tilde{S}_1^*, \ldots, \tilde{S}_m^*) \geq Z^*\).

**Proof.** For notational brevity, we let \(\hat{V}_i = V_i(\tilde{S}_i), V_i^* = V_i(S_i^*), \hat{R}_i = R_i(\tilde{S}_i)\) and \(R_i^* = R_i(S_i^*)\) throughout the proof. We partition the nests into two sets \(U = \{i \in M : R_i^* \geq Z^*\}\) and \(L = \{i \in M : R_i^* < Z^*\}\). We claim that \(\alpha \hat{V}_i^{\gamma_i}(\hat{R}_i - Z^*) \geq (V_i^*)^{\gamma_i}(R_i^* - Z^*)\) for all \(i \in U\). To see the claim, assume on the contrary that

\[
\alpha \hat{V}_i^{\gamma_i}(\hat{R}_i - Z^*) < (V_i^*)^{\gamma_i}(R_i^* - Z^*)
\]

for some \(i \in U\). Observe that we must have \(\hat{V}_i \neq 0\), since otherwise, the inequality above implies that \((V_i^*)^{\gamma_i}(R_i^* - Z^*) > 0\), indicating that \(V_i^* \neq 0\) and \(R_i^* > Z^*\). In this case, evaluating the objective function of the maximization problem on the right side of (6) with \(S_i = S_i^* \in \mathcal{C}_i\), it follows that \(\alpha \hat{V}_i (\hat{R}_i - \gamma_i Z^* - (1 - \gamma_i) R_i^*) \geq \gamma_i V_i^* (R_i^* - Z^*) > 0\), which cannot hold when \(\hat{V}_i = 0\). So, \(\hat{V}_i \neq 0\) and since \(\hat{V}_i \neq 0\), we can use (7) to upper bound \(\hat{R}_i\) as

\[
\hat{R}_i < \frac{1}{\alpha} \left( \frac{V_i^*}{\hat{V}_i} \right)^{\gamma_i} (R_i^* - Z^*) + Z^*.
\]

Using this upper bound on the left side of the inequality in (6), we get

\[
\alpha \hat{V}_i \left\{ \frac{1}{\alpha} \left( \frac{V_i^*}{\hat{V}_i} \right)^{\gamma_i} (R_i^* - Z^*) + Z^* - \gamma_i Z^* - (1 - \gamma_i) R_i^* \right\} > \max_{\tilde{S}_i \in \mathcal{C}_i} \left\{ V_i(S_i)(R_i(S_i) - \gamma_i Z^* - (1 - \gamma_i) R_i(S_i^*)) \right\} \geq \gamma_i V_i^* (R_i^* - Z^*),
\]

where the second inequality above follows by evaluating the objective function of the maximization problem above with \(S_i = S_i^* \in \mathcal{C}_i\). Focusing on the first and last expressions in the chain of
inequalities above, noting that \( \hat{V}_i \neq 0 \), dividing both sides of the inequality by \( \hat{V}_i \) and arranging the terms, we write this inequality as

\[
\left( \frac{V_i^*}{V_i} \right)^{\gamma_i} (R_i^* - Z^*) - \alpha (1 - \gamma_i) (R_i^* - Z^*) > \gamma_i \frac{V_i^*}{V_i} (R_i^* - Z^*).
\]

Since \( R_i^* \geq Z^* \) for \( i \in U \) by the definition of \( U \), canceling \( R_i^* - Z^* \) from both sides above and rearranging the terms, we obtain

\[
\left( \frac{V_i^*}{V_i} \right)^{\gamma_i} > \gamma_i \frac{V_i^*}{V_i} + \alpha (1 - \gamma_i).
\]

This inequality reads \( u^\gamma_i > \gamma_i u + \alpha (1 - \gamma_i) \) with \( u = V_i^*/\hat{V}_i \). Note that the function \( f(u) = u^\gamma_i \) is concave, so it should satisfy the subgradient inequality at point 1, which is \( u^\gamma_i \leq 1 + \gamma_i (u - 1) = \gamma_i u + (1 - \gamma_i) \leq \gamma_i u + \alpha (1 - \gamma_i) \). Thus, it is impossible for the last displayed inequality to hold and we get a contradiction. This observation implies that our claim holds so that \( \alpha \hat{V}_i^\gamma_i (\hat{R}_i - Z^*) \geq (V_i^*)^\gamma_i (R_i^* - Z^*) \) for all \( i \in U \).

For all \( i \in L \), it follows that \( \alpha V_i^\gamma_i (R_i(\emptyset) - Z^*) = 0 \geq (V_i^*)^\gamma_i (R_i^* - Z^*) \) by the definition of \( L \). Furthermore, a simple lemma, given as Lemma 10 in Online Supplement A, shows that \( V_i(S_i^*)^\gamma_i (R_i(S_i^*) - Z^*) = \max_{S_i \in C_i} V_i(S_i)^\gamma_i (R_i(S_i) - Z^*) \) for all \( i \in M \). In this case, if we define the assortment \((\tilde{S}_1, \ldots, \tilde{S}_m)\) such that \( \tilde{S}_i = \hat{S}_i \) for \( i \in U \) and \( \tilde{S}_i = \emptyset \) for \( i \in L \), then we obtain

\[
\alpha V_i(\tilde{S}_i)^\gamma_i (R_i(\tilde{S}_i) - Z^*) \geq V_i(S_i^*)^\gamma_i (R_i(S_i^*) - Z^*) = \max_{S_i \in C_i} V_i(S_i)^\gamma_i (R_i(S_i) - Z^*)
\]

for all \( i \in M \). The first inequality above follows for all \( i \in U \) by the claim that we establish at the beginning of the proof and it follows for all \( i \in L \) by the argument that we give at the beginning of this paragraph. The first and last expressions in the inequality above show that the assortment \((\tilde{S}_1, \ldots, \tilde{S}_m)\) satisfies the assumptions of Lemma 2. So, \( \alpha \Pi(\tilde{S}_1, \ldots, \tilde{S}_m) \geq Z^* \). \( \square \)

Lemma 3 indicates that if we use \( \hat{S}_i^\alpha \) to denote an \( \alpha \)-approximate solution to the maximization problem on the right side of (6), then we can use \( \{\hat{S}_i^\alpha, \emptyset\} \) as a collection of candidate assortments for nest \( i \). By this lemma, among all assortments of the form \((S_1, \ldots, S_m)\) with \( S_i \in \{\hat{S}_i^\alpha, \emptyset\} \), the expected revenue provided by the best assortment would deviate from the optimal expected revenue by no more than a factor of \( \alpha \). To find the best assortment among all assortments of the form \((S_1, \ldots, S_m)\) with \( S_i \in \{\hat{S}_i^\alpha, \emptyset\} \), we can make use of Theorem 1 with the collection of candidate assortments \( \{A_i^t : t \in T_i\} = \{\hat{S}_i^\alpha, \emptyset\} \). Noting that this collection includes two assortments for each nest, Theorem 1 shows that we can obtain the best assortment by solving a linear program with \( 1 + m \) decision variables and \( 1 + 2m \) constraints. Comparing Lemmas 2 and 3, we observe that Lemma 3 allows working with the maximization problem on the right side of (6) rather than the one on the right side of (5), but we need to consider the empty assortment also as a candidate assortment to be able to obtain a performance guarantee.

On the surface, Lemma 3 suffers from the main shortcoming of Lemma 2 in the sense that obtaining an \( \alpha \)-approximate solution to the maximization problem on the right side of (6) requires
knowing $Z^*$ and $S_i^\ast$, both of which are unknown. To resolve this difficulty, the important observation is that the quantity $\gamma_i Z^* + (1 - \gamma_i) R_i(S_i^\ast)$ in this maximization problem is a constant. So, for any $u \in \mathbb{R}_+$, we use $\hat{S}_i^u(u)$ to denote an $\alpha$-approximate solution to the problem

$$\max_{S_i \in \mathcal{S}_i} \{ V_i(S_i) (R_i(S_i) - u) \}. \quad (8)$$

In this case, the collection of assortments $\{\hat{S}_i^u(u) : u \in \mathbb{R}_+\}$ includes an $\alpha$-approximate solution to the maximization problem on the right side of (6), since problem (8) with $u = \gamma_i Z^* + (1 - \gamma_i) R_i(S_i^\ast)$ is identical to the maximization problem on the right side of (6). Also, we observe that if $u$ is large enough, then we have $R_i(S_i) - u \leq 0$ for all $S_i \subset N$ and the trivial optimal solution to problem (8) is the empty assortment giving an objective value of zero. In other words, the empty assortment is trivially included in the collection $\{\hat{S}_i^u(u) : u \in \mathbb{R}_+\}$. Therefore, the collection of assortments $\{\hat{S}_i^u(u) : u \in \mathbb{R}_+\}$ includes the collection $\{\hat{S}_i^u, \emptyset\}$, where $\hat{S}_i^u$ is as defined in the previous paragraph. This observation, in view of the discussion in the paragraph above, implies that if we use $\{\hat{S}_i^u(u) : u \in \mathbb{R}_+\}$ as the collection of candidate assortments for all $i \in M$, then the best assortment that we can stitch together from these collections provides an expected revenue that deviates from the optimal expected revenue by no more than a factor of $\alpha$.

To sum up the discussion thus far, assume that we can come up with a collection of assortments $\{A_i^t : t \in \mathcal{T}_i\}$ such that this collection always includes an $\alpha$-approximate solution to problem (8) for any $u \in \mathbb{R}_+$. In other words, $\{A_i^t : t \in \mathcal{T}_i\} \supset \{\hat{S}_i^u(u) : u \in \mathbb{R}_+\}$. In this case, the best assortment of the form $(\hat{S}_1, \ldots, \hat{S}_m)$ with $\hat{S}_i \in \{A_i^t : t \in \mathcal{T}_i\}$ provides an expected revenue that deviates from the optimal expected revenue for problem (1) by no more than a factor of $\alpha$. We record this observation in the next theorem.

**Theorem 4** Assume that the collection of assortments $\{A_i^t : t \in \mathcal{T}_i\}$ includes an $\alpha$-approximate solution to problem (8) for any $u \in \mathbb{R}_+$. Then, there exists an assortment $(\hat{S}_1, \ldots, \hat{S}_m)$ with $\hat{S}_i \in \{A_i^t : t \in \mathcal{T}_i\}$ such that $\alpha \Pi(\hat{S}_1, \ldots, \hat{S}_m) \geq Z^*$.

Theorems 1 and 4 play a key role to construct algorithms for solving problem (1). In particular, if we can come up with a reasonably small collection of assortments $\{A_i^t : t \in \mathcal{T}_i\}$ such that this collection includes an $\alpha$-approximate solution to problem (8) for any $u \in \mathbb{R}_+$, then Theorem 4 implies that the expected revenue from the best assortment $(\hat{S}_1, \ldots, \hat{S}_m)$ with $\hat{S}_i \in \{A_i^t : t \in \mathcal{T}_i\}$ deviates from the optimal expected revenue by at most a factor of $\alpha$. On the other hand, Theorem 1 implies that the best assortment $(\hat{S}_1, \ldots, \hat{S}_m)$ with $\hat{S}_i \in \{A_i^t : t \in \mathcal{T}_i\}$ can be obtained by solving a linear program with $1 + m$ decision variables and $1 + \sum_{i \in M} |\mathcal{T}_i|$ constraints. These observations show that if, for some $\alpha \geq 1$, we can come up with a reasonably small collection of assortments that includes an $\alpha$-approximate solution to problem (8) for any $u \in \mathbb{R}_+$, then we can solve a linear program with small numbers of decision variables and constraints to obtain an assortment whose expected revenue deviates from the optimal expected revenue by at most a factor of $\alpha$. This
discussion essentially reduces the job of obtaining good solutions to problem (1) to the job of obtaining good solutions to problem (8) for any \( u \in \mathbb{R}_+ \).

In general, it may not be possible to come up with a reasonably small collection of assortments \( \{A_i^t : t \in T_i\} \) that includes an \( \alpha \)-approximate solution to problem (8) for any \( u \in \mathbb{R}_+ \). In the next three sections, however, we show that if the feasible assortments are given by cardinality, parent product or space constraints, then we can indeed come up with a reasonably small collection of assortments that includes an \( \alpha \)-approximate solution to problem (8) for any \( u \in \mathbb{R}_+ \).

4 Cardinality Constraints

In this section, we consider the case where the feasible assortments are \( C_i = \{S_i \subset N : |S_i| \leq c_i\} \), which corresponds to constraining the cardinality of the assortment offered in nest \( i \) to \( c_i \). For this case, we show that we can come up with a collection of assortments \( \{A_i^t : t \in T_i\} \) such that this collection always includes an optimal, or 1-approximate, solution to problem (8) for any \( u \in \mathbb{R}_+ \). Furthermore, we show that the collection of assortments \( \{A_i^t : t \in T_i\} \) includes \( O(n^2) \) assortments and each one of the assortments in this collection can be identified in a tractable fashion. These results, in view of the discussion at the end of the previous section, imply that we can find the optimal solution to problem (1) under cardinality constraints simply by solving a linear program with \( 1 + m \) decision variables and \( 1 + O(mn^2) \) constraints.

To characterize the optimal solution to problem (8) for any \( u \in \mathbb{R}_+ \), noting the definitions of \( V_i(S_i) \) and \( R_i(S_i) \), we use the decision variables \( x_i = (x_{i1}, \ldots, x_{in}) \in \{0, 1\}^n \) to write problem (8) under cardinality constraints as

\[
\max \left\{ \sum_{j \in N} v_{ij} (r_{ij} - u) x_{ij} : \sum_{j \in N} x_{ij} \leq c_i, \ x_{ij} \in \{0, 1\} \ \forall \ j \in N \right\}.
\]

(9)

The problem above is a knapsack problem where the utility of product \( j \) is \( v_{ij} (r_{ij} - u) \) and each product consumes one unit of space. We can obtain an optimal solution to this knapsack problem by ordering the products with respect to their utilities and filling the knapsack starting from the product with the largest utility, as long as the utility of the product exceeds zero. Therefore, the optimal solution to problem (9) depends only on the ordering and signs of the utilities of the products. To exploit this observation, we define the linear functions \( f_{ij}(u) = v_{ij} (r_{ij} - u) \) for \( j \in N \) and \( f_{i0}(u) = 0 \), in which case, the function \( f_{ij}(u) \) corresponds to the utility of product \( j \) in the knapsack problem above. The \( n + 1 \) linear functions \( \{f_{ij}(\cdot) : j \in N \cup \{0\}\} \) intersect at \( O(n^2) \) points, which can be obtained by solving the equation \( f_{ij}(u) = f_{ik}(u) \) for \( u \) for all distinct \( j, k \in N \cup \{0\} \). We use \( \{\bar{u}_i^t : t \in T_i\} \) with \( |T_i| = O(n^2) \) to denote the set of intersection points that we obtain in this fashion. Since we are interested in the optimal solution to problem (9) for \( u \in \mathbb{R}_+ \), we drop the negative values from the set \( \{\bar{u}_i^t : t \in T_i\} \) and add the value zero into this set if it is not already included. In this case, the points in the set \( \{\bar{u}_i^t : t \in T_i\} \) partition the positive real line into \( |T_i| = O(n^2) \) intervals. We denote these intervals by \( \{I_i^t : t \in T_i\} \). In Figure 1, we show the
with

Figure 1: The linear functions \( \{f_{ij}(\cdot) : j \in N \cup \{0\}\} \), the points \( \{\hat{u}_i^t : t \in T_i\} \) and the intervals \( \{T_i^t : t \in T_i\} \) for a possible case with \( N = \{1, 2, 3\} \).

linear functions \( \{f_{ij}(\cdot) : j \in N \cup \{0\}\} \), the points \( \{\hat{u}_i^t : t \in T_i\} \) and the intervals \( \{T_i^t : t \in T_i\} \) for a possible case with \( N = \{1, 2, 3\} \). The solid lines show the functions \( \{f_{ij}(\cdot) : j \in N \cup \{0\}\} \), the circles on the horizontal axis show the points \( \{\hat{u}_i^t : t \in T_i\} \) and the braces on the horizontal axis show the intervals \( \{T_i^t : t \in T_i\} \). The elements of \( T_i \) are indexed by \( \{a, b, c, d, e, f, g\} \).

The key observation is that the ordering and signs of the utilities of the products in the knapsack problem above do not change as long as \( u \) takes values in one of the intervals \( \{T_i^t : t \in T_i\} \). Since the optimal solution to the knapsack problem depends only on the ordering and signs of the utilities of the products, the optimal solution does not change either as long as \( u \) takes values in one of these intervals. Therefore, by checking the ordering and signs of the utilities of the products in each one of the intervals \( \{T_i^t : t \in T_i\} \), we can come up with the optimal solution to the knapsack problem for any \( u \in \mathbb{R}_+ \). Since there are \( O(n^2) \) intervals in \( \{T_i^t : t \in T_i\} \), we can come up with a collection of assortments \( \{A_i^t : t \in T_i\} \) with \( |T_i| = O(n^2) \) such that this collection always includes an optimal solution to problem (8) for any \( u \in \mathbb{R}_+ \). The next theorem collects these observations.

**Theorem 5** Under cardinality constraints, there exists a collection of assortments \( \{A_i^t : t \in T_i\} \) with \( |T_i| = O(n^2) \) that includes an optimal solution to problem (8) for any \( u \in \mathbb{R}_+ \).

In this case, by Theorem 4, the best assortment \( (\hat{S}_1, \ldots, \hat{S}_m) \) with \( \hat{S}_i \in \{A_i^t : t \in T_i\} \) is an optimal solution to problem (1). By Theorem 1, on the other hand, we can find the best assortment \( (\hat{S}_1, \ldots, \hat{S}_m) \) with \( \hat{S}_i \in \{A_i^t : t \in T_i\} \) by solving a linear program with \( 1 + m \) decision variables and \( 1 + O(mn^2) \) constraints. An interesting side remark is that we can generate counterexamples to show that the optimal expected revenue is not necessarily a concave function of the capacities in the nests. So, additional units of capacity may not yield decreasing marginal returns.
5 Parent Product Constraints

In this section, we consider the case where the set of parent products in nest $i$ is $P_i$ and a parent product $j$ in nest $i$ has a set of child products $C_{ij}$. A child product cannot be offered unless its parent product is offered. If product $j$ is neither a parent product nor a child product, then we assume that product $j$ is a parent product with an empty set of child products. Also, we assume that $C_{ij} \cap C_{ik} = \emptyset$ for all distinct $j, k \in P_i$ so that the set of child products of two parent products are different. By the last two assumptions, the sets of products $P_i$ and $\{C_{ij} : j \in P_i\}$ collectively partition $N$. So, the feasible assortments are $C_i = \{S_i \subset N : j \notin S_i \cap P_i \Rightarrow S_i \cap C_{ij} = \emptyset\}$. For this case, we show that we can come up with a collection of assortments $\{A^i_t : t \in T\}$ that includes an optimal solution to problem (8) for any $u \in \mathbb{R}_+$. Furthermore, this collection of assortments includes $O(n)$ assortments and each one of the assortments in the collection can be identified in a tractable fashion. These results imply that we can solve a linear program with $1 + O(mn)$ constraints to obtain the optimal assortment under parent product constraints.

To characterize the optimal solution to problem (8) for any $u \in \mathbb{R}_+$, we use the decision variables $x_i = (x_{i1}, \ldots, x_{in}) \in \{0, 1\}^n$ to write problem (8) under parent product constraints as

$$\max \left\{ \sum_{j \in N} v_{ij} (r_{ij} - u) x_{ij} : x_{ik} \leq x_{ij} \forall j \in P_i, \ k \in C_{ij}, \ x_{ij} \in \{0, 1\} \forall j \in N \right\}. \quad (10)$$

The constraints in the problem above ensure that if a parent product is not offered, then we cannot offer any of its child products. Assume that parent product $j$ is offered in the optimal solution to problem (10). In this case, we are free to offer any of the child products of parent product $j$. Since it is optimal to offer one of these child products when their objective function coefficient is positive, if parent product $j$ is offered, then the total contribution of parent product $j$ and all of its child products to the objective function of the problem above is given by $f_{ij}(u) = v_{ij} (r_{ij} - u) + \sum_{k \in C_{ij}} v_{ik} |r_{ik} - u|^+$, where we use $[.]^+ = \max\{., 0\}$. On the other hand, if parent product $j$ is not offered, then none of its child products can be offered, in which case, parent product $j$ and all of its child products make a contribution of zero to the objective function of the problem above. Therefore, it is optimal to offer parent product $j$ as long as $f_{ij}(u) > 0$. The function $f_{ij}(\cdot)$ is decreasing and piecewise linear with points of nondifferentiability occurring at $\{r_{ik} : k \in C_{ij}\}$. Thus, we can find a value of $\bar{u}_{ij}$ such that $f_{ij}(u) > 0$ for any $u < \bar{u}_{ij}$ and $f_{ij}(u) \leq 0$ for any $u \geq \bar{u}_{ij}$, in which case, we offer parent product $j$ in the optimal solution to the problem above when $u < \bar{u}_{ij}$ and we do not offer parent product $j$ when $u \geq \bar{u}_{ij}$. If it is optimal not to offer parent product $j$, then none of its child products are offered, whereas if it is optimal to offer parent product $j$, then its child product $k$ is offered when $u < r_{ik}$. Therefore, by comparing the value of $u$ with $\bar{u}_{ij}$ and $\{r_{ik} : k \in C_{ij}\}$, we can decide whether it is optimal to offer parent product $j$ and any of its child products in the optimal solution to problem (10). Furthermore, it is straightforward to obtain the point $\bar{u}_{ij}$. Repeating the same reasoning for all of the parent products, we obtain the collections of points $\{\bar{u}_{ij} : j \in P_i\}$ and $\{r_{ik} : j \in P_i, \ k \in C_{ij}\}$. Since $P_i$ and $\{C_{ij} : j \in P_i\}$ partition $N$, there are a total of $n$ points in the collections $\{\bar{u}_{ij} : j \in P_i\}$ and $\{r_{ik} : j \in P_i, \ k \in C_{ij}\}$. These
points completely characterize the optimal solution to the problem above since we can compare \( u \) with \( \bar{u}_{ij} \) to decide whether it is optimal to offer parent product \( j \). If this is the case, then we can decide whether it is optimal to offer its child product \( k \) by comparing \( u \) with \( r_{ik} \).

Since there are a total of \( n \) points in the collections \( \{ \bar{u}_{ij} : j \in P_i \} \) and \( \{ r_{ik} : j \in P_i, k \in C_{ij} \} \), these points partition the positive real line into \( O(n) \) intervals and we denote these intervals by \( \{ I^t_i : t \in \mathcal{T}_i \} \) with \( |\mathcal{T}_i| = O(n) \). We observe that as long as \( u \) takes values in one of the intervals \( \{ I^t_i : t \in \mathcal{T}_i \} \), the ordering between \( u \) and any of the points in the collections \( \{ \bar{u}_{ij} : j \in P_i \} \) and \( \{ r_{ik} : j \in P_i, k \in C_{ij} \} \) does not change. This observation, in view of the discussion in the paragraph above, implies that the optimal solution to problem (10) does not change as long as \( u \) takes values in one of the intervals \( \{ I^t_i : t \in \mathcal{T}_i \} \). Thus, by comparing the value of \( u \) with \( \bar{u}_{ij} : j \in P_i \) and \( r_{ik} : j \in P_i, k \in C_{ij} \) in each one of the intervals \( \{ I^t_i : t \in \mathcal{T}_i \} \), we can come up with a collection of assortments \( \{ A^t_i : t \in \mathcal{T}_i \} \) with \( |\mathcal{T}_i| = O(n) \) such that this collection always includes an optimal solution to problem (8) for any \( u \in \mathbb{R}_+ \). The next theorem collects our observations.

**Theorem 6** Under parent product constraints, there exists a collection of assortments \( \{ A^t_i : t \in \mathcal{T}_i \} \) with \( |\mathcal{T}_i| = O(n) \) that includes an optimal solution to problem (8) for any \( u \in \mathbb{R}_+ \).

So, Theorem 4 with \( \alpha = 1 \) implies that the best assortment \( (\hat{S}_1, \ldots, \hat{S}_m) \) with \( \hat{S}_i \in \{ A^t_i : t \in \mathcal{T}_i \} \) is the optimal solution to problem (1). By Theorem 1, we can find this best assortment by solving a linear program with \( 1 + m \) decision variables and \( 1 + O(mn) \) constraints.

### 6 Space Constraints

In this section, we consider the case where the space consumption of the assortment offered in nest \( i \) is limited to \( c_i \) so that the feasible assortments are given by \( \mathcal{C}_i = \{ S_i \subseteq N : \sum_{j \in S_i} w_{ij} \leq c_i \} \). For this case, we show that we can come up with a collection of assortments \( \{ A^t_i : t \in \mathcal{T}_i \} \) that includes a 2-approximate solution to problem (8) for any \( u \in \mathbb{R}_+ \). Furthermore, this collection includes \( O(n^2) \) assortments. These results imply that we can solve a linear program with \( 1 + m \) decision variables and \( 1 + O(mn^2) \) constraints to obtain an assortment whose expected revenue deviates from the optimal expected revenue for problem (1) by at most a factor of two, when there are space constraints. The line of reasoning we follow in this section is similar to the one in Section 4, but we work with the linear programming relaxation of a knapsack problem. Using the decision variables \( x_i = (x_{i1}, \ldots, x_{in}) \in \{0, 1\}^n \), we write problem (8) under space constraints as

\[
\max \left\{ \sum_{j \in N} v_{ij} (r_{ij} - u) x_{ij} : \sum_{j \in N} w_{ij} x_{ij} \leq c_i, \ x_{ij} \in \{0, 1\} \forall j \in N \right\}. \tag{11}
\]

The problem above is a knapsack problem where the utility of product \( j \) is \( v_{ij} (r_{ij} - u) \) and the space consumption of product \( j \) is \( w_{ij} \). We can solve the linear programming relaxation of this knapsack problem by ordering the products with respect to their utility to space consumption ratios and
filling the knapsack starting from the product with the largest utility to space consumption ratio, as long as the utility of the product exceeds zero. Therefore, the optimal solution to the linear programming relaxation of the knapsack problem above does not change as long as the ordering and signs of the utility to space consumption ratios do not change. Also, it is useful to observe that the optimal solution to the linear programming relaxation obtained in this fashion includes at most one fractional decision variable.

Following an argument similar to the one in Section 4, we define the linear functions \( f_{ij}(u) = v_{ij}(r_{ij} - u)/w_{ij} \) for \( j \in N \) and \( f_{i0}(u) = 0 \), in which case \( f_{ij}(u) \) corresponds to the utility to space consumption ratio of product \( j \) in the knapsack problem in (11). Using \( \{ \bar{u}^g_i : g \in G_i \} \) to denote the set of intersection points of the linear functions \( \{ f_{ij}(\cdot) : j \in N \cup \{0\} \} \), dropping the negative intersection points and adding the value zero into the set of intersection points if necessary, the set of points \( \{ \bar{u}^g_i : g \in G_i \} \) partition the positive real line into \( |G_i| = O(n^2) \) intervals. Denoting these intervals by \( \{ T^g_i : g \in G_i \} \), the ordering and signs of the utility to space consumption ratios of the products do not change as long as \( u \) takes values in one of these intervals. Since the optimal solution to the linear programming relaxation of problem (11) depends only on the ordering and signs of the utility to space consumption ratios of the products, the optimal solution to the linear programming relaxation of the knapsack problem above does not change when \( u \) takes values in one of these intervals. We use \( x^g_i \) to denote the optimal solution to the linear programming relaxation of the knapsack problem in (11) when \( u \) takes values in the interval \( T^g_i \).

Using the solutions \( \{ x^g_i : g \in G_i \} \), we define the collection of assortments \( \{ S^g_i : g \in G_i \} \) such that \( S^g_i = \{ j \in N : x^g_{ij} = 1 \} \). That is, the assortment \( S^g_i \) includes the products that take value one in the solution \( x^g_i \). Augmenting the collection \( \{ S^g_i : g \in G_i \} \) with the singleton assortments \( \{ \{ j \} : j \in N \} \), we can show that the collection of assortments \( \{ S^g_i : g \in G_i \} \cup \{ \{ j \} : j \in N \} \) always includes a 2-approximate solution to the knapsack problem in (11) for any \( u \in \mathbb{R}_+ \). To see this result, assume that we solve the knapsack problem for some \( u \in \mathbb{R}_+ \). By the discussion in the paragraph above, if the point \( u \) lies in the interval \( T^g_i \), then the optimal solution to the linear programming relaxation of the knapsack problem is given by \( x^g_i \). In this case, using \( z^*(u) \) to denote the optimal objective value of the knapsack problem above and \( j^g \) to denote the fractional component of the solution \( x^g_i \) if there is one, we obtain

\[
 z^*(u) \leq \sum_{j \in N} v_{ij} (r_{ij} - u) x^g_{ij} \leq \sum_{j \in S^g_i} v_{ij} (r_{ij} - u) + v_{ij^g} (r_{ij^g} - u) 
\]

\[
 \leq 2 \max \left\{ \sum_{j \in S^g_i} v_{ij} (r_{ij} - u), v_{ij^g} (r_{ij^g} - u) \right\}, \quad (12)
\]

where the first inequality follows from the fact that the linear programming relaxation provides an upper bound on the optimal objective value of the knapsack problem and the second inequality follows from the fact that \( S^g_i \cup \{ j^g \} \) includes all components of the solution \( x^g_i \) that take strictly positive values. The chain of inequalities above imply that either one of the assortments \( S^g_i \) and \( \{ j^g \} \) is a 2-approximate solution to the knapsack problem in (11) and the desired result follows. Thus,
the intervals \( \{ T_i^u : g \in \mathcal{G}_i \} \) can be constructed by computing the intersection points of \( n + 1 \) linear functions. By checking the ordering and signs of the utility to space consumption ratios of the products in the intervals \( \{ T_i^u : g \in \mathcal{G}_i \} \), we can come up with all possible solutions \( \{ x_i^u : g \in \mathcal{G}_i \} \) to the linear programming relaxation of the knapsack problem for any \( u \in \mathbb{R}_+ \). Using these solutions, we can construct the assortments \( \{ S_i^g : g \in \mathcal{G}_i \} \). In this case, if we use \( \{ A_i^t : t \in T_i \} \) to denote the collection of assortments \( \{ S_i^g : g \in \mathcal{G}_i \} \cup \{ \{ j \} : j \in N \} \), then this collection of assortments satisfies \( |T_i| = O(n^2) \) and the chain of inequalities in (12) shows that this collection always includes a 2-approximate solution to problem (8) for any \( u \in \mathbb{R}_+ \).

It turns out that we can refine the approximation guarantee given above by making use of the problem data. In particular, we assume that each product consumes a bounded fraction of the capacity available in the nest, which is to say that \( w_{ij} \leq \epsilon c_i \) for some \( \epsilon \in (0, 1) \). In this case, it is possible to show that the collection of assortments \( \{ A_i^t : t \in T_i \} \) constructed by using the approach above includes a \( 1/(1 - \epsilon) \)-approximate solution to problem (9) for any \( u \in \mathbb{R}_+ \). Noting that \( 1/(1 - \epsilon) \to 1 \) as \( \epsilon \to 0 \), this result implies that the assortments obtained by our approach perform particularly well when each product, by itself, does not consume too much of the capacity available in the nest. To see that the collection of assortments \( \{ A_i^t : t \in T_i \} \) always includes a \( 1/(1 - \epsilon) \)-approximate solution to problem (8) for any \( u \in \mathbb{R}_+ \), assume that we solve the knapsack problem in (11) for some \( u \in \mathbb{R}_+ \). If the point \( u \) lies in the interval \( T_i^u \), then the optimal solution to the linear programming relaxation of the knapsack problem is given by \( x_i^u \). If the solution \( x_i^u \) does not have any fractional components, then the assortment \( S_i^g \in \{ A_i^t : t \in T_i \} \) is an optimal solution to the knapsack problem and the desired result follows immediately. Otherwise, if the solution \( x_i^u \) has a fractional component, then it must be the case that the capacity of the knapsack in problem (11) is totally consumed by the solution \( x_i^u \). Therefore, using \( j^9 \) to denote the fractional component of the solution \( x_i^u \) and noting the definition of \( S_i^g \), we obtain \( \sum_{j \in S_i^g} w_{ij} + w_{ij^9} \geq c_i \). On the other hand, noting that all of the products in the assortment \( S_i^g \) take value one in the optimal solution to the linear programming relaxation of the knapsack problem, but product \( j^9 \) takes a fractional value, the utility to space consumption ratios of the products in \( S_i^g \) must be larger than the utility to space consumption ratio of product \( j^9 \), which implies that \( \sum_{j \in S_i^g} v_{ij} (r_{ij} - u) = \sum_{j \in S_i^g} v_{ij^9} (r_{ij^9} - u) \geq v_{ij^9} (r_{ij^9} - u) / \sum_{j \in S_i^g} w_{ij} \). In this case, using \( z^*(u) \) to denote the optimal objective value of the knapsack problem in (11), we obtain

\[
z^*(u) \leq \sum_{j \in S_i^g} v_{ij} (r_{ij} - u) + v_{ij^9} (r_{ij^9} - u) = \left\{ 1 + \frac{v_{ij^9} (r_{ij^9} - u)}{\sum_{j \in S_i^g} v_{ij} (r_{ij} - u)} \right\} \sum_{j \in S_i^g} v_{ij} (r_{ij} - u) \\
\leq \left\{ 1 + \frac{w_{ij^9}}{\sum_{j \in S_i^g} w_{ij}} \right\} \sum_{j \in S_i^g} v_{ij} (r_{ij} - u) \leq \frac{1}{1 - \epsilon} \sum_{j \in S_i^g} v_{ij} (r_{ij} - u),
\]

where the first inequality is identical to the second inequality in (12), the second inequality uses the fact that \( \sum_{j \in S_i^g} w_{ij^9} (r_{ij^9} - u) \geq v_{ij^9} (r_{ij^9} - u) / \sum_{j \in S_i^g} w_{ij} \), the third inequality follows by noting that \( \sum_{j \in S_i^g} w_{ij} + w_{ij^9} \geq c_i \) and the fourth inequality is by the assumption that \( w_{ij} \leq \epsilon c_i \). The chain of inequalities above shows that \( S_i^g \) is a \( 1/(1 - \epsilon) \)-approximate solution to
problem (8). Thus, the collection of assortments \( \{A^t_i : t \in T_i\} \) as defined above always includes a \( 1/(1-\epsilon) \)-approximate solution to problem (8) for any \( u \in \mathbb{R}_+ \).

Putting the discussion so far in this section together, the collection of assortments \( \{A^t_i : t \in T_i\} \) can be constructed by checking the ordering and signs of the utility to space consumption ratios of the products in each one of the intervals \( \{I^g_i : g \in G_i\} \). The intervals \( \{I^g_i : g \in G_i\} \) can be obtained by finding the intersection points of \( n+1 \) linear functions. Also, the collection \( \{A^t_i : t \in T_i\} \) has \( O(n^2) \) assortments and it always includes a \( \min\{2, 1/(1-\epsilon)\} \)-approximate solution to problem (8) for any \( u \in \mathbb{R}_+ \). We record this result in the next theorem.

**Theorem 7** Under space constraints, there exists a collection of assortments \( \{A^t_i : t \in T_i\} \) with \( |T_i| = O(n^2) \) that includes a \( \min\{2, 1/(1-\epsilon)\} \)-approximate solution to problem (8) for any \( u \in \mathbb{R}_+ \).

Thus, by Theorem 4, the expected revenue from the best assortment \( \hat{S}_1, \ldots, \hat{S}_m \) with \( \hat{S}_i \in \{A^t_i : t \in T_i\} \) deviates from the optimal expected revenue by at most a factor of \( \min\{2, 1/(1-\epsilon)\} \). By Theorem 1, we can find the best assortment \( \hat{S}_1, \ldots, \hat{S}_m \) with \( \hat{S}_i \in \{A^t_i : t \in T_i\} \) by solving a linear program with \( 1+m \) decision variables and \( 1+O(mn^2) \) constraints.

### 7 Improving the Performance Guarantee under Space Constraints

In Section 6, we describe an approach to obtain a \( \min\{2, 1/(1-\epsilon)\} \)-approximate solution to problem (1) under space constraints. The smallest possible value of \( \epsilon \) that we can use in this performance guarantee is \( \bar{\epsilon} = \max\{w_{ij}/c_i : i \in M, j \in N\} \), indicating that the best performance guarantee from the approach described in Section 6 is given by \( \min\{2, 1/(1-\bar{\epsilon})\} \). In particular, even if we are willing to increase the computational effort, the approach described in Section 6 does not provide any guidance as to how we can improve this performance guarantee. In this section, our goal is to show how we can obtain better performance guarantees under space constraints as long as we are willing to increase the computational effort.

The starting point for the discussion in this section is problem (11), which is equivalent to problem (8) under space constraints. We recall that if we can come up with a collection of assortments \( \{A^t_i : t \in T_i\} \) such that this collection includes an \( \alpha \)-approximate solution to problem (11) for any \( u \in \mathbb{R}_+ \), then Theorem 4 implies that the best assortment \( \hat{S}_1, \ldots, \hat{S}_m \) with \( \hat{S}_i \in \{A^t_i : t \in T_i\} \) is an \( \alpha \)-approximate solution to problem (1). Furthermore, by Theorem 1, we can find this best assortment by solving a linear program with \( 1+m \) decision variables and \( 1+\sum_{i \in M} |T_i| \) constraints. In this section, using \( \lceil \cdot \rceil \) to denote the round up function, we show that if we are given any \( \alpha \geq 1 \), then we can come up with a collection of assortments \( \{A^t_i : t \in T_i\} \) with \( |T_i| = O(n^{\lceil \alpha/(1-\alpha) \rceil + 2}) \) such that this collection always includes an \( \alpha \)-approximate solution to problem (11) for any \( u \in \mathbb{R}_+ \). In this case, by Theorem 4, the best assortment \( \hat{S}_1, \ldots, \hat{S}_m \) with \( \hat{S}_i \in \{A^t_i : t \in T_i\} \) is an \( \alpha \)-approximate solution to problem (1). By Theorem 1, we can find this best assortment by solving a linear program with \( 1+m \) decision variables and \( 1+O(mn^{\lceil \alpha/(1-\alpha) \rceil + 2}) \)
To characterize approximate solutions to problem (11), we use a special linear programming relaxation to this problem. Letting \(1(\cdot)\) be the indicator function and using the decision variables \(x_i = (x_{i1}, \ldots, x_{in}) \in [0,1]^n\), for any given \(J \subset N\), we consider the problem

\[
\max \left\{ \sum_{j \in N} v_{ij} (r_{ij} - u) x_{ij} : \sum_{j \in N} w_{ij} x_{ij} \leq c_i, \quad x_{ij} = 1 \forall j \in J, \quad 0 \leq x_{ik} \leq 1 \left( v_{ik} (r_{ik} - u) \leq \min_{j \in J} \{ v_{ij} (r_{ij} - u) \} \right) \forall k \in N \setminus J \right\}. \tag{13}
\]

We can interpret the problem above as the linear programming relaxation of a knapsack problem, where the utility of product \(j\) is \(v_{ij} (r_{ij} - u)\) and the capacity consumption of product \(j\) is \(w_{ij}\). So, we can solve this problem by using the following procedure. We put all of the products in \(J\) into the knapsack and drop these products from consideration. We order the other products with respect to their utilities. If there are any products whose utilities exceed the smallest of the utilities of the products in \(J\), then we drop these products from consideration as well. Considering the remaining products, we fill the knapsack starting from the product with the largest utility to space consumption ratio, as long as the utility of the product exceeds zero. This procedure implies that the optimal solution to problem (13) does not change as long as the ordering of the utilities, ordering of the utility to space consumption ratios and signs of the utilities of the products do not change. Also, there is at most one fractional decision variable in the optimal solution to problem (13) obtained by using this procedure.

Similar to Section 6, we define the linear functions \(h_{ij}(u) = v_{ij} (r_{ij} - u)\) and \(f_{ij}(u) = v_{ij} (r_{ij} - u) / w_{ij}\) for \(j \in N\) and \(h_{i0}(u) = 0, f_{i0}(u) = 0\) so that \(h_{ij}(u)\) and \(f_{ij}(u)\) respectively capture the utility and utility to space consumption ratio of product \(j\). We use \(\{ \bar{w}_i^q : g \in H_i \}\) to denote the set of intersection points of the \(n + 1\) linear functions \(\{ h_{ij}(\cdot) : j \in N \cup \{ 0 \} \}\) and \(\{ \bar{w}_i^q : g \in H_i \}\) to denote the set of intersection points of the \(n + 1\) linear functions \(\{ f_{ij}(\cdot) : j \in N \cup \{ 0 \} \}\). As in Section 6, we have \(|H_i| = O(n^2)\). Furthermore, collecting the points \(\{ \bar{w}_i^q : g \in H_i \}\) and \(\{ \bar{w}_i^q : g \in H_i \}\) together, dropping the negative points and adding the value zero if necessary, the points \(\{ \bar{w}_i^q : g \in H_i \} \cup \{ \bar{w}_i^q : g \in H_i \}\) partition the positive real line into \(2|H_i| = O(n^2)\) intervals.
Theorem 9

Let \( \{ T_i^g : g \in \mathcal{G}_i \} \) with \( \mathcal{G}_i = O(n^2) \) to denote these intervals, in which case, the ordering of the utilities, ordering of the utility to space consumption ratios and signs of the utilities of the products in problem (13) do not change as long as \( u \) takes values in one of these intervals. Since the optimal solution to problem (13) depends only on the ordering of the utilities, ordering of the utility to space consumption ratios and signs of the utilities of the products, the optimal solution to problem (13) does not change either when \( u \) takes values in one of these intervals. We use \( x_i^q(J) \) to denote the optimal solution to problem (13) when \( u \) takes values in the interval \( T_i^g \). Our notation for \( x_i^q(J) \) captures that the optimal solution to problem (13) depends on the choice of \( J \).

Using the solution \( x_i^q(J) \), we define the assortment \( S_i^q(J) \) as \( S_i^q(J) = \{ j \in N : x_i^q(J)_{ij} = 1 \} \), which includes the products taking value one in the solution \( x_i^q(J) \). In this case, using \( \varphi_q \) to denote the set of subsets of \( N \) with cardinality not exceeding \( q \), we propose using the collection of assortments \( \{ S_i^q(J) : J \in \varphi_q, \; g \in \mathcal{G}_i \} \) as a collection of possibly good solutions to problem (11). Noting that \( |\varphi_q| = O(qn^q) \) and \( |\mathcal{G}_i| = O(n^2) \), there are \( O(qn^{q+2}) \) assortments in the collection \( \{ S_i^q(J) : J \in \varphi_q, \; g \in \mathcal{G}_i \} \), which can be manageable when \( q \) is not too large. The next lemma shows that this collection always includes a \( q/(q-1) \)-approximate solution to problem (11) for any \( u \in \mathbb{R}_+ \). We defer the proof of this lemma to Online Supplement A.

**Lemma 8** Letting \( S_i^q(J) \) be as defined above, the collection of assortments \( \{ S_i^q(J) : J \in \varphi_q, \; g \in \mathcal{G}_i \} \) includes a \( q/(q-1) \)-approximate solution to problem (11) for any \( u \in \mathbb{R}_+ \).

For any desired performance guarantee \( \alpha \geq 1 \), setting \( \alpha = q/(q-1) \) and solving for \( q \), we obtain \( q = \alpha/(\alpha-1) \). Thus, if we choose \( q = \lceil \alpha/(\alpha-1) \rceil \) in the lemma above, then the collection of assortments \( \{ S_i^q(J) : J \in \varphi_{\lceil \alpha/(\alpha-1) \rceil}, \; g \in \mathcal{G}_i \} \) includes an \( \alpha \)-approximate solution to problem (11) for any \( u \in \mathbb{R}_+ \). To come up with this collection of assortments, we compute the intervals \( \{ T_i^g : g \in \mathcal{G}_i \} \) by finding the intersection points of the linear functions \( \{ h_{ij}(\cdot) : j \in N \cup \{0\} \} \) and \( \{ f_{ij}(\cdot) : j \in N \cup \{0\} \} \). In this case, the ordering of the utilities, ordering of the utility to space consumption ratios and signs of the utilities of the products in problem (13) do not change when \( u \) takes values in one of the intervals \( \{ T_i^g : g \in \mathcal{G}_i \} \). Once these intervals are computed, we focus on each one of them one by one. For each interval \( T_i^g \) and for each \( J \in \varphi_{\lceil \alpha/(\alpha-1) \rceil} \), we solve problem (13) to get the optimal solution \( x_i^q(J) \) and define the assortment \( S_i^q(J) \) as above. Since \( |\varphi_{\lceil \alpha/(\alpha-1) \rceil}| = O(\lceil \alpha/(\alpha-1) \rceil \cdot n^{\lceil \alpha/(\alpha-1) \rceil}) \) and \( |\mathcal{G}_i| = O(n^2) \), there are \( O(\lceil \alpha/(\alpha-1) \rceil \cdot n^{\lceil \alpha/(\alpha-1) \rceil+2}) \) assortments in the collection \( \{ S_i^q(J) : J \in \varphi_{\lceil \alpha/(\alpha-1) \rceil}, \; g \in \mathcal{G}_i \} \). The next theorem collects our observations.

**Theorem 9** Under space constraints, for any \( \alpha \geq 1 \), there exists a collection of assortments \( \{ A_i^t \subset T_i \} \) with \( |T_i| = O(\lceil \alpha/(\alpha-1) \rceil \cdot n^{\lceil \alpha/(\alpha-1) \rceil+2}) \) such that this collection includes an \( \alpha \)-approximate solution to problem (8) for any \( u \in \mathbb{R}_+ \).

Thus, arguing as in the previous three sections, Theorem 4 implies that the best assortment \( (\hat{S}_1, \ldots, \hat{S}_m) \) with \( \hat{S}_i \in \{ A_i^t : t \in T_i \} \) provides a performance guarantee of \( \alpha \) for problem
Noting Theorem 1, this best assortment can be obtained by solving a linear program with $1 + m$ decision variables and $1 + O(m \alpha / (\alpha - 1)) n^{\alpha / (\alpha - 1) + 2}$ constraints. So, for any desired performance guarantee $\alpha \geq 1$, finding an assortment that provides this performance guarantee amounts to solving a linear program with $1 + m$ decision variables and $1 + O(m \alpha / (\alpha - 1)) n^{\alpha / (\alpha - 1) + 2}$ constraints. This result demonstrates how we can improve the performance guarantee by increasing the number of constraints in the linear program. This approach naturally becomes computationally intractable when $\alpha$ gets too close to one, but if, for example, we want a performance guarantee of $\alpha = 3/2$, then the number of constraints we need comes out to be $1 + O(mn^5)$.

The development in this section builds on Frieze and Clarke (1984), where the authors develop polynomial time approximation schemes for multi-dimensional knapsack problems. As is done in our approach, Frieze and Clarke (1984) enumerate a reasonably small number of subsets of products. For each subset of products, they fix the values of the decision variables corresponding to the products in the subset at one. If any of the remaining products has a utility that exceeds the minimum utility of the products in the subset, then they fix the value of the decision variable corresponding to this product at zero. After the values of some of the decision variables are fixed in this manner, they solve the linear programming relaxation of the multi-dimensional knapsack problem. However, the focus of Frieze and Clarke (1984) is on solving a single instance of a multi-dimensional knapsack problem. In contrast, we are interested in finding good solutions to problem (11) for all $u \in \mathbb{R}_+$. Beside Frieze and Clarke (1984), Sahni (1975) and Caprara et al. (2000) explore the idea of enumerating a reasonably small number of subsets of products to obtain polynomial time approximation schemes for knapsack problems.

8 Computational Experiments

Section 6 describes an approach that provides a $\min\{2, 1/(1 - \epsilon)\}$-approximate solution to problem (1) under space constraints. Throughout this section, we refer to this approach as CFP, standing for constant factor performance guarantee. On the other hand, Section 7 describes an alternative approach that can improve the performance guarantee as long as we are willing to increase the computational effort. We refer to this alternative approach as IMP, standing for improved performance guarantee. In this section, we give computational experiments that test the performance of the assortments obtained by CFP and IMP under space constraints. Noting that we can obtain the optimal assortment under cardinality and parent product constraints, we do not give computational experiments under these constraints.

8.1 Experimental Setup

The performance guarantee of $\min\{2, 1/(1 - \epsilon)\}$ for CFP can be assuring since this performance guarantee shows that CFP cannot perform arbitrarily badly, always obtaining at least half of the
optimal expected revenue. Similarly, IMP can provide a uniform performance guarantee by using the appropriate collection of assortments as indicated by Theorem 9. However, a guarantee of obtaining half of the optimal expected revenue may not be thoroughly satisfying from a practical perspective and a natural question is whether we can come up with a way to assess the performance of CFP and IMP for an individual problem instance. It turns out that we can solve a linear program to obtain an upper bound on the optimal expected revenue $Z^*$. In particular, it is possible to show that the optimal objective value of the linear program

$$\min \left \{ z : v_0 z \geq \sum_{i \in M} y_i, \ y_i \geq \left( \sum_{j \in N} v_{ij} x_{ij}^g \right)^\gamma \left\{ \frac{\sum_{j \in N} v_{ij} r_{ij} x_{ij}^g}{\sum_{j \in N} v_{ij} x_{ij}^g} - z \right \} \forall g \in G_i, \ i \in M \right \} \quad (14)$$

provides an upper bound on the optimal expected revenue $Z^*$ in problem (1) when there are space constraints on the offered assortment. We show this fact in Proposition 11 in Online Supplement A. In the linear program above, the decision variables are $(z, y)$. The quantity $x_{ij}^g$, as defined in Section 6, corresponds to the optimal solution to the linear programming relaxation of the knapsack problem in (11) when $u$ takes values in the interval $I_i^g$. So, we can get a feel for the optimality gap of the assortment obtained by CFP or IMP simply by comparing the expected revenue from this assortment with the upper bound on the optimal expected revenue provided by the optimal objective value of the linear program in (14).

In our computational experiments, we generate a large number of problem instances. For each problem instance, we compute the assortments obtained by CFP and IMP and solve the linear program in (14), in which case, we can bound the optimality gap of the assortments obtained by CFP and IMP. In all of our problem instances, the number of nests is $m = 5$. The number of products in each nest is either $n = 15$ or $n = 30$. To generate the dissimilarity parameter $\gamma_i$ of each nest $i$, we sample $\gamma_i$ from the uniform distribution over $[0.25, 0.75]$. To come up with the revenues and preference weights of the products, we generate $U_{ij}$ from the uniform distribution over $[0, 1]$ and generate $X_{ij}$ and $Y_{ij}$ from the uniform distribution over $[0.75, 1.25]$. In this case, we set the revenue of product $j$ in nest $i$ as $r_{ij} = 10 \times U_{ij}^2 \times X_{ij}$ and set the preference weight of product $j$ in nest $i$ as $v_{ij} = 10 \times (1 - U_{ij}) \times Y_{ij}$. The role of the parameter $U_{ij}$ is to introduce negative correlation between the revenues and the preference weights so that the more expensive products tend to have smaller preference weights. On the other hand, the role of the parameters $X_{ij}$ and $Y_{ij}$ is to introduce idiosyncratic noise in the revenues and the preference weights so that it is not always the case that expensive products have small preference weights. The exponent of two in $U_{ij}^2$ skews the distribution of the revenues so that we have a large number of products with small revenues, but a small number of products with large revenues. To generate the preference weight $v_0$ of the no purchase option, we calibrate this parameter so that a customer leaves without making any purchase with probability $P_0$ even when all of the products in all of the nests are offered. We use either $P_0 = 0.2$ or $P_0 = 0.4$ in our computational experiments.

To come up with the space consumptions of the products, we generate $w_{ij}$ from the uniform distribution over $[1, 10]$. We set the space availability $c_i$ of nest $i$ such that the space availability of
the nest is a fraction $\beta$ of the total space consumption of the products in this nest. In particular, we set the space availability of nest $i$ as $c_i = \beta \sum_{j \in N} w_{ij}$, but if $c_i$ comes out to be smaller than any of the space consumptions $\{w_{ij} : j \in N\}$ of the products in nest $i$, then we bump $c_i$ up to $\max\{w_{ij} : j \in N\}$. In this way, we ensure that each product fits into the space availability of its nest, satisfying $w_{ij} \leq c_i$ for all $i \in M, j \in N$. We use $\beta = 0.1$ or $\beta = 0.2$ or $\beta = 0.3$.

In our computational experiments, we use $n \in \{15, 30\}$, $P_0 \in \{0.2, 0.4\}$ and $\beta \in \{0.1, 0.2, 0.3\}$, yielding a total of 12 parameter combinations. For each parameter combination, we randomly generate 10,000 individual problem instances as described in the previous two paragraphs. For each problem instance, we compute the assortments obtained by CFP and IMP and we solve the linear program in (14) to obtain an upper bound on the optimal expected revenue.

### 8.2 Computational Results

Table 1 gives our computational results for CFP. The first column in this table shows the parameter combination of the test problems by using the triplet $(n, P_0, \beta)$. We recall that there are 10,000 problem instances generated in each parameter combination. For each problem instance, we compute the upper bound on the optimal expected revenue provided by the optimal objective value of problem (14). We let $\text{UB}^k$ be this upper bound for problem instance $k$. We use CFP to obtain an assortment with $\min\{2, 1/(1-\epsilon)\}$-approximation guarantee for each problem instance. For problem instance $k$, we let $\text{RCFP}^k$ be the expected revenue from the assortment obtained by CFP. The second column in Table 1 shows the average percent gap between $\text{UB}^k$ and $\text{RCFP}^k$ over the problem instances in a particular parameter combination. The third column shows the 95th percentile of the percent gaps between $\text{UB}^k$ and $\text{RCFP}^k$. In other words, the second and third columns respectively give the average and 95th percentile of the data $\{100 (\text{UB}^k - \text{RCFP}^k)/\text{UB}^k : k = 1, \ldots, 10,000\}$. The fourth column in Table 1 shows the number of problem instances where the percent gap between $\text{UB}^k$ and $\text{RCFP}^k$ is less than 1%. The interpretations of the fifth to ninth columns are similar to that of the fourth column, but these five columns respectively give the number of problem instances where the percent gap between $\text{UB}^k$ and $\text{RCFP}^k$ is less than 2%, 3%, 4%, 5% and 10%. The tenth column in Table 1 shows the average ratio between the space requirement of a product and the space availability of its nest, averaged over all products in all nests and over all problem instances in a parameter combination. If each item occupies a large fraction of the capacity availability in its nest, then the value of $\epsilon$ in the performance guarantee $\min\{2, 1/(1-\epsilon)\}$ becomes large and we expect the performance of CFP to deteriorate. Finally, the goal of the last column in Table 1 is to give a feel for the binding nature of the space constraints. In particular, this column shows the number of problem instances for which the unconstrained optimal solution to problem (1) violates the space constraint for at least one of the nests.

The results in Table 1 show that the assortments obtained by CFP perform quite well. The average optimality gap of these assortments is bounded by 2.12%. For nine out of 12 parameter combinations, the 95th percentile of the optimality gaps comes out to be smaller than 5%, which
in 109,571 out of 120,000 cases, the optimality gaps turn out to be less than 5%.

obtained by CFP than the performance guarantee of min \( \beta \). The linear program in (14) yields significantly tighter bounds on the optimality gaps of the assortments optimal expected revenue for more than 95% of the problem instances. We note that our use of the second to last column shows that each item occupies more than half of the space availability in a nest. Even in such a drastic case, CFP still performs within 10% of the upper bound on the optimality gap of the assortments provided by CFP is still 5.59% and the optimality gaps are no larger than 9.97% in more than 95% of the problem instances. For this parameter combination, the optimality gap of the assortments is on the large side. The most problematic parameter combination (15, 0.4, 0.1) corresponds to a small value of \( \beta \) with \( \beta = 0.1 \), but the average optimality gap of the assortments provided by CFP is still 5.59% and the optimality gaps are no larger than 9.97% in more than 95% of the problem instances. For this parameter combination, the second to last column shows that each item occupies more than half of the space availability in a nest on average, indicating that we would not be able to offer more than two or three products in each nest. Even in such a drastic case, CFP still performs within 10% of the upper bound on the optimal expected revenue for more than 95% of the problem instances. We note that our use of the linear program in (14) yields significantly tighter bounds on the optimality gaps of the assortments obtained by CFP than the performance guarantee of \( \min \{2, 1/(1 - \epsilon)\} \). Over all problem instances, in 109,571 out of 120,000 cases, the optimality gaps turn out to be less than 5%.

Table 2 gives our computational results for IMP. In our computational experiments, we ensure that IMP provides a performance guarantee of two by choosing \( \alpha = 2 \) in Theorem 9. In this case, IMP takes about five seconds to find an assortment for the test problems with \( n = 30 \) products. Given that we have 10,000 problem instances in each parameter combination, it is difficult to use a smaller value for \( \alpha \) in our computational experiments. However, if one is interested in obtaining a good solution for a single problem instance, then it may certainly be possible to work

<table>
<thead>
<tr>
<th>Param. Combin. ((n, P, \beta))</th>
<th>UB(^k) and RCFP(^k) % Gap</th>
<th>No. Probs. Given UB(^k) and RCFP(^k) % Gap</th>
<th>Avg. Cons. to. Cap. Rat.</th>
<th>No. of Constr. Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(15, 0.2, 0.3)</td>
<td>0.79</td>
<td>2.08</td>
<td>0.79</td>
<td>2.08</td>
</tr>
<tr>
<td>(15, 0.2, 0.2)</td>
<td>2.01</td>
<td>4.22</td>
<td>0.33</td>
<td>9.978</td>
</tr>
<tr>
<td>(15, 0.2, 0.1)</td>
<td>4.34</td>
<td>8.29</td>
<td>0.58</td>
<td>9.999</td>
</tr>
<tr>
<td>(15, 0.4, 0.3)</td>
<td>1.50</td>
<td>3.14</td>
<td>0.22</td>
<td>10.000</td>
</tr>
<tr>
<td>(15, 0.4, 0.2)</td>
<td>3.01</td>
<td>5.59</td>
<td>0.33</td>
<td>10.000</td>
</tr>
<tr>
<td>(15, 0.4, 0.1)</td>
<td>5.59</td>
<td>9.97</td>
<td>0.58</td>
<td>10.000</td>
</tr>
<tr>
<td>(30, 0.2, 0.3)</td>
<td>0.29</td>
<td>0.79</td>
<td>0.11</td>
<td>10.000</td>
</tr>
<tr>
<td>(30, 0.2, 0.2)</td>
<td>0.85</td>
<td>1.71</td>
<td>0.17</td>
<td>10.000</td>
</tr>
<tr>
<td>(30, 0.2, 0.1)</td>
<td>2.32</td>
<td>4.10</td>
<td>0.33</td>
<td>10.000</td>
</tr>
<tr>
<td>(30, 0.4, 0.3)</td>
<td>0.65</td>
<td>1.33</td>
<td>0.11</td>
<td>10.000</td>
</tr>
<tr>
<td>(30, 0.4, 0.2)</td>
<td>1.30</td>
<td>2.34</td>
<td>0.17</td>
<td>10.000</td>
</tr>
<tr>
<td>(30, 0.4, 0.1)</td>
<td>2.80</td>
<td>4.73</td>
<td>0.33</td>
<td>10.000</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>2.12</strong></td>
<td><strong>4.02</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Performance of CFP.

shows that the expected revenue loss from using CFP can be bounded by 5% for an overwhelming majority of our problem instances. The parameter combinations for which CFP provides the smallest optimality gaps correspond to the cases with \( \beta = 0.3 \). When the value of \( \beta \) is large, the space availability in each nest tends to be large and each product occupies a small fraction of the space available in a nest. This trend can be observed from the second to last column in Table 1, which shows that the average ratio between the space consumption of the products and the space availability of the nests is small when \( \beta = 0.3 \). For these cases, \( \epsilon \) in the performance guarantee of \( \min \{2, 1/(1 - \epsilon)\} \) comes out to be small and this performance guarantee becomes close to one. Thus, it is not too surprising that CFP performs well when \( \beta \) is on the large side. The most problematic parameter combination (15, 0.4, 0.1) corresponds to a small value of \( \beta \) with \( \beta = 0.1 \), but the average optimality gap of the assortments provided by CFP is still 5.59% and the optimality gaps are no larger than 9.97% in more than 95% of the problem instances. For this parameter combination, the second to last column shows that each item occupies more than half of the space availability in a nest on average, indicating that we would not be able to offer more than two or three products in each nest. Even in such a drastic case, CFP still performs within 10% of the upper bound on the optimal expected revenue for more than 95% of the problem instances. We note that our use of the linear program in (14) yields significantly tighter bounds on the optimality gaps of the assortments obtained by CFP than the performance guarantee of \( \min \{2, 1/(1 - \epsilon)\} \). Over all problem instances, in 109,571 out of 120,000 cases, the optimality gaps turn out to be less than 5%.
Table 2: Performance of IMP.

with a value of \( \alpha \) closer to one. The first column in Table 2 shows the parameter combination for our test problems. The second column is analogous to the second column in Table 1 and it shows the average percent gap between the upper bound on the optimal expected revenue and the expected revenue from the assortment obtained by IMP. In other words, if we use \( \text{RIMP}^k \) to denote the expected revenue from the assortment obtained by IMP for problem instance \( k \), then the second column shows the average of the data \( \{100 \left( \frac{\text{UB}^k - \text{RIMP}^k}{\text{UB}^k} \right) : k = 1, \ldots, 10,000\} \). The third column in Table 2 shows the 95th percentile of the percent gaps between the upper bound on the optimal expected revenue and the expected revenue from the assortments obtained by IMP. The fourth column shows the number of problem instances for which IMP gives better expected revenue than CFP. Finally, the fifth column shows the average percent gap between the expected revenues obtained by IMP and CFP, where the average is taken only over the problem instances for which IMP performs better than CFP.

Comparing the second and third columns in Table 2 with the second and third columns in Table 1, we observe that IMP improves the performance of CFP by a fraction of a percent. The 95th percentile of the optimality gaps for CFP and IMP can differ by a little over a percent in parameter combinations such as \((15, 0.2, 0.1)\) and \((15, 0.4, 0.1)\). For parameter combinations with \( \beta = 0.1 \), the fourth column in Table 2 indicates that IMP improves the performance of CFP in more than half of the problem instances. The average performance improvement we get over these problem instances can be slightly above a percent, as given in the fifth column. Similar to CFP, the 95th percentile of the optimality gaps assures us that the optimality gaps of the assortments obtained by IMP are no larger than 7-8% for an overwhelming majority of our problem instances. Also, although this statistic is not provided in Table 2, we note that IMP performs at least as well as CFP in all of our test problems. In fact, it is possible to check that for any assortment \( S_i^q \) as defined in Section 6, the assortment \( S_i^q(\emptyset) \) as defined in Section 7 provides an objective value for problem (8) that is at least as large as the one provided by \( S_i^q \). Similarly, for any singleton assortment \( \{j\} \), the assortment
$S^q_i(\{j\})$ provides an objective value for problem (8) that is at least as large as the one provided by $\{j\}$. Therefore, the collection of assortments $\{S^q_i(J) : J \in \varphi_q, \ g \in G_i\}$ used by IMP is at least as strong as the collection of assortments $\{S^q_i : g \in G_i\} \cup \{\{j\} : j \in N\}$ used by CFP even when $q = 1$. Thus, IMP would always perform at least as well as CFP.

Overall, our computational experiments indicate that CFP performs quite well, yielding less than 5% optimality gap for more than 109,000 out of 120,000 test problems. IMP provides small but consistent improvements over CFP. We note that the good performance of CFP and IMP in Tables 1 and 2 is not only due to the fact that these approaches find good assortments, but also due to the fact that the linear program in (14) provides good upper bounds on the optimal expected revenue. Even if we use an arbitrary heuristic to find a solution to problem (1), we can still use the linear program in (14) to obtain an upper bound on the optimal expected revenue and we can get a feel for the optimality gap of the heuristic by comparing this upper bound with the expected revenue from the assortment obtained by the heuristic. If the gap turns out to be small, then there is no need to look for better assortments. In this way, the linear program in (14) can be used as an efficient tool for checking the performance of any approximation method or heuristic.

9 Conclusions and Extensions

In this paper, we described an approach for solving assortment optimization problems under the nested logit model when we have constraints on the assortment offered in each nest. We showed that we can obtain the optimal assortment under cardinality or parent product constraints. For space constraints, we obtained approximate solutions with performance guarantees. We showed how to improve the performance guarantee as long as we are willing to increase the computational effort. Theorems 1 and 4 play a crucial role in developing our approach and it is important to emphasize the general nature of these results. In particular, both of these theorems are independent of the specification of the set of feasible assortments $C_i$ in nest $i$. Given a collection of feasible assortments for each nest, Theorem 1 gives a method to stitch together the best assortment over all nests. Theorem 4 shows that if we can come up with a reasonably small collection of assortments that includes an $\alpha$-approximate solution to problem (8) for any $u \in \mathbb{R}_+$, then we can use these assortments as a collection of candidate assortments to obtain an $\alpha$-approximate solution to the assortment optimization problem. In this way, the job of obtaining good solutions to problem (1) essentially reduces to the job of obtaining good solutions to problem (8) for any $u \in \mathbb{R}_+$.

Theorems 1 and 4 can be used to obtain good assortments under constraints other than the cardinality, parent product and space constraints that we consider in this paper. In this section, we make this point concrete by showing how to build on these theorems to solve a joint assortment offering and pricing problem, where we choose the products to offer and their prices. In Online Supplement B, we also show how to extend the ideas in this paper to deal with more involved capacity, space and parent product constraints. In particular, we give an approximation algorithm when we have both cardinality and space constraints on the offered assortment. Furthermore,
we show how to obtain the optimal assortment for a more general version of the parent product constraints. Specifically, we consider the case where each product is categorized as a parent, a child or a grandchild product. Each grandchild product has a child product serving as its ancestor and each child product has a parent product serving as its ancestor. A grandchild product cannot be offered unless its child product is offered and a child product cannot be offered unless its parent product is offered. We show that the optimal assortment can be obtained in a tractable fashion and this result extends the parent product constraints to arbitrary number of levels in the ancestor hierarchy. We hope that future research can yield other constraints under which we can obtain good assortments. At the end of this section, we finish with possible research directions.

9.1 Joint Assortment Offering and Pricing

We consider the case where there are \( p \) products in each nest indexed by \( P = \{1, \ldots, p\} \). Each product can be offered at \( b \) different price levels and we index the price levels by \( B = \{1, \ldots, b\} \). Our notation implies that the number of possible price levels for each product is the same, but relaxing this assumption is straightforward. The goal is to choose the products to offer and the prices of the offered products. This joint assortment offering and pricing problem can be formulated as an assortment optimization problem with constraints on the offered assortment. To that end, we create \( b \) copies of each product corresponding to different price levels. Thus, we have a total of \( n = pb \) product copies in each nest. We refer to each one of these product copies as a virtual product so that each virtual product corresponds to offering a certain product at a certain price level. We index the virtual products by \( N = \{1, \ldots, n\} \). We use \( N_l \) to denote the set of virtual products corresponding to product \( l \in P \). So, each one of the virtual products in \( N_l \) corresponds to offering product \( l \) at a certain price level and we have \( |N_l| = b \) and \( N_l \cap N_{l'} = \emptyset \) for all distinct \( l, l' \in P \). In this case, the joint assortment offering and pricing problem becomes that of finding a set of virtual products to offer subject to the constraint that we offer at most one virtual product corresponding to each product in \( P \). In other words, the feasible assortments in nest \( i \) are given by \( C_i = \{S_i \subset N : |S_i \cap N_l| \leq 1 \ \forall \ l \in P\} \), which ensures we offer at most one virtual product for each product \( l \in P \). Thus, if a product is offered, then it is offered at one price level. We use \( r_{ij} \) and \( v_{ij} \) to denote the revenue and preference weight associated with virtual product \( j \) in nest \( i \). Specifically, if virtual product \( j \) corresponds to offering product \( l \) at price level \( k \), then \( r_{ij} \) is the revenue corresponding to price level \( k \) for product \( l \) in nest \( i \) and \( v_{ij} \) is the preference weight associated with product \( l \) in nest \( i \) when this product is offered at price level \( k \). If a product is offered at a higher price level, then the preference weight associated with this product should naturally become smaller, but such an assumption is not required for our development.

Once we formulate the problem as an assortment optimization problem involving virtual products, we can follow an approach similar to the one in Section 4 to come up with a collection of assortments \( \{A^t_i : t \in T_i\} \) with \( |T_i| = O(bn) \) such that this collection includes an optimal solution to problem (8) for any \( u \in \mathbb{R}_+ \). In this case, we can obtain the optimal assortment by solving
a linear program with $1 + m$ decision variables and $1 + O(mb)$ constraints. To characterize the optimal solution to problem (8) with the definition of $C_i$ given above, we use the decision variables $x_i = (x_{i1}, \ldots, x_{in}) \in \{0, 1\}^n$ to write problem (8) as

$$
\max \left\{ \sum_{j \in N} v_{ij} (r_{ij} - u) x_{ij} : \sum_{j \in N_i} x_{ij} \leq 1 \\forall l \in P, \ x_{ij} \in \{0, 1\} \ \forall j \in N \right\}.
$$

(15)

The constraints above ensure that at most one virtual product is offered corresponding to any product in $P$. For any $l \in P$, we can find the optimal values of the decision variables $\{x_{ij} : j \in N_l\}$ in the problem above as follows. Noting the constraint $\sum_{j \in N_l} x_{ij} \leq 1$, among the decision variables $\{x_{ij} : j \in N_l\}$, we find the one with the largest objective function coefficient. If this objective function coefficient is positive, then we set the value of this decision variable to one and the values of the other decision variables in $\{x_{ij} : j \in N_l\}$ to zero. Otherwise, we set the values of all decision variables in $\{x_{ij} : j \in N_l\}$ to zero. Therefore, the optimal values of the decision variables $\{x_{ij} : j \in N_l\}$ in the problem above depends only on the ordering and signs of the objective function coefficients. In this case, defining the linear functions $f_{ij}(u) = v_{ij} (r_{ij} - u)$ for $j \in N$ and $f_{00}(u) = 0$, we can find the intersection points of $b + 1$ linear functions $\{f_{ij}(\cdot) : j \in N_l \cup \{0\}\}$. We use $\{\bar{u}_{il} : g \in G_{il}\}$ with $|G_{il}| = O(b^2)$ to denote the set of intersection points that we obtain in this fashion. Repeating the same argument for all $l \in P$, we obtain the points $\{\bar{u}^g_{il} : l \in P, \ g \in G_{il}\}$ and noting that $|G_{il}| = O(b^2)$ and $n = pb$, there are a total of $O(pb^2) = O(bn)$ points in the set $\{\bar{u}^g_{il} : l \in P, \ g \in G_{il}\}$. Thus, the points $\{\bar{u}^g_{il} : l \in P, \ g \in G_{il}\}$ partition the positive real line into $O(bn)$ intervals. We denote these intervals by $\{\mathcal{I}^g_{il} : t \in T_i\}$ with $|T_i| = O(bn)$. The key observation is that the ordering and signs of the objective function coefficients in problem (15) do not change as long as $u$ takes values in one of the intervals $\{\mathcal{I}^g_{il} : t \in T_i\}$. Thus, the optimal solution does not change either as long as $u$ takes values in one of these intervals. Therefore, by checking the ordering and signs of the objective function coefficients in each one of the intervals $\{\mathcal{I}^g_{il} : t \in T_i\}$, we can come up with the optimal solution to problem (15) for any $u \in \mathbb{R}_+$. Noting that there are $O(bn)$ intervals in $\{\mathcal{I}^g_{il} : t \in T_i\}$, we can come up with a collection of assortments $\{A^g_{il} : t \in T_i\}$ with $|T_i| = O(bn)$ such that this collection always includes an optimal solution to problem (8) for any $u \in \mathbb{R}_+$.

Besides choosing the prices of the offered products, small modifications in our development in this section would allow us to impose cardinality constraints on the assortment offered in each nest. In this case, we can come up with a collection of assortments $\{A^g_{il} : t \in T_i\}$ with $|T_i| = O(n^2)$ that always includes an optimal solution to problem (8) for any $u \in \mathbb{R}_+$.

9.2 Directions for Future Research

Our approach in this paper provides a systematic way to obtain good solutions to assortment optimization problems with constraints on the assortment offered in each nest. For many practical cases, each nest corresponds to a different product category, a different sales channel or a different retail store, in which case, it is sensible to separately put constraints on the assortment offered in each nest. In certain cases, however, we may be interested in constraining the cardinality or total
space consumption of all offered products in all nests and we end up with constraints linking the
different nests. An open question is to find tractable assortments with constant factor performance
guarantees when there are constraints linking multiple nests. Another item for future research
is extending our results to more general versions of the nested logit model. For example, we can
consider the case where a customer can still leave the system without purchasing anything even after
choosing a particular nest or where the dissimilarity parameters \((\gamma_1, \ldots, \gamma_m)\) can exceed one. In
either one of these cases, Davis et al. (2011) show that the assortment optimization problem is
NP-hard even without any constraints on the offered assortment. Thus, we may have to be content
with approximate solutions. Also, the proof of Lemma 3 depends on the fact that \(u_{\gamma_i}\) is concave
when \(\gamma_i \in (0, 1]\) and we need a new approach when the dissimilarity parameters exceed one.

References

Demand, MIT Press in Transportation Studies, Cambridge, MA.

Borsch-Supan, A. (1990), ‘On the compatibility of nested logit models with utility maximization’,


knapsack problems with cardinality constraints’, European Journal of Operational Research
123(2), 333–345.

optimal product line selection problem using choice-based conjoint analysis’, Management Science
46(2), 327–332.

Davis, J., Gallego, G. and Topaloglu, H. (2011), Assortment optimization under variants of the
nested logit model, Technical report, Cornell University, School of Operations Research and

Dong, L., Kouvelis, P. and Tian, Z. (2009), ‘Dynamic pricing and inventory control of substitute


Cambridge, MA.

Frieze, A. M. and Clarke, M. R. B. (1984), ‘Approximation algorithms for the \(m\)-dimensional 0-1
Research 15, 100–109.

University.

Gallego, G. and Wang, R. (2011), Multi-product price optimization and competition under the


A Online Supplement: Omitted Results

In this section, we give proofs for the omitted results.

A.1 Lemma 10

The next lemma is used in the proof of Lemma 3. The proof of this lemma follows from an argument similar to the one used in Section 2.

**Lemma 10** Let \( (S_1^*, \ldots, S_m^*) \) be an optimal solution to problem (1) with the optimal objective value \( Z^* \). Then, \( S_i^* \) satisfies \( V_i(S_i^*)^\gamma_i(R_i(S_i^*) - Z^*) = \max_{S_i \in C_i} \{ V_i(S_i)^\gamma_i(R_i(S_i) - Z) \} \) for all \( i \in M \).

**Proof.** Since \( (S_1^*, \ldots, S_m^*) \) is the optimal assortment, using the definition of \( \Pi(S_1, \ldots, S_m) \), we obtain \( Z^* = \Pi(S_1^*, \ldots, S_m^*) = \sum_{l \in M} V_l(S_l^*)^\gamma_l R_l(S_l^*)/(v_0 + \sum_{l \in M} V_l(S_l^*)^\gamma_l) \). Arranging the terms in the first and last expressions in this chain of equalities, we obtain

\[
v_0 Z^* = \sum_{l \in M} V_l(S_l^*)^\gamma_l (R_l(S_l^*) - Z^*). \tag{16}
\]

To get a contradiction, assume that there exists \( i \in M \) that satisfies the equality in the lemma as an inequality. So, letting \( \tilde{S}_i \) be the optimal solution to the maximization problem in the lemma for nest \( i \), we get \( V_i(S_i^*)^\gamma_l (R_l(S_i^*) - Z^*) < V_i(\tilde{S}_i)^\gamma_l (R_l(\tilde{S}_i) - Z^*) \). Considering the assortment \((\tilde{S}_1, \ldots, \tilde{S}_m)\) with \( \tilde{S}_l = S_l^* \) for all \( l \in M \setminus \{i\} \) and \( \tilde{S}_i = \tilde{S}_i \), the last inequality yields

\[
\sum_{l \in M} V_l(\tilde{S}_l)^\gamma_l (R_l(\tilde{S}_l) - Z^*) > \sum_{l \in M} V_l(S_l^*)^\gamma_l (R_l(S_l^*) - Z^*) = v_0 Z^*,
\]

where the equality follows from (16). Focusing on the first and last expressions in the chain of inequalities above, solving for \( Z^* \) and noting the definition of \( \Pi(S_1, \ldots, S_m) \), we obtain \( Z^* < \sum_{l \in M} V_l(\tilde{S}_l)^\gamma_l R_l(\tilde{S}_l)/(v_0 + \sum_{l \in M} V_l(\tilde{S}_l)^\gamma_l) = \Pi(\tilde{S}_1, \ldots, \tilde{S}_m) \). Thus, the assortment \((\tilde{S}_1, \ldots, \tilde{S}_m)\) provides a strictly larger expected revenue than \( Z^* \), which is a contradiction. \( \square \)

A.2 Lemma 8

This section gives a proof for Lemma 8. Fixing \( u \) at an arbitrary \( \hat{u} \in \mathbb{R}_+ \), we use \( x_i^* \) to denote the optimal solution to problem (11) when solved with \( u = \hat{u} \). We have the assortment \( S_i^* = \{j \in N : x_i^* = 1\} \) corresponding to this optimal solution. Throughout, we let \( \hat{g} \) be such that \( \hat{u} \) takes a value in the interval \( T_i^j \), where the intervals \( \{T_i^g : g \in G_i\} \) are as defined in Section 7. We begin by considering the case \( |S_i^*| \leq q \). If \( |S_i^*| \leq q \), then we have \( S_i^* \in \varphi_q \). In this case, noting that \( J \subset S_i^\varphi(J) \) by the definitions of \( x_i^\varphi(J) \) and \( S_i^\varphi(J) \), we have \( S_i^* \subset S_i^\varphi(J) \), which implies that \( \sum_{j \in S_i^*} v_{ij} (r_{ij} - \hat{u}) \leq \sum_{j \in S_i^\varphi(J)} v_{ij} (r_{ij} - \hat{u}) \). Thus, the assortment \( S_i^\varphi(J) \) provides a better objective value for problem (11) than the assortment \( S_i^* \) when this problem is solved with \( u = \hat{u} \). So, the assortment \( S_i^\varphi(J) \) is also optimal to problem (11) when solved with \( u = \hat{u} \). Also, since
In the rest of the proof, we assume that $|S^*_i| > q$. We let $J^*_i$ be the subset of $S^*_i$ that includes the $q$ elements of $S^*_i$ with the largest utilities in problem (11) when this problem is solved with $u = \hat{u}$. Since $|S^*_i| > q$, $J^*_i$ is well defined. Consider the optimal solution $x^i(J^*_i)$ to problem (13) when this problem is solved with $u = \hat{u}$ and $J = J^*_i$. If this solution has a fractional component $j'$, then the third set of constraints in problem (13) implies that product $j'$ satisfies $v_{ij'} (r_{ij'} - \hat{u}) \leq v_{ij} (r_{ij} - \hat{u})$ for all $j \in J^*_i$. Thus, $z^*(\hat{u}) = \sum_{j \in S^*_i} v_{ij} (r_{ij} - \hat{u}) = \sum_{j \in J^*_i} v_{ij} (r_{ij} - \hat{u}) + \sum_{j \in S^*_i \setminus J^*_i} v_{ij} (r_{ij} - \hat{u}) \geq q v_{ij'} (r_{ij'} - \hat{u})$, where the inequality follows by the fact that $|J^*_i| = q$ and $v_{ij'} (r_{ij'} - \hat{u}) \leq v_{ij} (r_{ij} - \hat{u})$ for all $j \in J^*_i$. Thus, the last chain of inequalities yields $v_{ij'} (r_{ij'} - \hat{u}) \leq z^*(\hat{u})/q$.

To finish the proof, let $z^*(u)$ and $\zeta^*(u, J)$ respectively be the optimal objective values of problems (11) and (13). We claim that $z^*(\hat{u}) \leq \zeta^*(\hat{u}, J^*_i)$. To see this claim, we note that $J^*_i$ includes the $q$ products with the largest utilities among the products taking value one in the optimal solution to problem (11) when this problem is solved with $u = \hat{u}$. The products in $S^*_i \setminus J^*_i$ also take value one in the optimal solution to problem (11), but by the definition of $J^*_i$, these products satisfy $v_{ik} (r_{ik} - \hat{u}) \leq \min_{j \in J^*_i} \{v_{ij} (r_{ij} - \hat{u})\}$ for all $k \in S^*_i \setminus J^*_i$. This inequality, together with the third set of constraints in problem (13), implies that if we solve problem (13) with $u = \hat{u}$ and $J = J^*_i$, then we fix the values of the decision variables corresponding to the products in $J^*_i$ at one, but the values of the decision variables corresponding to the products in $S^*_i \setminus J^*_i$ are free between zero and one. Furthermore, problem (13) does not require the decision variables to take binary values. Thus, the claim holds and $z^*(\hat{u}) \leq \zeta^*(\hat{u}, J^*_i)$. In this case, if we use, as above, $x^i(J^*_i)$ to denote the optimal solution to problem (13) when this problem is solved with $u = \hat{u}$ and $J = J^*_i$, then letting $j'$ be the fractional component of $x^i(J^*_i)$ when there is one, we obtain

$$z^*(\hat{u}) \leq \zeta^*(\hat{u}, J^*_i) = \sum_{j \in N} v_{ij} (r_{ij} - \hat{u}) x^i(J^*_i) \leq \sum_{j \in S^i(J^*_i)} v_{ij} (r_{ij} - \hat{u}) + \sum_{j \in S^i(J^*_i)} v_{ij} (r_{ij} - \hat{u}) + z^*(\hat{u})/q,$$

where the second inequality is by noting that $S^i(J^*_i)$ includes all strictly positive and integer valued components of $x^i(J^*_i)$ and the only possibly fractional component is $j'$ and the third inequality is by the fact that $v_{ij'} (r_{ij'} - \hat{u}) \leq z^*(\hat{u})/q$, which is shown in the paragraph above. Focusing on the first and last terms in the chain of inequalities above, we have $z^*(\hat{u}) \leq (q/(q-1)) \sum_{j \in S^i(J^*_i)} v_{ij} (r_{ij} - \hat{u})$, showing that the assortment $S^i(J^*_i)$ is a $q/(q-1)$-approximate solution to problem (11) when solved with $u = \hat{u}$. Since $|J^*_i| = q$, we have $S^i(J^*_i) \in \{S^i(J) : J \in \varphi_q, g \in G_i\}$. Noting that the choice of $\hat{u}$ is arbitrary, we conclude that the collection of assortments $\{S^i(J) : J \in \varphi_q, g \in G_i\}$ includes a $q/(q-1)$-approximate solution to problem (11) for any $u \in \mathbb{R}_+$. □
A.3 Proposition 11

Proposition 11 is used in Section 8 and it shows that the optimal objective value of problem (14) is an upper bound on the optimal expected revenue when we have space constraints.

**Proposition 11** If we use $\hat{z}$ to denote the optimal objective value of the linear program in (14), then we have $\hat{z} \geq Z^*$.

**Proof.** We let $(\hat{z}, \hat{y})$ be an optimal solution to problem (14) and $(S_1^*, \ldots, S_m^*)$ be an optimal solution to problem (1). We claim that $\hat{y}_i \geq V_i(S_i^*)\gamma_i(R_i(S_i^*) - \hat{z})$ for all $i \in M$. To establish this claim, we note that if $u$ is large enough, then the optimal solution to the linear programming relaxation of problem (11) is zero. Thus, letting $I_1$ and $x_i^\circ$ be as defined in Section 6, there exists an interval $I_1$ such that the solution $x_i^\circ$ to the linear programming relaxation of problem (11) is zero when $u$ takes values in this interval. In this case, the second set of constraints in problem (14) implies that $\hat{y}_i \geq 0$ for all $i \in M$. Therefore, if $S_i^* = \emptyset$ or $R_i(S_i^*) \leq \hat{z}$, then we have $V_i(S_i^*)\gamma_i(R_i(S_i^*) - \hat{z}) \leq 0 \leq \hat{y}_i$ and our claim trivially holds when $S_i^* = \emptyset$ or $R_i(S_i^*) \leq \hat{z}$. So, it is enough to establish our claim with $S_i^* \neq \emptyset$ and $R_i(S_i^*) > \hat{z}$.

We let $\hat{u} = \gamma_i \hat{z} + (1 - \gamma_i) R_i(S_i^*)$. Letting $g$ be the index of the interval $I_1$ that includes $\hat{u}$, by definition, $x_i^\circ$ is the optimal solution to the linear programming relaxation of problem (11) when we solve this problem with $u = \hat{u}$. Therefore, we have $\sum_{j \in N} v_{ij} (r_{ij} - \hat{u}) x_i^\circ \geq \sum_{j \in N} v_{ij} (r_{ij} - \hat{u})$, where we use the fact that $x_i^\circ$ is the optimal solution to the linear programming relaxation of problem (11) when we solve this problem with $u = \hat{u}$ but offering the products in $S_i^*$ provides a feasible, but not necessarily an optimal, solution to this problem. First, we assume that $x_i^\circ \neq 0$. In this case, slightly abusing the notation to let $V_i(x_i^\circ) = \sum_{j \in N} v_{ij} x_i^\circ$ and $R_i(x_i^\circ) = \sum_{j \in N} v_{ij} r_{ij} x_i^\circ / V_i(x_i^\circ)$, the last inequality can equivalently be written as $V_i(x_i^\circ) (R_i(x_i^\circ) - \hat{u}) \geq V_i(S_i^*) (R_i(S_i^*) - \hat{u})$.

Noting that $(\hat{z}, \hat{y})$ is a feasible solution to problem (14), this solution satisfies the second set of constraints in problem (14) for the index $g$ defined at the beginning of the previous paragraph, in which case, we can write this constraint in a compact fashion as $\hat{y}_i \geq V_i(x_i^\circ)\gamma_i(R_i(x_i^\circ) - \hat{z})$. Also, using the fact that $u^\gamma_i$ is concave function of $u$, the subgradient inequality yields $V_i(S_i^*)^\gamma_i \leq V_i(x_i^\circ)^{\gamma_i} + \gamma_i V_i(x_i^\circ)^{\gamma_i - 1} (V_i(S_i^*) - V_i(x_i^\circ)) = \gamma_i V_i(x_i^\circ)^{\gamma_i - 1} V_i(S_i^*) + (1 - \gamma_i) V_i(x_i^\circ)^{\gamma_i}$. Using these observations, we have the chain of inequalities

\[
\begin{align*}
\hat{y}_i &\geq V_i(x_i^\circ)^\gamma_i (R_i(x_i^\circ) - \hat{z}) = V_i(x_i^\circ)^\gamma_i (R_i(x_i^\circ) - \hat{u}) + (1 - \gamma_i) V_i(x_i^\circ)^\gamma_i (R_i(S_i^*) - \hat{z}) \\
&\geq V_i(x_i^\circ)^{\gamma_i - 1} V_i(S_i^*) (R_i(S_i^*) - \hat{u}) + (1 - \gamma_i) V_i(x_i^\circ)^{\gamma_i} (R_i(S_i^*) - \hat{z}) \\
&= \gamma_i V_i(x_i^\circ)^{\gamma_i - 1} V_i(S_i^*) (R_i(S_i^*) - \hat{z}) + (1 - \gamma_i) V_i(x_i^\circ)^{\gamma_i} (R_i(S_i^*) - \hat{z}) \\
&\geq V_i(S_i^*)^{\gamma_i} (R_i(S_i^*) - \hat{z}),
\end{align*}
\]

where the first inequality follows from the inequality we establish at the beginning of this paragraph, the first equality follows by using the definition of $\hat{u}$ and arranging the terms, the second inequality
follows from the fact that \( V_i(x_i^0) (R_i(x_i^0) - \bar{u}) \geq V_i(S_i^*) (R_i(S_i^*) - \bar{u}) \), which is established in the previous paragraph, the second equality follows by using the definition of \( \bar{u} \) and the third inequality follows by noting that \( V_i(S_i^*)^{\gamma_i} \leq \gamma_i V_i(x_i^0)^{\gamma_i - 1} V_i(S_i^*) + (1 - \gamma_i) V_i(x_i^0)^{\gamma_i} \), which is established above by using the subgradient inequality. So, our claim holds when \( x_i^0 \neq 0 \).

Second, we assume that \( x_i^0 = 0 \) so that the optimal solution to the linear programming relaxation of problem (11) is zero when this problem is solved with \( u = \bar{u} \). Thus, the utility of each product in this problem should be negative, yielding \( r_{ij} \leq \bar{u} = \gamma_i \hat{z} + (1 - \gamma_i) R_i(S_i^*) \) for all \( j \in N \), in which case, noting \( R_i(S_i^*) > \hat{z} \), the last inequality implies that \( r_{ij} < R_i(S_i^*) \) for all \( j \in N \). However, since \( R_i(S_i^*) = \sum_{j \in S_i^*} v_{ij} r_{ij} / \sum_{j \in S_i^*} v_{ij} \) by definition, \( R_i(S_i^*) \) is a weighted average of the product revenues \( \{ r_{ij} : j \in S_i^* \} \). So, it is impossible to have \( r_{ij} < R_i(S_i^*) \) for all \( j \in N \), indicating that the case with \( x_i^0 = 0 \) cannot occur. Thus, our claim is established and we have \( \hat{y}_i \geq V_i(S_i^*)^{\gamma_i} (R_i(S_i^*) - \hat{z}) \) for all \( i \in M \). Adding these inequalities over all \( i \in M \) and noting that \( v_0 \hat{z} \geq \sum_{i \in M} \hat{y}_i \) by the first constraint in problem (14), we obtain \( v_0 \hat{z} \geq \sum_{i \in M} V_i(S_i^*)^{\gamma_i} (R_i(S_i^*) - \hat{z}) \), in which case, solving for \( \hat{z} \) in this inequality and noting the definition of \( \Pi(S_1, \ldots, S_m) \), we get \( \hat{z} \geq \Pi(S_1^*, \ldots, S_m^*) \). So, \( \hat{z} \) is an upper bound on the expected revenue from the optimal assortment, as desired. \qed

B Online Supplement: Extensions to Other Constraints

In this section, we show how to extend our approach to more involved types of constraints.

B.1 Cardinality and Space Constraints

In this section, we consider the case with both cardinality and space constraints on the offered assortment so that the feasible assortments in nest \( i \) are \( C_i = \{ S_i \subset N : |S_i| \leq b_i, \sum_{j \in S_i} w_{ij} \leq c_i \} \), where \( b_i \) is the limit on the cardinality of the offered assortment and \( c_i \) is the space availability in nest \( i \). For this case, it is possible to come up with a collection of assortments \( \{ A_i^t : t \in T_i \} \) with \( |T_i| = O(n^2) \) such that this collection includes a 3-approximate solution to problem (8) for any \( u \in \mathbb{R}_+ \). Thus, we can solve a linear program with \( 1 + m \) decision variables and \( 1 + O(mn^2) \) constraints to obtain a solution to the assortment optimization problem whose expected revenue deviates from the optimal expected revenue by no more than a factor of three.

The discussion in this section follows the development in Sections 4 and 6 closely. So, we mostly focus on the main points. Using the decision variables \( x_i = (x_{i1}, \ldots, x_{in}) \in \{0,1\}^n \), we write problem (8) under cardinality and space constraints as

\[
\max \left\{ \sum_{j \in N} v_{ij} (r_{ij} - u) x_{ij} : \sum_{j \in N} x_{ij} \leq b_i, \sum_{j \in N} w_{ij} x_{ij} \leq c_i, x_{ij} \in \{0,1\} \ \forall \ j \in N \right\},
\]

which is a cardinality constrained knapsack problem. A basic exercise in duality theory shows that there are at most \( n^2 \) possible optimal bases to the linear programming relaxation of problem (17). Naturally, for any \( u \in \mathbb{R}_+ \), the optimal solution to the linear programming relaxation of
problem (17) must correspond to one of these optimal bases. In other words, for any \( u \in \mathbb{R}_+ \), the optimal solution to the linear programming relaxation of problem (17) is one of \( n^2 \) solutions. By using the parametric simplex method over \( u \in \mathbb{R}_+ \), we can generate all of these \( n^2 \) solutions. We use \( \{x_i^u : g \in G_i\} \) with \( |G_i| = O(n^2) \) to denote all possible solutions to the linear programming relaxation of problem (17). So, for any \( u \in \mathbb{R}_+ \), there exists some \( x_i^u \) with \( g \in G_i \) such that \( x_i^u \) is the optimal solution to the linear programming relaxation of problem (17).

Implicitly treating the upper bounds \( 0 \leq x_{ij} \leq 1 \) for all \( j \in N \) in the linear programming relaxation of problem (17), we observe that there must be two basic decision variables in any basic optimal solution and the other decision variables have integer values. Thus, the solution \( x_i^u \) has at most two fractional components. Using the solution \( x_i^u \), we define the assortment \( S_i^u \) as \( S_i^u = \{j \in N : x_{ij}^u = 1\} \), including the products that take value one in the solution \( x_i^u \). In this case, augmenting the collection \( \{S_i^u : g \in G_i\} \) with the collection of singleton assortments \( \{(j) : j \in N\} \), it is possible to show that the collection of assortments \( \{S_i^u : g \in G_i\} \cup \{(j) : j \in N\} \) includes a 3-approximate solution to problem (17) for any \( u \in \mathbb{R}_+ \). To see this result, assume that we solve problem (17) for some \( u \in \mathbb{R}_+ \) and let \( g \) be such that \( x_i^u \) is the optimal solution to the linear programming relaxation of problem (17) when we solve this problem with the value of \( u \) in consideration. By the discussion at the beginning of this paragraph, the solution \( x_i^u \) has at most two fractional components. We use \( j_i^1 \) to denote the first fractional component of \( x_i^u \) when there is one. Similarly, we use \( j_i^2 \) to denote the second fractional component of \( x_i^u \) when there is one. In this case, if we let \( z^*(u) \) be the optimal objective value of problem (17), then noting that the optimal objective value of the linear programming relaxation provides an upper bound on \( z^*(u) \), we obtain the chain of inequalities

\[
z^*(u) \leq \sum_{j \in N} v_{ij} (r_{ij} - u) x_{ij}^u \leq \sum_{j \in S_i^u} v_{ij} (r_{ij} - u) + v_{ij}^1 (r_{ij}^1 - u) + v_{ij}^2 (r_{ij}^2 - u)
\]

\[
\leq 3 \max \left\{ \sum_{j \in S_i^u} v_{ij} (r_{ij} - u), v_{ij}^1 (r_{ij}^1 - u), v_{ij}^2 (r_{ij}^2 - u) \right\}, \quad (18)
\]

where the second inequality follows from the fact that \( S_i^u \cup \{j_i^1\} \cup \{j_i^2\} \) includes all components of the solution \( x_i^u \) that take strictly positive values. If the solution \( x_i^u \) does not have any fractional components, then the chain of inequalities above continue to hold by ignoring all terms that involve \( j_i^1 \) and \( j_i^2 \). Similarly, if the solution \( x_i^u \) has one fractional component, then the chain of inequalities continue to hold by ignoring all terms that involve \( j_i^2 \). The chain of inequalities above implies that either one of the assortments \( S_i^u \), \( \{j_i^1\} \) and \( \{j_i^2\} \) is a 3-approximate solution to problem (17). Therefore, the collection of assortments \( \{S_i^u : g \in G_i\} \cup \{(j) : j \in N\} \) includes a 3-approximate solution to problem (17) for any \( u \in \mathbb{R}_+ \). Noting that \( |G_i| = O(n^2) \), there are \( O(n^2) \) assortments in this collection and we obtain the desired result.

It turns out that we can tighten the approximation guarantee from three to two by using a somewhat more involved definition of the assortment \( S_i^u \). If the solution \( x_i^u \) has zero or one fractional
component, then we continue defining $S_i^g$ as $S_i^g = \{ j \in N : x_{ij}^g = 1 \}$. However, if the solution $x_i^g$ has two fractional components, then using $j_1^g$ and $j_2^g$ to denote these fractional components with the convention that $w_{ij_1^g} \leq w_{ij_2^g}$, we define the assortment $S_i^g$ as $S_i^g = \{ j \in N : x_{ij}^g = 1\} \cup \{j_1^g\}$, including all products that take value one and the product with the smaller space requirement that takes a fractional value in the solution $x_i^g$. In this case, we can show that the collection of assortments \{ $S_i^g : g \in G_i \} \cup \{ \{ j \} : j \in N \}$ includes a 2-approximate solution to problem (17) for any $u \in \mathbb{R}_+$. To see this result, we can follow another basic exercise in duality theory to show that if there are two fractional components in a basic optimal solution to the linear programming relaxation of problem (17), then both constraints must be satisfied as equality. Thus, if $x_i^g$ has two fractional components $j_1^g$ and $j_2^g$, then it must satisfy $x_{ij_1^g}^g + x_{ij_2^g}^g = 1$. The last expression, together with the fact that $w_{ij_1^g} \leq w_{ij_2^g}$ and $x_i^g$ is a feasible solution to the linear programming relaxation of problem (17), yields $c_i \geq \sum_{j \in N} w_{ij} x_{ij}^g = \sum_{j \in S_i^g} w_{ij} 1(x_{ij}^g = 1) + w_{ij_1^g} x_{ij_1^g}^g + w_{ij_2^g} x_{ij_2^g}^g \geq \sum_{j \in S_i^g} w_{ij} 1(x_{ij}^g = 1) + w_{ij_2^g}$.

So, the assortment $S_i^g = \{ j \in N : x_{ij}^g = 1\} \cup \{j_1^g\}$ is feasible to problem (17). In this case, we can use the same line of reasoning in (18) to get

$$z^*(u) \leq \sum_{j \in N} v_{ij} (r_{ij} - u) x_{ij}^g \leq \sum_{j \in S_i^g} v_{ij} (r_{ij} - u) + v_{ij_2^g} (r_{ij_2^g} - u),$$

where the second inequality follows since the assortment $S_i^g$ includes all strictly positive components of $x_i^g$ except for $j_2^g$. We can use $2 \max\{\sum_{j \in S_i^g} v_{ij} (r_{ij} - u), v_{ij_2^g} (r_{ij_2^g} - u)\}$ to bound the right side above so that either $S_i^g$ or $\{j_2^g\}$ is a 2-approximate solution to problem (17), as desired.

B.2 Hierarchical Parent Product Constraints

Parent product constraints hierarchically organize the products in two levels so that a child product cannot be offered unless its parent product is offered. It turns out that we can extend the levels of hierarchy to more than two without too much difficulty. To demonstrate, beside the parent products $P_1$ and the child products $\{C_{ij} : j \in P_1\}$, we assume that each child product $k$ in nest $i$ has a set of grandchild products $G_{ik}$ so that a grandchild product cannot be offered unless its corresponding child product is offered. Similar to our assumption for child products, we assume that $G_{ik} \cap G_{il}$ for all distinct child products $k, l \in \cup_{j \in P_1} C_{ij}$ so that the sets of grandchild products of two child products are different. We allow a child product $k$ not to have any grandchild products, in which case, $G_{ik} = \emptyset$. Under these assumptions, the sets of parent products $P_1$, child products $\{C_{ij} : j \in P_1\}$ and grandchild products $\{G_{ik} : k \in C_{ij}, j \in P_1\}$ collectively partition $N$. Also, we can organize the products in a tree hierarchy as shown in Figure 2, where the three levels of the tree respectively show the parent, child and grandchild products. In this figure, the parent products are $P_i = \{1, 2\}$. For parent product 1, $C_{i1} = \{3, 4\}$. For child product 3, $G_{i3} = \{7, 8, 9\}$.

If we have parent product constraints with a hierarchy of three levels, then we can follow precisely the same reasoning in Section 5 to come up with a collection of assortments $\{A_t^i : t \in T_i\}$ with $|T_i| = O(n)$ such that this collection includes an optimal solution to problem (8) for any $u \in \mathbb{R}_+$. In this case, we can obtain the optimal solution to problem (1) by solving a linear
Figure 2: Parent product constraints with a hierarchy of three levels.

program with $1 + m$ decision variables and $1 + O(mn)$ constraints. To obtain this result, we use the idea in Section 5 to observe that if child product $k$ is offered, then this child product and all of its grandchild products make a total contribution of $f_{ik}(u) = v_{ik} (r_{ik} - u) + \sum_{l \in G_{ik}} v_{il} [r_{il} - u]^+$ to the objective function of problem (8). Thus, it is optimal to offer child product $k$ only if $f_{ik}(u) > 0$ and it is optimal to offer grandchild product $l$ of this child product only if $u < r_{il}$. Since $f_{ik}(\cdot)$ is decreasing and piecewise linear, it is straightforward to find a value of $\bar{u}_{ik}$ such that $f_{ik}(u) > 0$ for any $u < \bar{u}_{ik}$ and $f_{ik}(u) \leq 0$ for any $u \geq \bar{u}_{ik}$. Going one level up the hierarchy, if we offer parent product $j$, then this parent product, all of its child products and all of the grandchild products of its child products make a total contribution of $f_{ij}(u) = v_{ij} (r_{ij} - u) + \sum_{k \in C_{ij}} [f_{ik}(u)]^+$ to the objective function of problem (8). So, it is optimal to offer parent product $j$ only if $f_{ij}(u) > 0$. Since $f_{ij}(\cdot) : k \in C_{ij}$ are decreasing and piecewise linear, so is $f_{ij}(\cdot)$ and we can easily find a value of $\bar{u}_{ij}$ such that $f_{ij}(u) > 0$ for any $u < \bar{u}_{ij}$ and $f_{ij}(u) \leq 0$ for any $u \geq \bar{u}_{ij}$.

If we repeat the same reasoning for all of the parent products, then we obtain the collections of points $\{\bar{u}_{ij} : j \in P_i\}$ for the parent products, $\{\bar{u}_{ik} : j \in P_i, k \in C_{ij}\}$ for the child products and $\{r_{il} : j \in P_i, k \in C_{ij}, l \in G_{ik}\}$ for the grandchild products. Since $P_i$, $\{C_{ij} : j \in P_i\}$ and $\{G_{ik} : j \in P_i, k \in C_{ij}\}$ collectively partition $N$, there are a total of $n$ points in the collections $\{\bar{u}_{ij} : j \in P_i\}$, $\{\bar{u}_{ik} : j \in P_i, k \in C_{ij}\}$ and $\{r_{il} : j \in P_i, k \in C_{ij}, l \in G_{ik}\}$. These points completely characterize the optimal solution to problem (8) under parent product constraints with three levels, since we can compare $u$ with $\bar{u}_{ij}$ to decide whether it is optimal to offer parent product $j$. If this is the case, then we can decide whether it is optimal to offer its child product $k$ by comparing $u$ with $\bar{u}_{ik}$. Finally, if it is optimal to offer child product $k$, then we can decide whether to offer its grandchild product $l$ by comparing $u$ with $r_{il}$. The total of $n$ points in the collections $\{\bar{u}_{ij} : j \in P_i\}$, $\{\bar{u}_{ik} : j \in P_i, k \in C_{ij}\}$ and $\{r_{il} : j \in P_i, k \in C_{ij}, l \in G_{ik}\}$ partition the positive real line into $O(n)$ intervals. In this case, we can follow the same line of reasoning in Section 5 to obtain a collection of assortments with $O(n)$ assortments in it such that this collection includes the optimal solution to problem (8) for any $u \in \mathbb{R}_+$. 

39