Capacity Constraints Across Nests in Assortment Optimization Under the Nested Logit Model

Jacob B. Feldman
School of Operations Research and Information Engineering,
Cornell University, Ithaca, New York 14853, USA
jbf232@cornell.edu

Huseyin Topaloglu
School of Operations Research and Information Engineering,
Cornell University, Ithaca, New York 14853, USA
topaloglu@orie.cornell.edu

March 28, 2014

Abstract

We consider assortment optimization problems when customers choose according to the nested logit model and there is a capacity constraint limiting the total capacity consumption of all products offered in all nests. When each product consumes one unit of capacity, our capacity constraint limits the cardinality of the offered assortment. For the cardinality constrained case, we develop an efficient algorithm to compute the optimal assortment. When the capacity consumption of each product is arbitrary, we give an algorithm to obtain a 4-approximate solution. Furthermore, we develop a convex program that computes an upper bound on the optimal expected revenue for an individual problem instance. In our numerical experiments, we consider problem instances involving products with arbitrary capacity consumptions. Comparing the expected revenues from the assortments obtained by our 4-approximation algorithm with the upper bounds on the optimal expected revenues, our numerical results indicate that the 4-approximation algorithm performs quite well, yielding about 2% optimality gap on average.

A conventional approach to modeling demand in revenue management is to assume that each customer arrives into the system with the intention of purchasing a fixed product. If this product is available for sale, then the customer purchases it. Otherwise, the customer leaves the system without making a purchase. In reality, however, there may be multiple products that can potentially serve the needs of a customer, in which case, customers may make a choice between the products and may substitute a product for another one when their favorite product is not available. This kind of a choice process creates interactions between the demand for the different products, inflating the demand for an available product when some other product is not available so that customers satisfy their needs by substituting for the available product. A common question that arises in this setting is what products to make available to customers so as to maximize the expected revenue, given that customers choose and substitute according to a particular choice model.

In this paper, we consider assortment optimization problems when customers choose according to the nested logit model and there is limited capacity for the products in the offered assortment. We consider a setting where we need to decide which assortment of products to offer. Each arriving customer chooses among the offered products according to the nested logit model. Under the nested logit model, the products are organized in nests. Each customer, after viewing the offered assortment, decides either to make a purchase within one of the nests or to leave the system without purchasing anything. If a nest is chosen, then the customer purchases one of the products within the chosen nest. There is a capacity constraint limiting the total capacity consumption of the products in the offered assortment. The goal is to choose an assortment of products to offer so as to maximize the expected revenue obtained from each customer. We consider two types of capacity constraints. In the first type of constraints, each product occupies one unit of space, in which case, the capacity constraint limits the total number of products in the offered assortment. We refer to this type of a capacity constraint as a *cardinality constraint*. In the second type of constraints, the capacity consumption of a product is arbitrary, possibly reflecting the space or capital requirement of a product. We refer to this type of a capacity constraint as a *space constraint*.

Under a cardinality constraint, we show that we can compute the optimal assortment in a tractable fashion. As far as we are aware, the assortment problem was not known to be tractable when customers choose according to the nested logit model and there is a cardinality constraint limiting the total number of products in the offered assortment. This paper gives the first exact solution method for this problem. On the other hand, we give a 4-approximation algorithm under a space constraint, providing an assortment whose expected revenue deviates from the optimal expected revenue by at most a factor of four. The running time of this algorithm scales gracefully with the number of products and the number of nests. To our knowledge, this paper gives the first algorithm for the assortment problem that scales gracefully with the number of nests and the number of products, when there is a capacity constraint on the space consumption of all offered products and customers choose according to the nested logit model. In addition to giving algorithms to solve the assortment problem, we give a tractable convex program that computes an upper bound on the optimal expected revenue. By comparing the expected revenues of the assortments obtained by our 4-approximation algorithm with the upper bounds on the optimal expected revenues, we demonstrate that our 4-approximation algorithm performs quite well in practice, yielding solutions with about 2% optimality gap on average.

Main Contributions. Under a cardinality constraint, we show that we can obtain an optimal assortment by solving a linear program with $O(m^2 n)$ decision variables and $O(m^2 n^4)$ constraints, where m in the number of nests and n is the number of products in each nest. To our knowledge, this is the first efficient algorithm for solving the assortment optimization problem under the nested logit model with a constraint on the total cardinality of the assortment. Rusmevichientong et al. (2010) solve a similar assortment optimization problem efficiently when customers choose according to the multinomial logit model. Our result is naturally more general than theirs as the multinomial logit model is a special case of the nested logit model with a single nest.

Under a space constraint, we show that we can obtain a 4-approximate solution by solving a linear program with $O(m)$ decision variables and $O(mn^4)$ constraints. Computational work for our approach scales well with both the number of nests and the number of products. As far as we are aware, ours is the first algorithm that scales well with the number of nests. Rusmevichientong et al. (2009) and Desir and Goyal (2013) give approximation schemes for related assortment problems, but the running time for their approaches unfortunately grows exponentially with the number of nests. For example, to obtain a 4-approximate solution, Rusmevichientong et al. (2009) need $O(m(m^6 n^6 \log(mn))^m)$ operations, which gets prohibitive when m exceeds two or three.

We give a tractable convex program to obtain an upper bound on the optimal expected revenue under a space constraint. By comparing the expected revenues from the assortments obtained by our 4-approximation algorithm with the upper bounds on the optimal expected revenues, we can bound the optimality gap of the 4-approximate solution for an individual problem instance. Letting c be the total capacity availability, this convex program is obtained by partitioning $[0, c]$ into a number of intervals. The convex program has $O(m)$ decision variables and one constraint for each interval and for each nest. Any arbitrary partition of $[0, c]$ works to obtain an upper bound, but finer partitions yield tighter upper bounds, at the expense of a larger number of constraints.

Related Literature. A popular approach for modeling customer choice is to use the multinomial logit model, dating back to the work of Luce (1959) and McFadden (1974). This choice model is compatible with utility maximization principle, where each customer associates a random utility with each product and chooses the product with the largest utility. The multinomial logit model is built under the assumption that the utilities associated with different products are independent of each other. In other words, how highly a customer evaluates a certain product has nothing to do with how highly the same customer evaluates another product. The nested logit model remedies this shortcoming. The nested logit model dates back to Williams (1977) and it is also compatible with utility maximization principle. The nested logit model organizes the products in nests such that the utilities associated with the products in the same nest can be dependent on each other; see

Borsch-Supan (1990) and Train (2003). This feature allows the modeler to capture situations where the products in the same nest are alike and how highly a customer evaluates a certain product can be a strong indicator of how highly the same customer evaluates another product.

Talluri and van Ryzin (2004) consider the assortment problem under the multinomial logit model without any constraints and show that the optimal assortment includes a certain number of products with the highest unit revenues. Rusmevichientong et al. (2010), Wang (2012) and Wang (2013) consider assortment problems under variants of the multinomial logit model with a cardinality constraint on the offered assortment and show that the problem can be solved efficiently. Bront et al. (2009), Mendez-Diaz et al. (2010) and Rusmevichientong et al. (2013) consider assortment problems where there are multiple customer types and customers of different types choose according to multinomial logit models with different parameters. The authors show that the problem is NP-complete, study heuristics and investigate valid cuts for integer programming formulations.

Davis et al. (2013) consider the assortment problem under the nested logit model without any constraints. They characterize the optimal subset of products to offer in each nest. Their characterization reduces the number of possible subsets to consider in each nest to a reasonable number, but the optimal assortment is still intractable to compute since there are exponentially many different ways to combine the subsets from different nests. They resolve this difficulty by formulating a linear program to combine the subsets from different nests. Li and Rusmevichientong (2012) give a greedy algorithm for the same problem. Gallego and Topaloglu (2014) study constrained assortment problems under the nested logit model, but they impose capacity constraints separately on the assortment offered in each nest. The authors view each nest as a different product category and their goal is to separately limit the capacity consumption of the products offered in each product category. Their constraints are separable by the nests and do not apply to the practical situation where we want to limit the total capacity consumption of the products offered in all product categories. In this paper, we treat the products in all of the nests as products that occupy a common space and limit the total capacity consumption of the products offered in all nests. Our constraints create interactions between the nests and make the assortment problem much more difficult. As we discuss at the end of the paper, our approach is flexible enough to limit the total capacity consumption of the products offered in all nests, along with the capacity consumption within each nest. Similar to us, Rusmevichientong et al. (2009) and Desir and Goyal (2013) consider the case where there is a constraint on the total capacity consumption of the products offered in all nests. They give approximation schemes that tradeoff running time with solution quality, but the running time for their approaches grows exponentially with the number of nests. Li and Huh (2011), Kok and Xu (2011) and Gallego and Wang (2011) study related pricing and assortment problems without constraints.

A fruitful line of attack for assortment optimization under the nested logit has been to identify a collection of good candidate subsets to offer in each nest. Once these collections are identified, it is possible to solve a separate linear program to pick a subset to offer in each nest so that the combined subsets over all nests provide a good assortment. This is the strategy followed by Davis et al. (2013) and Gallego and Topaloglu (2014). We follow a similar approach in identifying the collections of candidate subsets in each nest. However, due to the fact that our capacity constraint limits the total capacity consumption of the subsets of products offered in all nests, different nests interact with each other, making the assortment problem substantially more difficult. It is not possible to build on the earlier work to figure out how to pick a subset to offer in each nest so that the combined subsets over all nests provides the highest possible expected revenue and a novel approach is required. The linear programs used by Davis et al. (2013) and Gallego and Topaloglu (2014) become completely ineffective. To overcome this difficulty, our key observation is that the problem of picking a subset to offer in each nest so that the combined subsets over all nests provides the highest possible expected revenue can be formulated as a multiple choice knapsack problem. Under a cardinality constraint, we use the dynamic programming formulation of this knapsack problem to find the optimal assortment. Under a space constraint, we use the linear programming relaxation of this knapsack problem to get an approximation method.

Finally, our work is related to revenue management models with customer choice. Talluri and van Ryzin (2004) consider a revenue management problem on a single flight leg, where the airline dynamically adjusts the set of available fare classes and customers choose among the available fare classes. Gallego et al. (2004), Liu and van Ryzin (2008), Kunnumkal and Topaloglu (2008), Zhang and Adelman (2009), Gallego et al. (2011), Kunnumkal (2013), Talluri (2011) and Meissner et al. (2012) extend this model to an airline network. The main idea in this set of papers is to formulate various linear programming approximations, where each decision variable corresponds to the duration of time during which a certain subset of itinerary products is made available to customers. Since these is one decision variable for each subset of itinerary products, the number of decision variables can be quite large and the linear programming approximations are solved through column generation. This column generation subproblem is exactly the assortment problem we consider in this paper when customer choices are governed by the nested logit model. Closing our literature review, we note that our review focuses on assortment optimization with multinomial and nested logit models. For assortment optimization under other choice models, we refer to Kok et al. (2008), Natarajan et al. (2009) and Farias et al. (2012) and Farias et al. (2013).

Organization. In Section 1, we formulate our assortment optimization problem. In Section 2, we assume that we have access to a collection of candidate assortments for each nest so that we can stitch together a good solution to the assortment optimization problem by picking one assortment from each one of these collections. Given these collections, we study the problem of how to stitch together the best solution. In Section 3, we study the problem of how to come up with a good collection of candidate assortments for each nest. In Section 4, we apply the results in Sections 2 and 3 to develop an algorithm to obtain the optimal assortment under a cardinality constraint. In Section 5, we apply the results in Sections 2 and 3 to develop a 4-approximation algorithm under a space constraint. In Section 6, we give a convex program to obtain an upper bound on the optimal expected revenue. In Section 7, we give numerical experiments. In Section 8, we conclude.

1 Problem Formulation

In this section, we formulate the assortment optimization problem that we want to solve. There are m nests indexed by $M = \{1, \ldots, m\}$. In each nest, there are n products that we can offer to customers and we index the products by $N = \{1, \ldots, n\}$. Under the nested logit model, a customer decides either to make a purchase within one of the nests or to leave without purchasing anything. If the customer decides to make a purchase within one of the nests, then the customer chooses one of the products offered in this nest. We let v_{ij} be the preference weight associated with product j in nest i. Given that we offer the assortment $S_i \subset N$ of products in nest i, we use $V_i(S_i) = \sum_{j \in S_i} v_{ij}$ to denote the total preference weight of the products in the offered assortment. According to the nested logit model, if we offer the assortment S_i in nest i and a customer has already decided to make a purchase in this nest, then this customer chooses product $j \in S_i$ with probability $v_{ij}/V_i(S_i)$. We let r_{ij} be the revenue associated with product j in nest i. Given that we offer the assortment S_i in nest i and a customer has already decided to make a purchase in this nest, the expected revenue that we obtain from this customer can be written as

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

We use the convention that $R_i(\emptyset) = 0$ so that the expected revenue from the empty assortment is zero. The notation so far implicitly asserts that each nest includes the same number of products, but this assumption is only for notational brevity and all of our results naturally extend to the case where different nests have different numbers of products.

Associated with each nest, there is a parameter $\gamma_i \in [0, 1]$ capturing the degree of dissimilarity between the products in nest i. We use v_0 to denote the preference weight of the no purchase option. According to the nested logit model, if we offer the assortment $(S_1, \ldots, S_m)$ over all nests with $S_i \subset N$ for all $i \in M$, then a customer chooses nest i with probability

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

The expression above computes the probability that a customer is attracted to nest i as a function of the assortment $(S_1, \ldots, S_m)$ offered over all nests. The form of the choice probabilities above can be derived by using utility maximization principle; see McFadden (1974) and Train (2003). In particular, it is possible to show that if a customer associates a random utility with each product and the no purchase option, choosing the option with the largest utility, then as along as the utilities have a certain generalized extreme value distribution, the probability of choosing a certain nest and the probability of choosing a certain product in a nest have the forms specified above. The utilities of the products in different nests are assumed to be independent, but the parameter γ_i is related to the correlation between the utilities of the products in nest i. Smaller values of γ_i implies that the utilities of the products in nest i are more strongly correlated, in which case, the products in nest i tend to be more similar to each other. If a customer evaluates one product highly, then this customer is likely to evaluate other products in the nest highly as well. So, if we offer the assortment $(S_1, \ldots, S_m)$ over all nests, then we obtain an expected revenue of

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

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

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

$$z^* = \max_{\substack{(S_1, \ldots, S_m)\,: \\ \sum_{i \in M} C_i(S_i) \leq c, \\ S_i \subset N\ \forall\, i \in M}} \Big\{ \Pi(S_1, \ldots, S_m) \Big\}, \tag{1}$$

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

In problem (1), the assortments that we offer in different nests interact with each other due to two reasons. First, the probability $Q_i(S_1, \ldots, S_m)$ that a customer chooses nest i jointly depends on the assortment $(S_1, \ldots, S_m)$ offered over all nests. Second, the constraint in this problem limits the total capacity consumption in all nests and the capacity consumption of the assortment offered in one nests determines the capacity available for the assortments offered in other nests. In this paper, we show that if we have a cardinality constraint on the offered assortment, then problem (1) can be solved efficiently. In particular, the approach that we give requires solving a linear program with $O(m^2 n)$ decision variables and $O(m^2 n^4)$ constraints. On the other hand, if we have a space constraint, then Lemma 2.1 in Rusmevichientong et al. (2009) shows that problem (1) is NP-hard even when there is a single nest with a dissimilarity parameter of one. So, it is likely to be intractable to get an optimal solution to problem (1) under a space constraint. In this paper, we establish that we can solve a linear program with $O(m)$ decision variables and $O(mn^4)$ constraints to get a 4-approximate solution.

2 Fixed Point Representation

In this section, we lay out the connection of problem (1) to the problem of computing the fixed point of an appropriately defined function. By using this connection, we ultimately answer the following crucial question. Assume that we are given a collection of candidate assortments $\mathcal{A}_i = \{A_{it} : t \in \mathcal{T}_i\}$ for each nest i such that $A_{it} \subset N$ for all $t \in \mathcal{T}_i$. We know that by picking one assortment from each one of the collections $\mathcal{A}_1, \ldots, \mathcal{A}_m$, we can stitch together an α-approximate solution to problem (1). That is, there exists $(\hat{S}_1, \ldots, \hat{S}_m)$ with $\hat{S}_i \in \mathcal{A}_i$ for all $i \in M$ and $\sum_{i\in M} C_i(\hat{S}_i) \leq c$ such that $\alpha\,\Pi(\hat{S}_1, \ldots, \hat{S}_m) \geq z^*$. The question is how to find this α-approximate solution to problem (1) in a tractable fashion. Finding an answer to this question through complete enumeration is intractable since there are $|\mathcal{A}_1| \times \ldots \times |\mathcal{A}_m|$ combinations of assortments to consider from the different nests. To answer this question, we define $f(\cdot)$ as

$$f(z) \;=\; \max_{\substack{(S_1,\ldots,S_m)\,:\\ \sum_{i\in M} C_i(S_i) \leq c,\\ S_i \in \mathcal{A}_i \; \forall\, i \in M}} \left\{ \sum_{i\in M} V_i(S_i)^{\gamma_i} (R_i(S_i) - z) \right\}. \tag{2}$$

Consider $\hat{z}$ that satisfies $v_0\,\hat{z} = f(\hat{z})$, which corresponds to the fixed point of the function $f(\cdot)/v_0$. Such a value of $\hat{z}$ always exists since $f(z)$ is decreasing and $v_0\,z$ is increasing in z, with $f(0) \geq 0$. In the next theorem, we show that the value of $\hat{z}$ that satisfies $v_0\,\hat{z} = f(\hat{z})$ is useful in identifying an α-approximate solution to problem (1).

Theorem 1 *Assume that we can stitch together an α-approximate solution to problem* (1) *by picking one assortment from each one of the collections $\mathcal{A}_1, \ldots, \mathcal{A}_m$. Let $\hat{z}$ satisfy $v_0\,\hat{z} = f(\hat{z})$ and $(\hat{S}_1, \ldots, \hat{S}_m)$ be an optimal solution to problem* (2) *when we solve this problem with $z = \hat{z}$. Then, we have $\alpha\hat{z} = \alpha\,\Pi(\hat{S}_1, \ldots, \hat{S}_m) \geq z^*$.*

Proof. By our hypothesis, there exists $(\tilde{S}_1, \ldots, \tilde{S}_m)$ with $\tilde{S}_i \in \mathcal{A}_i$ and $\sum_{i\in M} C_i(\tilde{S}_i) \leq c$ such that $\alpha\,\Pi(\tilde{S}_1, \ldots, \tilde{S}_m) \geq z^*$. So, $(\tilde{S}_1, \ldots, \tilde{S}_m)$ is a feasible solution to problem (2) when solved with $z = \hat{z}$, in which case, we obtain $v_0\,\hat{z} = f(\hat{z}) \geq \sum_{i\in M} V_i(\tilde{S}_i)^{\gamma_i}(R_i(\tilde{S}_i) - \hat{z})$. If we focus on the first and last terms in the last chain of inequalities and solve for $\hat{z}$, then we obtain $\hat{z} \geq \sum_{i\in M} V_i(\tilde{S}_i)^{\gamma_i} R_i(\tilde{S}_i)/(v_0 + \sum_{i\in M} V_i(\tilde{S}_i)^{\gamma_i}) = \Pi(\tilde{S}_1, \ldots, \tilde{S}_m)$. Multiplying this chain of inequalities with α and noting that $\alpha\,\Pi(\tilde{S}_1, \ldots, \tilde{S}_m) \geq z^*$, we get $\alpha\,\hat{z} \geq z^*$. To complete the proof, we show that $\Pi(\hat{S}_1, \ldots, \hat{S}) = \hat{z}$. Since $v_0\,\hat{z} = f(\hat{z})$ and $(\hat{S}_1, \ldots, \hat{S}_m)$ is an optimal solution to problem (2) when solved with $z = \hat{z}$, we have $v_0\,\hat{z} = \sum_{i\in M} V_i(\hat{S}_i)^{\gamma_i}(R_i(\hat{S}_i) - \hat{z})$ and solving for $\hat{z}$ in this equality yields $\hat{z} = \sum_{i\in M} V_i(\hat{S}_i)^{\gamma_i} R_i(\hat{S}_i)/(v_0 + \sum_{i\in M} V_i(\hat{S}_i)^{\gamma_i}) = \Pi(\hat{S}_1, \ldots, \hat{S}_m)$. □

The theorem above suggests the following procedure to obtain an α-approximate solution to problem (1). We find $\hat{z}$ such that $v_0\,\hat{z} = f(\hat{z})$. In this case, we can solve problem (2) with $z = \hat{z}$ to obtain the optimal solution $(\hat{S}_1, \ldots, \hat{S}_m)$ and this solution is an α-approximate solution to problem (1). As we show later in the paper, we can efficiently find $\hat{z}$ that satisfies $v_0\,\hat{z} = f(\hat{z})$ when we have a cardinality constraint on the offered assortment. However, finding such $\hat{z}$ may be difficult when we have a space constraint. In the next corollary, we give an approximate version of Theorem 1 that does not require finding $\hat{z}$ such that $v_0\,\hat{z} = f(\hat{z})$. To state this corollary, we let $f^R(\cdot)$ be a relaxation of $f(\cdot)$ that satisfies $f^R(z) \geq f(z)$ for all $z \in \Re_+$. We do not yet specify how to construct this relaxation. We only assume that $f^R(z)$ is decreasing in z similar to $f(z)$, with $f^R(0) \geq 0$, in which case, we can always find $\hat{z}$ satisfying $v_0\,\hat{z} = f^R(\hat{z})$. The next corollary to Theorem 1 shows how we can use this value of $\hat{z}$ to get an approximation guarantee for problem (1).

Corollary 2 *Assume that we can stitch together an α-approximate solution to problem* (1) *by picking one assortment from each one of the collections $\mathcal{A}_1, \ldots, \mathcal{A}_m$ and $f^R(\cdot)$ is a relaxation of $f(\cdot)$ that satisfies $f^R(z) \geq f(z)$ for all $z \in \Re_+$. Let $\hat{z}$ be such that $v_0\,\hat{z} = f^R(\hat{z})$. If the assortment $(\hat{S}_1, \ldots, \hat{S}_m)$ satisfies*

$$\beta \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i} (R_i(\hat{S}_i) - \hat{z}) \geq v_0\,\hat{z}$$

for some $\beta \geq 1$, then we have $\alpha\,\beta\,\Pi(\hat{S}_1, \ldots, \hat{S}_m) \geq z^*$.

Proof. Let $\tilde{z}$ be such that $v_0\,\tilde{z} = f(\tilde{z})$. We observe that $\hat{z} \geq \tilde{z}$, since otherwise, we obtain $f(\tilde{z}) = v_0\,\tilde{z} > v_0\,\hat{z} = f^R(\hat{z}) \geq f^R(\tilde{z})$, where the last inequality follows by noting that $f^R(\cdot)$ is decreasing and $\hat{z} < \tilde{z}$. The last chain of inequalities yields $f(\tilde{z}) > f^R(\tilde{z})$, contradicting the fact that $f^R(z) \geq f(z)$ for all $z \in \Re_+$. So, since $\alpha\,\tilde{z} \geq z^*$ by Theorem 1 and $\hat{z} \geq \tilde{z}$, we get $\alpha\,\beta\,\hat{z} \geq \alpha\,\hat{z} \geq z^*$, which implies that the assortment $(\hat{S}_1, \ldots, \hat{S}_m)$ satisfies $\sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i} (\alpha\,\beta R_i(\hat{S}_i) - z^*) \geq \alpha\,\beta \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i} (R_i(\hat{S}_i) - \hat{z}) \geq \alpha\,v_0 \hat{z} \geq v_0\,z^*$, where the second inequality follows from the inequality in the corollary. Focusing on the first and last terms in the last chain of inequalities and solving for z^*, we get $z^* \leq \sum_{i \in M} \alpha\,\beta\,V_i(\hat{S}_i)^{\gamma_i} R_i(\hat{S}_i) / (v_0 + \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i}) = \alpha\,\beta\,\Pi(\hat{S}_1, \ldots, \hat{S}_m)$. $\square$

We make use of use Theorem 1 and Corollary 2 as follows. Assume that the collections of candidate assortments $\mathcal{A}_1, \ldots, \mathcal{A}_m$ allow us to stitch together an α-approximate solution to problem (1). Under a cardinality constraint, we show that we can efficiently solve problem (2) and find $\hat{z}$ satisfying $v_0\,\hat{z} = f(\hat{z})$ in a tractable fashion. So, by Theorem 1, solving problem (2) with $z = \hat{z}$ immediately yields an α-approximate solution to problem (1). Under a space constraint, we construct an appropriate relaxation $f^R(\cdot)$ of $f(\cdot)$ and this relaxation allows us to find $\hat{z}$ satisfying $v_0\,\hat{z} = f^R(\hat{z})$. Also, for this value of $\hat{z}$, we show that we can obtain an assortment $(\hat{S}_1, \ldots, \hat{S}_m)$ that satisfies the inequality in Corollary 2 with $\beta = 2$. So, noting Corollary 2, the assortment $(\hat{S}_1, \ldots, \hat{S}_m)$ provides a 2α-approximate solution to problem (1).

Theorem 1 and Corollary 2 are under the assumption that we already have collections of candidate assortments $\mathcal{A}_1, \ldots, \mathcal{A}_m$ such that we can stitch together an α-approximate solution to problem (1) by picking one assortment from each one of these collections. In the next section, we consider the question of how to construct such collections of candidate assortments.

3 Candidate Assortments

In this section, we show how to construct a collection of candidate assortments $\mathcal{A}_i$ for each nest i such that we can stitch together an α-approximate solution to problem (1) by picking one assortment from each one of the collections $\mathcal{A}_1, \ldots, \mathcal{A}_m$. In the next lemma, we give a simple condition for an assortment to provide an α-approximate solution to problem (1).

Lemma 3 *Let $(S_1^*, \ldots, S_m^*)$ be an optimal solution to problem (1) with an objective value of z^* and $b_i^* = C_i(S_i^*)$. If the assortments $\hat{S}_i$, $i \in M$ satisfy $C_i(\hat{S}_i) \leq b_i^*$ and*

$$\alpha \, V_i(\hat{S}_i)^{\gamma_i} \, (R_i(\hat{S}_i) - z^*) \geq V_i(S_i^*)^{\gamma_i} \, (R_i(S_i^*) - z^*) \tag{3}$$

for some $\alpha \geq 1$, then $(\hat{S}_1, \ldots, \hat{S}_m)$ is a feasible solution to problem (1) and the objective value provided by this solution satisfies $\alpha \, \Pi(\hat{S}_1, \ldots, \hat{S}_m) \geq z^*$.

Proof. Since $\sum_{i \in M} C_i(\hat{S}_i) \leq \sum_{i \in M} b_i^* = \sum_{i \in M} C_i(S_i^*) \leq c$, $(\hat{S}_1, \ldots, \hat{S}_m)$ is a feasible solution to problem (1). Noting that $\Pi(S_1^*, \ldots, S_m^*) = \sum_{i \in M} V_i(S_i^*)^{\gamma_i} \, R_i(S_i^*)/(v_0 + \sum_{i \in M} V_i(S_i^*)^{\gamma_i}) = z^*$, rearranging the terms in the last equality yields $v_0 \, z^* = \sum_{i \in M} V_i(S_i^*)^{\gamma_i} \, (R_i(S_i^*) - z^*)$. Therefore, if we add (3) over all $i \in M$, then it follows that $\alpha \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i} \, (R_i(\hat{S}_i) - z^*) \geq \sum_{i \in M} V_i(S_i^*)^{\gamma_i} \, (R_i(S_i^*) - z^*) = v_0 \, z^*$. Furthermore, noting that $\alpha \geq 1$ and $z^* \geq 0$, we also have $\alpha \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i} (R_i(\hat{S}_i) - z^*) \leq \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i} \, (\alpha R_i(\hat{S}_i) - z^*)$, in which case, the last two chains of inequalities 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 the last inequality, we obtain $\alpha \, \Pi(\hat{S}_1, \ldots, \hat{S}_m) = \alpha \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i} \, R_i(\hat{S}_i)/(v_0 + \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i}) \geq z^*$. □

The lemma above is not immediately useful to characterize assortments that provide an approximation guarantee of α since checking (3) requires knowing S_i^* and z^*. In the next lemma, we use the approach in Gallego and Topaloglu (2014) and the previous lemma to give a more tractable condition for an assortment to provide an α-approximate solution to problem (1).

Lemma 4 *Let $(S_1^*, \ldots, S_m^*)$ be an optimal solution to problem (1) with an objective value of z^*, $b_i^* = C_i(S_i^*)$ and $u_i^* = \max\{z^*, \gamma_i \, z^* + (1 - \gamma_i) \, R_i(S_i^*)\}$. If the assortments $\hat{S}_i$, $i \in M$ satisfy $C_i(\hat{S}_i) \leq b_i^*$ and*

$$\alpha \, V_i(\hat{S}_i) \, (R_i(\hat{S}_i) - u_i^*) \geq V_i(S_i^*) \, (R_i(S_i^*) - u_i^*) \tag{4}$$

for some $\alpha \geq 1$, then $(\hat{S}_1, \ldots, \hat{S}_m)$ is a feasible solution to problem (1) and the objective value provided by this solution satisfies $\alpha \, \Pi(\hat{S}_1, \ldots, \hat{S}_m) \geq z^*$.

Proof. The fact that the assortment $(\hat{S}_1, \ldots, \hat{S}_m)$ is a feasible solution to problem (1) follows from the same argument in the proof of Lemma 3. For notational brevity, we let $V_i^* = V_i(S_i^*)$, $R_i^* = R_i(S_i^*)$, $\hat{V}_i = V_i(\hat{S}_i)$ and $\hat{R}_i = R_i(\hat{S}_i)$. First, assume that $R_i^* < z^*$. A simple lemma, labeled as Lemma 10 in the appendix, shows that if $R_i^* < z^*$, then $S_i^* = \emptyset$. Thus, we have

$V_i^* = V_i(\emptyset) = 0$. Also, since $R_i^* < z^*$, the definition of u_i^* yields $u_i^* = z^*$, in which case, (4) implies that $\alpha \hat{V}_i \, (\hat{R}_i - z^*) \geq 0$. Thus, we obtain $\alpha \, \hat{V}_i^{\gamma_i} \, (\hat{R}_i - z^*) \geq 0 = (V_i^*)^{\gamma_i} \, (R_i^* - z^*)$ under the case $R_i^* < z^*$. Second, assume that $R_i^* \geq z^*$. The definition of u_i^* yields $u_i^* = \gamma_i \, z^* + (1-\gamma_i) \, R_i^*$, in which case, (4) implies that $\alpha \, \hat{V}_i \, (\hat{R}_i - \gamma_i \, z^* - (1-\gamma_i) \, R_i^*) \geq \gamma_i \, V_i^* \, (R_i^* - z^*)$. If $\hat{V}_i = 0$, then the last inequality implies that $V_i^* \, (R_i^* - z^*) = 0$ and we obtain $\alpha \, \hat{V}_i^{\gamma_i} \, (\hat{R}_i - z^*) = 0 = (V_i^*)^{\gamma_i} \, (R_i^* - z^*)$ under the case $R_i^* \geq z^*$. If, on the other hand, $\hat{V}_i > 0$, then we write the last inequality as

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

Since $g(x) = x^{\gamma_i}$ is a concave function of x with $g(1) = 1$ and $g'(1) = \gamma_i$, we have $x^{\gamma_i} \leq 1 + \gamma_i \, (x-1) = \gamma_i \, x + (1-\gamma_i) \leq \gamma_i \, x + \alpha \, (1-\gamma_i)$ for all $x \in \Re_+$. Using this inequality with $x = V_i^*/\hat{V}_i$ on the right side of the inequality above, we get $\alpha \, (\hat{R}_i - z^*) \geq (V_i^*/\hat{V}_i)^{\gamma_i} \, (R_i^* - z^*)$ Arranging the terms in the last inequality yields $\alpha \, \hat{V}_i^{\gamma_i} \, (\hat{R}_i - z^*) \geq (V_i^*)^{\gamma_i} \, (R_i^* - z^*)$ under the case $R_i^* \geq z^*$. So, under both cases, we have $\alpha \, \hat{V}_i^{\gamma_i} \, (\hat{R}_i - z^*) \geq (V_i^*)^{\gamma_i} \, (R_i^* - z^*)$, implying that the assortment $(\hat{S}_1, \ldots, \hat{S}_m)$ satisfies the assumption of Lemma 3. Thus, we must have $\alpha \, \Pi(\hat{S}_1, \ldots, \hat{S}_m) \geq z^*$. ☐

If we let $\hat{S}_i^\alpha$ be an α-approximate solution to the problem $\max_{S_i \,:\, C_i(S_i) \leq b_i^*, \, S_i \subset N} V_i(S_i) \, (R_i - u_i^*)$, then $\hat{S}_i^\alpha$ satisfies $\alpha \, V_i(\hat{S}_i^\alpha) \, (R_i(\hat{S}_i^\alpha) - u_i^*) \geq \max_{S_i \,:\, C_i(S_i) \leq b_i^*, \, S_i \subset N} V_i(S_i) \, (R_i - u_i^*)$ and $C_i(\hat{S}_i^\alpha) \leq b_i^*$. So, the last two inequalities indicate that $\hat{S}_i^\alpha$ satisfies (4) and $C_i(\hat{S}_i^\alpha) \leq b_i^*$, in which case, Lemma 4 implies that the assortment $(\hat{S}_1^\alpha, \ldots, \hat{S}_m^\alpha)$ provides an α-approximate solution to problem (1). Thus, if the collection of candidate assortments $\mathcal{A}_i$ for nest i includes just the single assortment $\hat{S}_i^\alpha$, then we can stitch together an α-approximate solution to problem (1) by picking one assortment from each one of the collections $\mathcal{A}_1, \ldots, \mathcal{A}_m$. The difficulty with this reasoning is that finding $\hat{S}_i^\alpha$, an α-approximate solution to the problem at the beginning of this paragraph, requires knowing b_i^* and u_i^*, neither of which is known without knowing the optimal solution to problem (1). To get around this difficulty, we propose finding an α-approximate solution to the problem

$$\max_{\substack{S_i \,:\, C_i(S_i) \leq b_i, \\ S_i \subset N}} \left\{ V_i(S_i) \, (R_i(S_i) - u_i) \right\} \tag{5}$$

for all $(u_i, b_i) \in \Re_+^2$. As a function of (u_i, b_i), we let $\hat{S}_i^\alpha(u_i, b_i)$ be an α-approximate solution to problem (5) and propose using $\{\hat{S}_i^\alpha(u_i, b_i) : (u_i, b_i) \in \Re_+^2\}$ as a collection of candidate assortments for nest i. In this case, Lemma 4 immediately implies that we can construct an α-approximate solution to problem (1) by picking one assortment from this collection for each nest i. To see this implication, observe that $\hat{S}_i^\alpha(u_i^*, b_i^*) \in \{\hat{S}_i^\alpha(u_i, b_i) : (u_i, b_i) \in \Re_+^2\}$. Also, by the definition of $\hat{S}_i^\alpha(u_i^*, b_i^*)$, this assortment satisfies (4) with $\hat{S}_i = \hat{S}_i^\alpha(u_i^*, b_i^*)$. Furthermore, $C_i(\hat{S}_i^\alpha(u_i^*, b_i^*)) \leq b_i^*$ by the definition of $\hat{S}_i^\alpha(u_i^*, b_i^*)$. So, the assortment $(\hat{S}_1^\alpha(u_i^*, b_i^*), \ldots, \hat{S}_m^\alpha(u_i^*, b_i^*))$ satisfies the assumptions of Lemma 4, in which case, this lemma implies that $(\hat{S}_1^\alpha(u_i^*, b_i^*), \ldots, \hat{S}_m^\alpha(u_i^*, b_i^*))$ is an α-approximate solution to problem (1). Thus, we can pick one assortment from $\{\hat{S}_i^\alpha(u_i, b_i) : (u_i, b_i) \in \Re_+^2\}$ for each nest i to stitch together an α-approximate solution to problem (1), as desired.

The next theorem records the crucial portion of the discussion in the paragraph above.

Theorem 5 *Assume that the collection of candidate assortments $\mathcal{A}_i$ includes an α-approximate solution to problem (5) for any $(u_i, b_i) \in \Re^2_+$. Then, there exists an assortment $(\hat{S}_1, \ldots \hat{S}_m)$ such that $\hat{S}_i \in \mathcal{A}_i$ for all $i \in M$, $\sum_{i \in M} C_i(\hat{S}_i) \leq c$ and $\alpha\, \Pi(\hat{S}_1, \ldots, \hat{S}_m) \geq z^*$.*

In other words, if the collection $\mathcal{A}_i$ includes an α-approximate solution to problem (5) for any $(u_i, b_i) \in \Re_+$, then we can use the collections $\mathcal{A}_1, \ldots, \mathcal{A}_m$ to stitch together an α-approximate solution to problem (1). Thus, the theorem above shows that the job of identifying a collection of candidate assortments $\mathcal{A}_i$ for nest i reduces to the job of identifying an α-approximate solution to problem (5) for all $(u_i, b_i) \in \Re^2_+$.

4 Cardinality Constraint

In this section, we use the results in the previous two sections to develop a tractable method that obtains an optimal solution to problem (1) under a cardinality constraint. Our analysis is in two stages. In Section 4.1, we assume that we already have the collections of candidate assortments $\mathcal{A}_1, \ldots, \mathcal{A}_m$ such that we can stitch together an α-approximate solution to problem (1) by picking one assortment from each one of these collections. In this case, we proceed to showing that we can solve a linear program with $O(m^2 n)$ decision variables and $\sum_{i \in M} O(mn|\mathcal{A}_i|)$ constraints to find $\hat{z}$ that satisfies $v_0\, \hat{z} = f(\hat{z})$. Furthermore, we show that we can solve problem (2) for a particular value of z with negligible computational effort, in which case, noting Theorem 1, we can solve problem (2) with $z = \hat{z}$ to obtain an α-approximate solution to problem (1). These results imply that we can obtain an α-approximate solution to problem (1) by solving a linear program with $O(m^2 n)$ decision variables and $\sum_{i \in M} O(mn|\mathcal{A}_i|)$ constraints.

In Section 4.2, on the other hand, we show that we can come up with collections of candidate assortments $\mathcal{A}_1, \ldots, \mathcal{A}_m$ such that we can stitch together an optimal, or 1-approximate solution, to problem (1) by picking one assortment from each one of these collections. Furthermore, each one of the collections $\mathcal{A}_1, \ldots, \mathcal{A}_m$ satisfies $|\mathcal{A}_i| = O(n^3)$. Thus, noting the discussion in the previous paragraph, we can obtain an optimal solution to problem (1) by solving a linear program with $O(m^2 n)$ decision variables and $\sum_{i \in M} O(mn|\mathcal{A}_i|) = O(m^2 n^4)$ constraints.

4.1 Finding Fixed Point

Our goal is to show that we can solve problem (2) in a tractable fashion and we can find $\hat{z}$ that satisfies $v_0\, \hat{z} = f(\hat{z})$ by solving a linear program, when the capacity consumption $C_i(S_i)$ corresponds to the cardinality of the assortment S_i. To solve problem (2) in a tractable fashion, we observe that the objective function of this problem is separable by the nests. If we offer the assortment S_i in nest i, then we obtain a contribution of $V_i(S_i)^{\gamma_i}(R_i(S_i) - z)$ and problem (2) finds one assortment to offer in each nest to maximize the total contribution subject to the constraint that the total cardinality of the assortments offered over all nests does not exceed c. Therefore, we can solve problem (2) by using a dynamic program. In this dynamic program, the decision epochs correspond to the nests. The state variable in each decision epoch is the remaining capacity left from the earlier nests just before choosing the assortment offered in a particular nest. Finally, the action variable is the assortment offered in a particular nest. Thus, for a particular value of z, we can obtain $f(z)$ by solving the dynamic program

$$J_i(b \,|\, z) = \max_{\substack{S_i \,:\, C_i(S_i) \le b \\ S_i \in \mathcal{A}_i}} \left\{ V_i(S_i)^{\gamma_i} (R_i(S_i) - z) + J_{i+1}(b - C_i(S_i) \,|\, z) \right\}, \tag{6}$$

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

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

$$\min \quad \Theta_1(c) \tag{7}$$

$$\text{st} \quad \Theta_i(b) \ge V_i(S_i)^{\gamma_i}(R_i(S_i) - z) + \Theta_{i+1}(b - C_i(S_i)) \quad \forall\, i \in M,\ b = 0, \ldots, mn,\ S_i \in \mathcal{F}_i(b) \tag{8}$$

$$\Theta_1(c) = v_0\, z, \tag{9}$$

to find z satisfying $v_0\, z = f(z)$. The decision variables are $\Theta = \{\Theta_i(b) : i \in M,\ b = 0, \ldots, mn\}$ and z in the linear program above. We use the convention that $\Theta_{m+1}(b) = 0$ for all $b = 0, \ldots, mn$. The set $\mathcal{F}_i(b)$ is given by $\mathcal{F}_i(b) = \{S_i : C_i(S_i) \le b,\ S_i \in \mathcal{A}_i\}$, capturing the set of feasible actions at decision epoch i and state b. If we drop the second constraint in problem (7)-(9) and minimize the objective function subject to the first set of constraints for a fixed value of z, then it is well-known that the optimal value of the decision variable $\Theta_1(c)$ gives the value function $J_1(b \,|\, z)$ computed through the dynamic program in (6); see Puterman (1994). Interestingly, if we solve problem (7)-(9) as formulated, then the optimal value of the decision variable z gives the value of z satisfying $v_0\, z = f(z)$. We show this result in the next lemma.

Lemma 6 *Letting* $(\hat{\Theta}, \hat{z})$ *be an optimal solution to problem* (7)-(9)*,* $\hat{z}$ *satisfies* $v_0\, \hat{z} = f(\hat{z})$*.*

Proof. We let $(\hat{S}_1, \ldots, \hat{S}_m)$ be an optimal solution to problem (2) when we solve this problem with $z = \hat{z}$. We define $\{\hat{b}_i : i \in M\}$ as $\hat{b}_1 = c$ and $\hat{b}_{i+1} = \hat{b}_i - C_i(\hat{S}_i)$ so that $\hat{b}_i$ corresponds to the total capacity consumption of the assortment $(\hat{S}_1, \ldots, \hat{S}_m)$ in nests $1, \ldots, i-1$. Since $(\hat{\Theta}, \hat{z})$ is a feasible solution to problem (7)-(9), it satisfies the first set of constraints for state and action $(\hat{b}_i, \hat{S}_i)$ for all $i \in M$. So, we have $\hat{\Theta}_i(\hat{b}_i) \geq V_i(\hat{S}_i)^{\gamma_i}\,(R_i(\hat{S}_i) - \hat{z}) + \hat{\Theta}_{i+1}(\hat{b}_i - C_i(\hat{S}_i))$ for all $i \in M$. Adding these inequalities gives $v_0\,\hat{z} = \hat{\Theta}_1(c) \geq \sum_{i \in M} V_i(\hat{S}_i)^{\gamma_i}\,(R_i(\hat{S}_i) - \hat{z}) = f(\hat{z})$, where the first equality uses the fact that $(\hat{\Theta}, \hat{z})$ satisfies the second constraint in problem (7)-(9) and the second equality follows from the definitions of $(\hat{S}_1, \ldots, \hat{S}_m)$ and $\hat{b}_i$. So, we have $v_0\,\hat{z} \geq f(\hat{z})$. To get a contradiction to the lemma, assume that $v_0\,\hat{z} > f(\hat{z})$ in the rest of the proof and let $\tilde{z}$ be such that $v_0\,\tilde{z} = f(\tilde{z})$.

Compute the value functions $J(\tilde{z}) = \{J_i(b \,|\, \tilde{z}) : i \in M,\ b = 0, \ldots, mn\}$ through the dynamic program in (6) with $z = \tilde{z}$. Noting the way the value functions are computed in (6), $(J(\tilde{z}), \tilde{z})$ satisfies the first set of constraints in problem (7)-(9). Also, we know that $J_1(c \,|\, \tilde{z})$ provides the optimal objective value of problem (2) when this problem is solved with $z = \tilde{z}$, so that $J_1(c \,|\, \tilde{z}) = f(\tilde{z}) = v_0\,\tilde{z}$. Thus, $(J(\tilde{z}), \tilde{z})$ satisfies the second constraint in problem (7)-(9) as well. The optimal objective value of problem (7)-(9) must be smaller than the objective value at the feasible solution $(J(\tilde{z}), \tilde{z})$, implying $v_0\,\hat{z} = \hat{\Theta}_1(c) \leq J_1(c \,|\, \tilde{z}) = v_0\,\tilde{z}$. So, we obtain $f(\hat{z}) < v_0\,\hat{z} \leq v_0\,\tilde{z} = f(\tilde{z})$, but since $f(\cdot)$ is decreasing, we cannot have $v_0\,\hat{z} \leq v_0\,\tilde{z}$ and $f(\hat{z}) < f(\tilde{z})$, yielding a contradiction. $\square$

Thus, we can solve the linear program in (7)-(9) to obtain $\hat{z}$ satisfying $v_0\,\hat{z} = f(\hat{z})$. Noting that $|\mathcal{F}_i(b)| \leq |\mathcal{A}_i|$, there are $O(m^2 n)$ decision variables and $\sum_{i \in M} O(mn|\mathcal{A}_i|)$ constraints in this linear program. Once we have $\hat{z}$, noting Theorem 1, we can solve problem (2) with $z = \hat{z}$ to obtain an α-approximate solution to problem (1), as long as the collections $\mathcal{A}_1, \ldots, \mathcal{A}_m$ are such that they allow us stitching together an α-approximate solution to problem (1). To solve problem (2), we can use the dynamic program in (6) and the computational effort for solving this dynamic program is negligible when compared with that for solving the linear program in (7)-(9).

4.2 Constructing Candidate Assortments

Here, our goal is to show how to construct collections of candidate assortments $\mathcal{A}_1, \ldots, \mathcal{A}_m$ such that we can stitch together an optimal, or 1-approximate, solution to problem (1) by choosing one assortment from each one of these collections. We build on Theorem 5 for this purpose. In particular, we construct the collection of candidate assortments $\mathcal{A}_i$ such that $\mathcal{A}_i$ includes an optimal solution to problem (5) for any $(u_i, b_i) \in \Re^2_+$. In this case, by Theorem 5, we can indeed use the collections $\mathcal{A}_1, \ldots, \mathcal{A}_m$ to the stitch together an optimal solution to problem (1).

To characterize the optimal solution to problem (5) for any $(u_i, b_i) \in \Re^2_+$, using the definitions of $V_i(S_i)$ and $R_i(S_i)$, we write its objective function as $\sum_{j \in S_i} v_{ij}\,(r_{ij} - u_i)$. So, using the decision variables $x_i = (x_{i1}, \ldots, x_{in}) \in \{0,1\}^n$, we write problem (5) under a cardinality constraint as

$$\max_{\substack{x_i \,:\, \sum_{j \in N} x_{ij} \leq b_i, \\ x_i \in \{0,1\}^n}} \left\{ \sum_{j \in N} v_{ij}\,(r_{ij} - u_i)\,x_{ij} \right\}, \tag{10}$$

which is a knapsack problem where each item occupies one unit of space. In this knapsack problem, the utility of item j is $v_{ij}\,(r_{ij} - u_i)$ and the capacity of the knapsack is b_i. For any $(u_i, b_i) \in \Re^2_+$, we can solve problem (10) by ordering the items according to their utilities and filling the knapsack starting from the item with the largest utility. The parameter u_i determines the ordering of the utilities of the items, whereas b_i determines the number of items that we can put into the knapsack. To characterize that optimal solution to problem (5) for any $(u_i, b_i) \in \Re^2_+$, we observe that we can come up with only $O(n^2)$ different orderings of the utilities of the items as u_i takes values over $\Re_+$. To see this result, we define n linear functions $\{h_{ij}(\cdot) : j \in N\}$ as $h_{ij}(u_i) = v_{ij}\,(r_{ij} - u_i)$, capturing the utility of item j in the problem above. These n linear functions intersect at $O(n^2)$ points and these points can be found by solving $h_{ij}(u_i) = h_{ij'}(u_i)$ for u_i for all distinct $j, j' \in N$. The intersection points computed in this fashion correspond to the only values of u_i where the ordering of the utilities of the items can change and the desired result follows. In Figure 1, we show a possible case with $n = 3$. The bold lines show the linear functions $\{h_{ij}(\cdot) : j \in N\}$ and the white circles show the pairwise intersection points of them. Each one of the intervals between these intersection points is associated with a particular ordering of the linear functions. In Figure 1, these orderings are $\{h_{i1}(\cdot), h_{i2}(\cdot), h_{i3}(\cdot)\}$, $\{h_{i2}(\cdot), h_{i1}(\cdot), h_{i3}(\cdot)\}$, $\{h_{i2}(\cdot), h_{i3}(\cdot), h_{i1}(\cdot)\}$ and $\{h_{i3}(\cdot), h_{i2}(\cdot), h_{i1}(\cdot)\}$.

By the discussion above, as u_i takes values over $\Re_+$, we can come up with only $O(n^2)$ different orderings of the utilities of the items in problem (10). We denote these orderings by $\{\sigma_i^g : g \in \mathcal{G}_i\}$ with $|\mathcal{G}_i| = O(n^2)$. We let $x_i^k(\sigma_i^g) \in \{0,1\}^n$ be a possible solution to problem (10) obtained by ordering the items according to the ordering σ_i^g and putting the first k items into the knapsack by following this ordering. For any $(u_i, b_i) \in \Re^2_+$, the crucial observation is that an optimal solution to problem (10) must be one of the solutions $\{x_i^k(\sigma_i^g) : g \in \mathcal{G}_i,\ k = 0, \ldots, n\}$, since the optimal solution to this problem can be obtained by ordering the items according to their utilities and putting a certain number of items by following this ordering. We use $S_i^k(\sigma_i^g)$ to denote the assortment corresponding to the solution $x_i^k(\sigma_i^g)$, given by $S_i^k(\sigma_i^g) = \{j \in N : x_{ij}^k(\sigma_i^g) = 1\}$, including the products used in the solution $x_i^k(\sigma_i^g)$. Thus, letting $\mathcal{A}_i = \{S_i^k(\sigma_i^g) : g \in \mathcal{G}_i,\ k = 0, \ldots, n\}$, this collection includes an optimal solution to problem (10) for any $(u_i, b_i) \in \Re^2_+$. Furthermore, noting that $|\mathcal{G}_i| = O(n^2)$, there are $O(n^3)$ assortments in the collection $\mathcal{A}_i$.

To sum up, as u_i takes values over $\Re_+$, there are only $O(n^2)$ different orderings of the utilities of the items in problem (10) and these orderings can be identified by finding the pairwise intersection points of the linear functions $\{h_{ij}(\cdot) : j \in N\}$. We construct the solution $x_i^k(\sigma_i^g)$ by putting the first k items into the knapsack according to the ordering of the utilities σ_i^g. By the discussion above, the collection $\mathcal{A}_i = \{S_i^k(\sigma_i^g) : g \in \mathcal{G}_i,\ k = 0, \ldots, n\}$ includes an optimal solution to problem (5) for any $(u_i, b_i) \in \Re^2_+$. In this case, by Theorem 5, we can stitch together an optimal solution to problem (1) by picking one assortment from each one of the collections $\mathcal{A}_1, \ldots, \mathcal{A}_m$. Furthermore, we have $|\mathcal{A}_i| = O(n^3)$. Also, the discussion in Section 4.1 shows that if we can stitch together an α-approximate solution to problem (1) by picking one assortment from each one of the collections $\mathcal{A}_1, \ldots, \mathcal{A}_m$, then we can obtain an α-approximate solution to problem (1) by solving a linear program with $O(m^2 n)$ decision variables and $\sum_{i \in M} O(mn|\mathcal{A}_i|)$ constraints. Combining these results, it follows that we can obtain an optimal solution to problem (1) by solving a linear program with $O(m^2 n)$ decision variables and $O(m^2 n^4)$ constraints. Gallego and Topaloglu (2014) generate candidate assortments similarly, but our approach deals with capacity constraints that link different nests.

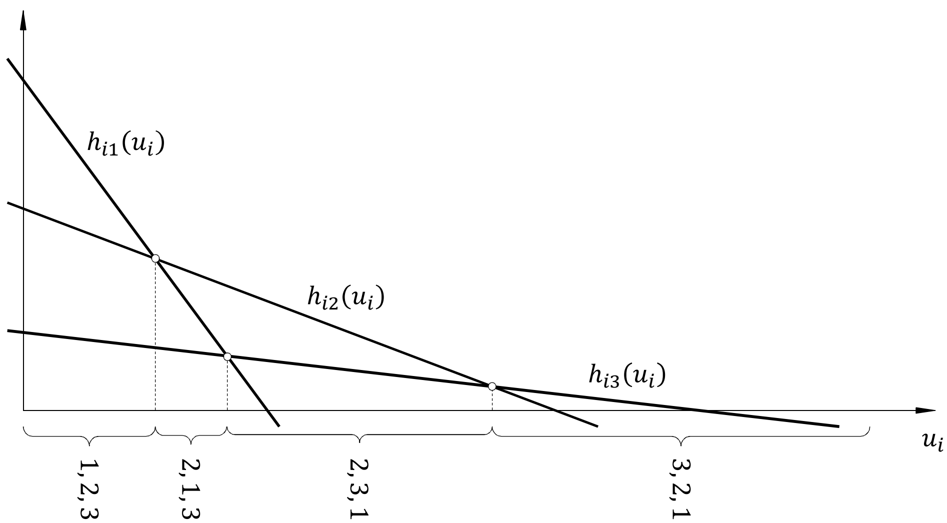

Figure 1: The linear functions $\{h_{ij}(\cdot) : j \in N\}$ and their pairwise intersection points for a possible case with $n = 3$.

5 Space Constraint

In this section, we use the results in Section 2 and 3 to obtain a 4-approximate solution to problem (1) under a space constraint. Similar to our analysis under a cardinality constraint, our analysis proceeds in two stages. In Section 5.1, we assume that we have access to collections of candidate assortments $\mathcal{A}_1, \ldots, \mathcal{A}_m$ that allow us to stitch together an α-approximate solution to problem (1). Without loss of generality, we assume that $C_i(S_i) \leq c$ for all $S_i \in \mathcal{A}_i$ and $i \in M$. Otherwise, such assortments are never used in a feasible solution to problem (1). We define an appropriate relaxation $f^R(\cdot)$ of $f(\cdot)$. Working with this relaxation, we show that we can solve a linear program with $O(m)$ decision variables and $\sum_{i \in M} O(|\mathcal{A}_i|)$ constraints to find $\hat{z}$ that satisfies $v_0 \, \hat{z} = f^R(\hat{z})$. Once we have such $\hat{z}$, we show that we can find an assortment $(\hat{S}_1, \ldots, \hat{S}_m)$ that satisfies the inequality in Corollary 2 with $\beta = 2$ and this assortment can be computed with negligible computational effort. In view of Corollary 2, this assortment provides a 2α-approximate solution to problem (1). These results imply that we can obtain a 2α-approximate solution to problem (1) by solving a linear program with $O(m)$ decision variables and $\sum_{i \in M} O(|\mathcal{A}_i|)$ constraints.

In Section 5.2, on the other hand, we show how to construct the collections of candidate assortments $\mathcal{A}_1, \ldots, \mathcal{A}_m$ such that we can stitch together a 2-approximate solution to problem (1) by picking one assortment from each one of these collections. Furthermore, each one of these collections of candidate assortments satisfy $|\mathcal{A}_i| = O(n^4)$. Thus, in view of the discussion in the previous paragraph, we can solve a linear program with $O(m)$ decision variables and $\sum_{i \in M} O(|\mathcal{A}_i|) = O(mn^4)$ constraints to obtain a 4-approximate solution to problem (1) under a space constraint.

5.1 Finding Fixed Point

Here, we define an appropriate relaxation $f^R(\cdot)$ of $f(\cdot)$ under a space constraint and show that we can solve a linear program to find $\hat{z}$ that satisfies $v_0\,\hat{z} = f^R(\hat{z})$. Furthermore, we show that there exists an assortment $(\hat{S}_1, \ldots, \hat{S}_m)$ that satisfies the inequality in Corollary 2 with $\beta = 2$ and this assortment can be computed efficiently. To define a relaxation $f^R(\cdot)$ of $f(\cdot)$, we use the linear programing relaxation of problem (2). In particular, using the decision variables $x = \{x_i(S_i) : i \in M,\ S_i \in \mathcal{A}_i\}$, we define $f^R(\cdot)$ as

$$
\begin{aligned}
f^R(z) \quad = \quad \max \quad & \sum_{i\in M}\sum_{S_i\in\mathcal{A}_i} V_i(S_i)^{\gamma_i}(R_i(S_i) - z)\,x_i(S_i) && (11)\\
\text{st} \quad & \sum_{i\in M}\sum_{S_i\in\mathcal{A}_i} C_i(S_i)\,x_i(S_i) \le c && (12)\\
& \sum_{S_i\in\mathcal{A}_i} x_i(S_i) = 1 \qquad \forall\, i \in M && (13)\\
& x_i(S_i) \ge 0 \qquad \forall\, i \in M,\ S_i \in \mathcal{A}_i. && (14)
\end{aligned}
$$

In the problem above, the decision variable $x_i(S_i)$ takes value one if we offer the assortment S_i in nest i and takes value zero otherwise. The first constraint ensures that the total capacity consumption of the assortments offered in all nests do not violate the capacity availability. The second set of constraints ensure that we choose one assortment in each nest. If we impose the constraint $x_i(S_i) \in \mathbb{Z}_+$ for all $i \in M$, $S_i \in \mathcal{A}_i$ in problem (11)-(14), then this problem would find one assortment to offer in each nest subject to the capacity constraint, in which case, it would be equivalent to problem (2). However, the way problem (11)-(14) is formulated, it is a relaxation of problem (2) and we have $f^R(z) \ge f(z)$ as desired. Considering the question of how to find $\hat{z}$ satisfying $v_0\,\hat{z} = f^R(\hat{z})$, we make use of the dual of problem (11)-(14) for this purpose. Associating the dual variables Δ and $y = \{y_i : i \in M\}$ respectively with the two sets of constraints in the problem above, we propose solving the linear program

$$
\begin{aligned}
\min \quad & c\,\Delta + \sum_{i\in M} y_i && (15)\\
\text{st} \quad & C_i(S_i)\,\Delta + y_i \ge V_i(S_i)^{\gamma_i}(R_i(S_i) - z) \qquad \forall\, i \in M,\ S_i \in \mathcal{A}_i && (16)\\
& c\,\Delta + \sum_{i\in M} y_i = v_0\, z && (17)\\
& \Delta \ge 0,\ y_i \text{ is free},\ z \text{ is free} \qquad \forall\, i \in M && (18)
\end{aligned}
$$

to find $\hat{z}$ satisfying $v_0\,\hat{z} = f^R(\hat{z})$. The decision variables are Δ, y and z in the problem above. If we drop the second constraint in problem (15)-(18) and minimize the objective function subject to the first set of constraints for a fixed value of z, then this problem corresponds to the dual of problem (11)-(14). With the second constraint added, problem (15)-(18) allows us to find $\hat{z}$ satisfying $v_0\,\hat{z} = f^R(\hat{z})$, as shown in the next lemma. The proof of this lemma follows from an argument similar to the proof of Lemma 6 and we defer it to the appendix.

Lemma 7 *Letting* $(\hat{\Delta}, \hat{y}, \hat{z})$ *be an optimal solution to problem* (15)-(18), $\hat{z}$ *satisfies* $v_0\,\hat{z} = f^R(\hat{z})$.

Lemma 7 shows that we can find $\hat{z}$ satisfying $v_0\,\hat{z} = f^R(\hat{z})$ by solving a linear program with $O(m)$ decision variables and $\sum_{i \in M} O(|\mathcal{A}_i|)$ constraints. Next, we shift our attention to constructing an assortment $(\hat{S}_1, \ldots, \hat{S}_m)$ that satisfies the inequality in Corollary 2 with $\beta = 2$. In our initial analysis, we show how to obtain an assortment satisfying the inequality in Corollary 2 with $\beta = 3$. After this result, we tighten our analysis to get $\beta = 2$.

It is a simple exercise in linear programming duality to show that any basic optimal solution to problem (11)-(14) includes at most two fractional components; see Sinha and Zoltners (1979). We let $\hat{x}$ be a basic optimal solution to problem (11)-(14) when we solve this problem with $z = \hat{z}$. We make two observations. First, if $\hat{x}_{i'}(P_{i'}) \in (0,1]$ for some nest $i' \in M$ and assortment $P_{i'} \in \mathcal{A}_{i'}$, then noting the second set of constraints in problem (11)-(14), there must be some other assortment $Q_{i'} \in \mathcal{A}_{i'}$ such that $\hat{x}_{i'}(Q_{i'}) \in [0,1)$ as well. Second, since $\hat{x}$ has as most two fractional components, there can be no other fractional component of $\hat{x}$. In this case, noting the second set of constraints in problem (11)-(14) once more, it follows that for each nest $i \in M \setminus \{i'\}$, there exists a single assortment $\tilde{S}_i$ such that $\hat{x}_i(\tilde{S}_i) = 1$. Therefore $\{\hat{x}_i(\tilde{S}_i) : i \in M \setminus \{i'\}\} \cup \{\hat{x}_{i'}(P_{i'})\} \cup \{\hat{x}_{i'}(Q_{i'}\}$ includes all components of $\hat{x}$ taking strictly positive values. Using the solution $\hat{x}$, we construct three assortments $(\hat{S}^0_1, \ldots, \hat{S}^0_m)$, $(\hat{S}^1_1, \ldots, \hat{S}^1_m)$ and $(\hat{S}^2_1, \ldots, \hat{S}^2_m)$ as follows. The first one of these assortments is constructed as $(\hat{S}^0_1, \ldots, \hat{S}^0_m) = (\tilde{S}_1, \ldots \tilde{S}_{i'-1}, \emptyset, \tilde{S}_{i'+1}, \ldots, \tilde{S}_m)$. In other words, the assortment $(\hat{S}^0_1, \ldots, \hat{S}^0_m)$ uses the components of the solution $\hat{x}$ that take value one and this assortment simply uses the empty subset in nest i'. On the other hand, we construct the second and third assortments as $(\hat{S}^1_1, \ldots, \hat{S}^1_m) = (\emptyset, \ldots, \emptyset, P_{i'}, \emptyset, \ldots, \emptyset)$ and $(\hat{S}^2_1, \ldots, \hat{S}^2_m) = (\emptyset, \ldots, \emptyset, Q_{i'}, \emptyset, \ldots, \emptyset)$. Thus, each one of the assortments $(\hat{S}^1_1, \ldots, \hat{S}^1_m)$ and $(\hat{S}^2_1, \ldots, \hat{S}^2_m)$ only uses each one of the two potentially fractional components of the solution $\hat{x}$. The important observation is that the three assortments $(\hat{S}^0_1, \ldots, \hat{S}^0_m)$, $(\hat{S}^1_1, \ldots, \hat{S}^1_m)$ and $(\hat{S}^2_1, \ldots, \hat{S}^2_m)$ as defined above collectively include all components of the solution $\hat{x} = \{\hat{x}_i(S_i) : i \in M,\ S_i \in \mathcal{A}_i\}$ that take a strictly positive value. In this case, for the value $\hat{z}$ satisfying $v_0\,\hat{z} = f^R(\hat{z})$, we obtain

$$\begin{aligned}
v_0\,\hat{z} = f^R(\hat{z}) &= \sum_{i \in M} \sum_{S_i \in \mathcal{A}_i} V_i(S_i)^{\gamma_i} (R_i(S_i) - \hat{z})\,\hat{x}_i(S_i) \\
&\leq \sum_{i \in M} V_i(\hat{S}^0_i)^{\gamma_i} (R_i(\hat{S}^0_i) - \hat{z}) + \sum_{i \in M} V_i(\hat{S}^1_i)^{\gamma_i} (R_i(\hat{S}^1_i) - \hat{z}) + \sum_{i \in M} V_i(\hat{S}^2_i)^{\gamma_i} (R_i(\hat{S}^2_i) - \hat{z}) \\
&\leq 3 \max \left\{ \sum_{i \in M} V_i(\hat{S}^0_i)^{\gamma_i} (R_i(\hat{S}^0_i) - \hat{z}), \sum_{i \in M} V_i(\hat{S}^1_i)^{\gamma_i} (R_i(\hat{S}^1_i) - \hat{z}), \sum_{i \in M} V_i(\hat{S}^2_i)^{\gamma_i} (R_i(\hat{S}^2_i) - \hat{z}) \right\}.
\end{aligned}$$

In the chain of inequalities above, the second equality is by the fact that $\hat{x}$ is an optimal solution to problem (11)-(14) when this problem is solved with $z = \hat{z}$. The first inequality is by the fact that the three assortments $(\hat{S}^0_1, \ldots, \hat{S}^0_m)$, $(\hat{S}^1_1, \ldots, \hat{S}^1_m)$ and $(\hat{S}^2_1, \ldots, \hat{S}^2_m)$ as defined above collectively include all strictly positive components of the solution $\hat{x}$, implying that if $\hat{x}_i(S_i) > 0$ for some $i \in M$ and $S_i \in \mathcal{A}_i$, then it is the case that $S_i = \hat{S}^0_i$ or $S_i = \hat{S}^1_i$ or $S_i = \hat{S}^2_i$. The chain of inequalities above show that one of the assortments $(\hat{S}^0_1, \ldots, \hat{S}^0_m)$, $(\hat{S}^1_1, \ldots, \hat{S}^1_m)$ and $(\hat{S}^2_1, \ldots, \hat{S}^2_m)$ satisfies the inequality in Corollary 2 with $\beta = 3$, as desired. In the remainder of this section, we tighten our analysis to construct an assortment satisfying the inequality in Corollary 2 with $\beta = 2$.

As mentioned above, if the basic optimal solution $\hat{x}$ to problem (11)-(14) has two fractional components, then there exist some nest $i' \in M$ and assortments $P_{i'}, Q_{i'} \in \mathcal{A}_{i'}$ such that $\hat{x}_{i'}(P_{i'}), \hat{x}_{i'}(Q_{i'}) \in (0,1)$ and there is no other fractional component of $\hat{x}$. Without loss of generality, we assume that $C_{i'}(P_{i'}) \leq C_i(Q_{i'})$. Furthermore, for each nest $i \in M \setminus \{i'\}$, there exists a single assortment $\tilde{S}_i$ such that $\hat{x}_i(\tilde{S}_i) = 1$. Using the solution $\hat{x}$, we construct two assortments $(\hat{S}^1_1, \ldots, \hat{S}^1_m)$ and $(\hat{S}^2_1, \ldots, \hat{S}^2_m)$ as follows. The first one of these assortments is constructed as $(\hat{S}^1_1, \ldots, \hat{S}^1_m) = (\tilde{S}_1, \ldots \tilde{S}_{i'-1}, P_{i'}, \tilde{S}_{i'+1}, \ldots, \tilde{S}_m)$. In other words, the assortment $(\hat{S}^1_1, \ldots, \hat{S}^1_m)$ uses the components of the solution $\hat{x}$ that take value one, along with the fractional component of the solution $\hat{x}$ with the smaller capacity consumption. On the other hand, we construct the second assortment as $(\hat{S}^2_1, \ldots, \hat{S}^2_m) = (\emptyset, \ldots, \emptyset, Q_{i'}, \emptyset, \ldots, \emptyset)$, offering the subset $Q_{i'}$ in nest i', but offering empty subsets in all of the other nests. Once again, the crucial observation is that the two assortments $(\hat{S}^1_1, \ldots, \hat{S}^1_m)$ and $(\hat{S}^2_1, \ldots, \hat{S}^2_m)$ as defined above collectively include all components of the solution $\hat{x} = \{\hat{x}_i(S_i) : i \in M,\ S_i \in \mathcal{A}_i\}$ taking a strictly positive value. In this case, we get

$$\begin{aligned}
v_0\, \hat{z} = f^R(\hat{z}) &= \sum_{i \in M} \sum_{S_i \in \mathcal{A}_i} V_i(S_i)^{\gamma_i} (R_i(S_i) - \hat{z})\, \hat{x}_i(S_i) \\
&\leq \sum_{i \in M} V_i(\hat{S}^1_i)^{\gamma_i} (R_i(\hat{S}^1_i) - \hat{z}) + \sum_{i \in M} V_i(\hat{S}^2_i)^{\gamma_i} (R_i(\hat{S}^2_i) - \hat{z}) \\
&\leq 2 \max \left\{ \sum_{i \in M} V_i(\hat{S}^1_i)^{\gamma_i} (R_i(\hat{S}^1_i) - \hat{z}), \sum_{i \in M} V_i(\hat{S}^2_i)^{\gamma_i} (R_i(\hat{S}^2_i) - \hat{z}) \right\}.
\end{aligned}$$

The first inequality is, once again, by the fact that if $\hat{x}_i(S_i) > 0$ for some $i \in M$ and $S_i \in \mathcal{A}_i$, then we have $S_i = \hat{S}^1_i$ or $\hat{S}_i = \hat{S}^2_i$. Thus, the chain of inequalities above shows that one of the assortments $(\hat{S}^1_1, \ldots, \hat{S}^1_m)$ and $(\hat{S}^2_1, \ldots, \hat{S}^2_m)$ satisfies the inequality in Corollary 2 with $\beta = 2$, as desired. Furthermore, we observe that both of the solutions $(\hat{S}^1_1, \ldots, \hat{S}^1_m)$ and $(\hat{S}^2_1, \ldots, \hat{S}^2_m)$ are feasible to problem (1). It is simple to see the feasibility of the solution $(\hat{S}^2_1, \ldots, \hat{S}^2_m)$ since this assortment only offers $Q_{i'}$ in nest i', in which case, we have $\sum_{i \in M} C_i(\hat{S}^2_i) = C_{i'}(Q_{i'}) \leq c$, where the last inequality uses the assumption at the beginning of this section that $C_i(S_i) \leq c$ for all $i \in M$, $S_i \in \mathcal{A}_i$. To see the feasibility of the solution $(\hat{S}^1_1, \ldots, \hat{S}^1_m)$ to problem (1), we observe that $\sum_{i \in M} C_i(\hat{S}^1_i) = \sum_{i \in M \setminus \{i'\}} C_i(\tilde{S}_i) + C_{i'}(P_{i'}) \leq \sum_{i \in M \setminus \{i'\}} C_i(\tilde{S}_i) + C_{i'}(P_{i'})\, \hat{x}_{i'}(P_{i'}) + C_{i'}(Q_{i'})\, \hat{x}_{i'}(Q_{i'}) = \sum_{i \in M} \sum_{S_i \in \mathcal{A}_i} C_i(S_i)\, \hat{x}_i(S_i) \leq c$, where the first inequality uses the fact that $C_{i'}(P_{i'}) \leq C_{i'}(Q_{i'})$ and $\hat{x}_{i'}(P_{i'}) + \hat{x}_{i'}(Q_{i'}) = 1$ by the second set of constraints in problem (11)-(14) and the second equality uses the fact that $\{\hat{x}_i(\tilde{S}_i) : i \in M \setminus \{i'\}\} \cup \{\hat{x}_{i'}(P_{i'})\} \cup \{\hat{x}_{i'}(Q_{i'}\}$ are all of the components of $\hat{x}$ taking strictly positive values.

To sum up the discussion so far in this section, assume that we can stitch together an α-approximate solution to problem (1) by picking one assortment from each one of the collections $\mathcal{A}_1, \ldots, \mathcal{A}_m$. We can find $\hat{z}$ satisfying $v_0\, \hat{z} = f^R(\hat{z})$ by solving problem (15)-(18). Once we have

$\hat{z}$, we can solve problem (11)-(14) with $z = \hat{z}$. Denoting the optimal solution to the latter problem by $\hat{x}$, we can construct the two assortments $(\hat{S}_1^1, \ldots, \hat{S}_m^1)$ and $(\hat{S}_1^2, \ldots, \hat{S}_m^2)$ as described in the previous paragraph. We know that one of these assortments satisfies the inequality in Corollary 2 with $\beta = 2$, in which case, Corollary 2 implies that $2\alpha\,\Pi(\hat{S}_1^1, \ldots, \hat{S}_m^1) \geq z^*$ or $2\alpha\,\Pi(\hat{S}_1^2, \ldots, \hat{S}_m^2) \geq z^*$. Furthermore, both of these assortments provide a feasible solution to problem (1). Thus, one of the assortments $(\hat{S}_1^1, \ldots, \hat{S}_m^1)$ and $(\hat{S}_1^2, \ldots, \hat{S}_m^2)$ is a 2α-approximate solution to problem (1). By checking the expected revenue from each one of these assortments and picking the better one, we obtain a 2α-approximate solution to problem (1). These results indicate that if the collections $\mathcal{A}_1, \ldots, \mathcal{A}_m$ allow us to stitch together an α-approximate solution to problem (1), then we can obtain a 2α-approximate solution to problem (1) by solving a linear program with $O(m)$ decision variables and $\sum_{i \in M} O(|\mathcal{A}_i|)$ constraints.

5.2 Constructing Candidate Assortments

Our goal is to show how to construct the collections of candidate assortments $\mathcal{A}_1, \ldots, \mathcal{A}_m$ such that we can stitch together a 2-approximate solution to problem (1) by picking one assortment from each one of these collections. Noting Theorem 5, it is enough we construct a collection of candidate assortments $\mathcal{A}_i$ such that this collection includes a 2-approximate solution to problem (5) for any $(u_i, b_i) \in \Re_+^2$.

To construct an approximate solution to problem (5) for any $(u_i, b_i) \in \Re_+^2$, we use a linear programming relaxation of this problem. Using the decision variables $x_i = (x_{i1}, \ldots, x_{in}) \in \{0,1\}^n$, a linear programming relaxation of problem (5) under a space constraint is given by

$$\max_{\substack{x_i \,:\, \sum_{j \in N} w_{ij}\, x_{ij} \leq b_i, \\ 0 \leq x_{ij} \leq \mathbf{1}(w_{ij} \leq b_i) \;\; \forall\, j \in N}} \left\{ \sum_{j \in N} v_{ij}\,(r_{ij} - u_i)\, x_{ij} \right\}, \tag{19}$$

where $\mathbf{1}(\cdot)$ is the indicator function. If we impose the constraint $x_i \in \{0,1\}^n$ in the problem above, then this problem would be equivalent to problem (5), but without these constraints, the problem above is a linear programming relaxation of problem (5). Problem (19) is a knapsack problem where the utility of item j is $v_{ij}\,(r_{ij} - u_i)$, the space consumption of item j is w_{ij} and the capacity of the knapsack is b_i. The indicator function in the second set of constraints ensures that we only use the items whose capacity consumptions do not exceed b_i. We can use the following procedure to solve problem (19). We order the items according to their utility to space consumption ratios. We focus only on the items whose capacity consumptions do not exceed b_i. We fill the knapsack with these items starting from the item with the largest utility to space consumption ratio. It is also worthwhile to observe that the solution to problem (19) that we obtain in this fashion includes at most one fractional component and this component corresponds to the item with the smallest utility to space consumption ratio that we put into the knapsack. To exploit these observations, we define n linear functions $\{h_{ij}(\cdot) : j \in N\}$ as $h_{ij}(u_i) = v_{ij}\,(r_{ij} - u_i)/w_{ij}$, capturing the utility to space consumption ratio of item j in problem (19). Similar to our development under a cardinality constraint, these n linear functions intersect at $O(n^2)$ points, which implies that we can come up with only $O(n^2)$ different orderings of the utility to space consumption ratios as u_i takes values over $\Re_+$. We denote these possible orderings by $\{\sigma_i^g : g \in \mathcal{G}_i\}$ with $|\mathcal{G}_i| = O(n^2)$.

We let $x_i^k(\sigma_i^g, b_i) \in \{0,1\}^n$ be a solution to problem (19) that we obtain by focusing only on the items whose capacity consumptions do not exceed b_i, ordering these items according to the ordering σ_i^g and putting the first k of these items into the knapsack by following the ordering σ_i^g. The key observation is that if we solve problem (19) for any $(u_i, b_i) \in \Re_+$, then the integer portion of the optimal solution must be one of the solutions in $\{x_i^k(\sigma_i^g, b_i) : g \in \mathcal{G}_i,\ k = 0, \ldots, n,\ b_i \in \Re_+\}$. To see this result, it is enough to recall that the integer portion of the solution to problem (19) can be obtained by focusing on the items whose capacity consumptions do not exceed b_i, ordering these items according to their utility to space consumption ratios and filling the knapsack in the order of decreasing utility to space consumption ratios. Furthermore, without loss of generality, we can consider only $b_i \in \{w_{i1}, \ldots, w_{in}\}$, rather than $b_i \in \Re_+$, in the collection $\{x_i^k(\sigma_i^g, b_i) : g \in \mathcal{G}_i,\ k = 0, \ldots, n,\ b_i \in \Re_+\}$, since if b_i takes a value other than $\{w_{i1}, \ldots, w_{in}\}$, then we can decrease the value of b_i to the closest element in $\{w_{i1}, \ldots, w_{in}\}$ without changing the set of products whose space consumptions do not exceed b_i. Similar to our notation under a cardinality constraint, we use $S_i^k(\sigma_i^g, b_i) = \{j \in N : x_{ij}^k(\sigma_i^g, b_i) = 1\}$ to denote the assortment corresponding to the solution $x_i^k(\sigma_i^g, b_i)$, which includes the items taking value one in the solution $x_i^k(\sigma_i^g, b_i)$. Augmenting the collection of assortments $\{S_i^k(\sigma_i^g, b_i) : g \in \mathcal{G}_i,\ k = 0, \ldots, n, b_i \in \Re_+\}$ with the singleton assortments, we propose using $\mathcal{A}_i = \{S_i^k(\sigma_i^g, b_i) : g \in \mathcal{G}_i,\ k = 0, \ldots, n,\ b_i \in \Re_+\} \cup \{\{j\} : j \in N\}$ as a collection of candidate assortments for nest i. In the rest of this section, we show that this collection includes a 2-approximate solution to problem (5) for any $(u_i, b_i) \in \Re_+$.

To see this, assume we solve problem (19) with $(u_i, b_i) = (u_i', b_i')$ for arbitrary $(u_i', b_i') \in \Re_+^2$. We let g' and k' be such that $x_i^{k'}(\sigma_i^{g'}, b_i')$ is the integer portion of the optimal solution to problem (19) when this problem is solved with $(u_i, b_i) = (u_i', b_i')$. Also, we let j' be the item taking a fractional value in the optimal solution to problem (19), when there is one. Letting $\zeta^*(u_i, b_i)$ and $\xi^*(u_i, b_i)$ respectively be the optimal objective values of problems (5) and (19), we get

$$\zeta^*(u_i', b_i') \le \xi^*(u_i', b_i') \le \sum_{j \in N} v_{ij}\ (r_{ij} - u_i')\ x_{ij}^{k'}(\sigma_i^{g'}, b_i') + v_{ij'}\ (r_{ij'} - u_i')$$
$$= \sum_{j \in S_i^{k'}(\sigma_i^{g'}, b_i')} v_{ij}\ (r_{ij} - u_i') + v_{ij'}\ (r_{ij'} - u_i') \le 2 \max \left\{ \sum_{j \in S_i^{k'}(\sigma_i^{g'}, b_i')} v_{ij}\ (r_{ij} - u_i'),\ v_{ij'}\ (r_{ij'} - u_i') \right\},$$

where the first inequality is by the fact that problem (19) is a linear programming relaxation of problem (5) and the second inequality is by the fact that the solution $x_i^{k'}(\sigma_i^{g'}, b_i')$ together with the component j' includes all components taking a strictly positive value in the optimal solution to problem (19) when this problem is solved with $(u_i, b_i) = (u_i', b_i')$. Since $x_i^{k'}(\sigma_i^{g'}, b_i')$ is the integer portion of the optimal solution to problem (19) when solved with $(u_i, b_i) = (u_i', b_i')$, we have $\sum_{j \in S_i^{k'}(\sigma_i^{g'}, b_i')} w_{ij} = \sum_{j \in N} w_{ij}\, x_{ij}^{k'}(\sigma_i^{g'}, b_i') \leq b_i'$. We note that since item j' takes a fractional value in the optimal solution, it must be the case that $\mathbf{1}(w_{ij'} \leq b_i') = 1$, yielding $w_{ij'} \leq b_i'$. So, the solutions $S_i^{k'}(\sigma_i^{g'}, b_i')$ or $\{j'\}$ are both feasible to problem (5) when this problem is solved with $(u_i, b_i) = (u_i', b_i')$. Therefore, the chain of inequalities above shows that one of the assortments $S_i^{k'}(\sigma_i^{g'}, b_i')$ or $\{j'\}$ is a 2-approximate solution to problem (5). Since our choice of (u_i', b_i') in the discussion above is arbitrary, we conclude that the collection $\mathcal{A}_i = \{S_i^k(\sigma_i^g, b_i) : g \in \mathcal{G}_i,\ k = 0, \ldots, n,\ b_i \in \Re_+\} \cup \{\{j\} : j \in N\}$ includes a 2-approximate solution to problem (5) for any $(u_i, b_i) \in \Re_+$, as desired. Noting that we can focus only on $b_i \in \{w_{i1}, \ldots, w_{in}\}$ in this collection, instead of $b_i \in \Re_+$, the collection $\mathcal{A}_i$ includes $O(|\mathcal{G}_i|\, n^2) = O(n^4)$ assortments.

Summarizing our results under a space constraint, the discussion in Section 5.1 shows that if we can stitch together an α-approximate solution to problem (1) by picking one assortment from each one of the collections $\mathcal{A}_1, \ldots, \mathcal{A}_m$, then we can solve a linear program with $O(m)$ decision variables and $\sum_{i \in M} O(|\mathcal{A}_i|)$ constraints to obtain a 2α-approximate solution to problem (1). On the other hand, the development in this section shows how to construct the collections of candidate assortments $\mathcal{A}_1, \ldots, \mathcal{A}_m$, with $|\mathcal{A}_i| = O(n^4)$, such that we can stitch together a 2-approximate solution to problem (1) by using these collections. Thus, we can solve a linear program with $O(m)$ decision variables and $O(mn^4)$ constraints to get a 4-approximate solution to problem (1).

6 Upper Bound on Optimal Expected Revenue

The approach described in Section 4 obtains the optimal solution to problem (1) under a cardinality constraint, along with the optimal expected revenue. On the other hand, the approach described in Section 5 obtains a 4-approximate solution under a space constraint. An approximation guarantee of four can be comforting, showing that this approach never performs arbitrarily badly, but a guarantee of obtaining at least quarter of the optimal expected revenue may not be thoroughly satisfying from a practical perspective. In this section, our goal is to develop a tractable approach for obtaining an upper bound on the optimal expected revenue for an individual instance of problem (1) under a space constraint. By comparing this upper bound on the optimal expected revenue with the expected revenue from a particular solution, we can get a feel for the optimality gap of the specific solution on hand.

To construct an upper bound on the optimal expected revenue in problem (1), we partition the interval $[0, c]$ into K intervals $\{[b_i^{k-1}, b_i^k] : k = 1, \ldots, K\}$ with $0 = b_i^0 \leq b_i^1 \leq \ldots \leq b_i^{K-1} \leq b_i^K = c$. Using the decision variables $x_i = (x_{i1}, \ldots, x_{in}) \in [0,1]^n$, we define $\phi_i^k(z)$ as

$$\phi_i^k(z) = \max_{\substack{x_i \,:\, C_i(x_i) \leq b_i^k \\ 0 \leq x_{ij} \leq \mathbf{1}(w_{ij} \leq b_i^k) \ \ \forall\, j \in N}} \Big\{ V_i(x_i)^{\gamma_i} \, (R_i(x_i) - z) \Big\}, \tag{20}$$

where we let $V_i(x_i) = \sum_{j \in N} v_{ij}\, x_{ij}$, $R_i(x_i) = \sum_{j \in N} v_{ij}\, r_{ij}\, x_{ij} / V_i(x_i)$ and $C_i(x_i) = \sum_{j \in N} w_{ij}\, x_{ij}$. The choice of the intervals $\{[b_i^{k-1}, b_i^k] : k = 1, \ldots, K\}$ can be completely arbitrary. A familiar derivation in nonlinear programming shows that $\phi_i^k(z)$ is a convex function of z. To see this result, assume that $\hat{x}_i$ is an optimal solution to problem (20) when this problem is solved with $z = \hat{z}$. So, we have $\phi_i^k(\hat{z}) = V_i(\hat{x}_i)^{\gamma_i}(R_i(\hat{x}_i) - \hat{z})$. If we solve problem (20) with an arbitrary value of z, $\hat{x}_i$ provides a feasible solution and we get $\phi_i^k(z) \geq V_i(\hat{x}_i)^{\gamma_i}(R_i(\hat{x}_i) - z)$. Subtracting these two expressions, we obtain $\phi_i^k(z) \geq \phi_i^k(\hat{z}) - V_i(\hat{x}_i)^{\gamma_i}\,(z - \hat{z})$. Therefore, $\phi_i^k(\cdot)$ satisfies the subgradient inequality with subgradient $-V_i(\hat{x}_i)^{\gamma_i}$ at point $\hat{z}$ and it must be convex. Also, this derivation shows that if we want to obtain a subgradient of $\phi_i^k(\cdot)$ at $\hat{z}$, then we can solve problem (20) with $z = \hat{z}$ and letting $\hat{x}_i$ be an optimal solution, a subgradient of $\phi_i^k(\cdot)$ at $\hat{z}$ is given by $-V_i(\hat{x}_i)^{\gamma_i}$. To obtain an upper bound on the optimal expected revenue, we propose solving the problem

$$\min \quad c\,\Delta + \sum_{i \in M} y_i \tag{21}$$

$$\text{st} \quad b_i^{k-1}\,\Delta + y_i \geq \phi_i^k(z) \qquad \forall\, i \in M,\; k = 1, \ldots, K \tag{22}$$

$$c\,\Delta + \sum_{i \in M} y_i = v_0\, z \tag{23}$$

$$\Delta \geq 0,\; y_i \text{ is free},\; z \text{ is free} \quad \forall\, i \in M, \tag{24}$$

where the decision variables are Δ, $y = \{y_i : i \in M\}$ and z. The problem above is a convex program since $\phi_i^k(\cdot)$ is convex. The next theorem shows that we can use this problem to obtain an upper bound on the optimal expected revenue z^* in problem (1).

Theorem 8 *Letting $(\hat{\Delta}, \hat{y}, \hat{z})$ be an optimal solution to problem* (21)-(24)*, we have $\hat{z} \geq z^*$.*

Proof. We let $(S_1^*, \ldots, S_m^*)$ be an optimal solution to problem (1), $b_i^* = C_i(S_i^*)$ and k_i' be such that $b_i^* \in [b_i^{k_i'-1}, b_i^{k_i'}]$. Consider a solution x_i^* to problem (20) obtained by letting $x_{ij}^* = 1$ if $j \in S_i^*$ and $x_{ij}^* = 0$ otherwise. Since $C_i(x_i^*) = C_i(S_i^*) = b_i^* \leq b_i^{k_i'}$, the solution x_i^* is feasible to problem (20) when we solve this problem with $k = k_i'$ and $z = \hat{z}$. So, we get $\phi_i^{k_i'}(\hat{z}) \geq V_i(x_i^*)^{\gamma_i}\,(R_i(x_i^*) - \hat{z}) = V_i(S_i^*)^{\gamma_i}\,(R_i(S_i^*) - \hat{z})$. Noting that $(\hat{\Delta}, \hat{y}, \hat{z})$ satisfies the first set of constraints in problem (21)-(24) with $k = k_i'$ and adding these constraints over all $i \in M$, we have

$$\sum_{i \in M} \hat{y}_i \geq \sum_{i \in M} \phi_i^{k_i'}(\hat{z}) - \sum_{i \in M} b_i^{k_i'-1}\,\hat{\Delta} \geq \sum_{i \in M} \phi_i^{k_i'}(\hat{z}) - \sum_{i \in M} C_i(S_i^*)\,\hat{\Delta} \geq \sum_{i \in M} \phi_i^{k_i'}(\hat{z}) - c\,\hat{\Delta},$$

where the second inequality is by the fact $C_i(S_i^*) = b_i^* \geq b_i^{k_i'-1}$ and third inequality is by the fact that $\sum_{i \in M} C_i(S_i^*) \leq c$. Since $(\hat{\Delta}, \hat{y}, \hat{z})$ satisfies the second constraint in problem (21)-(24) and noting the first and last terms in the chain of inequalities above, we get $v_0\,\hat{z} = c\,\hat{\Delta} + \sum_{i \in M} \hat{y}_i \geq \sum_{i \in M} \phi_i^{k_i'}(\hat{z}) \geq \sum_{i \in M} V_i(S_i^*)^{\gamma_i}\,(R_i(S_i^*) - \hat{z})$, where the last inequality is by the fact that $\phi_i^{k_i'}(\hat{z}) \geq V_i(S_i^*)^{\gamma_i}\,(R_i(S_i^*) - \hat{z})$, which is established above. So, we obtain $v_0\,\hat{z} \geq \sum_{i \in M} V_i(S_i^*)^{\gamma_i}\,(R_i(S_i^*) - \hat{z})$ and solving for $\hat{z}$ in this inequality yields $\hat{z} \geq \sum_{i \in M} V_i(S_i^*)^{\gamma_i}\,R_i(S_i^*) / (v_0 + \sum_{i \in M} V_i(S_i^*)^{\gamma_i}) = \Pi(S_1^*, \ldots, S_m^*) = z^*$. $\square$

Therefore, we can obtain an upper bound on the optimal expected revenue by using an optimal solution to problem (21)-(24). Since $\phi_i^k(\cdot)$ is convex, we can solve problem (21)-(24) by using a standard cutting plane method; see Ruszczynski (2006). All that a cutting plane method requires is subgradients of $\phi_i^k(\cdot)$. By the discussion right before Theorem 8, if we can solve problem (20) with $z = \hat{z}$ for an arbitrary value of $\hat{z}$, then letting $\hat{x}_i$ be an optimal solution to this problem, $-V_i(\hat{x}_i)^{\gamma_i}$ provides a subgradient of $\phi_i^k(\cdot)$ at $\hat{z}$. Thus, we can solve problem (21)-(24) by using a cutting plane method as long as we can solve problem (20) in a tractable fashion.

In the rest of this section, we focus on solving problem (20). Our development closely follows the results in Sections 3 and 5. In the next lemma, we begin by giving an alternative representation of problem (20). The proof of this lemma follows from an argument similar to the proof of Lemma 4 and we defer it to the appendix.

Lemma 9 *Let x_i^* be an optimal solution to problem* (20) *and* $u_i^* = \max\{z, \gamma_i\, z + (1-\gamma_i)\, R_i(x_i^*)\}$. *If $\hat{x}_i$ is an optimal solution to the problem*

$$\max_{\substack{x_i \,:\, C_i(x_i) \le b_i^k \\ 0 \le x_{ij} \le \mathbf{1}(w_{ij} \le b_i^k) \;\; \forall\, j \in N}} \Big\{ V_i(x_i)\,(R_i(x_i) - u_i^*) \Big\}, \tag{25}$$

then $\hat{x}_i$ is an optimal solution to problem (20) *as well.*

The lemma above shows that we can obtain an optimal solution to problem (20) by solving problem (25). The difficulty with solving problem (25), however, is that this problem requires knowing u_i^*, which in turn, requires knowing the optimal solution to problem (20). To resolve this difficulty, similar to our argument in Section 3, we propose solving the problem

$$\max_{\substack{x_i \,:\, C_i(x_i) \le b_i^k \\ 0 \le x_{ij} \le \mathbf{1}(w_{ij} \le b_i^k) \;\; \forall\, j \in N}} \Big\{ V_i(x_i)\,(R_i(x_i) - u_i) \Big\}, \tag{26}$$

for all $u_i \in \Re_+$. As a function of u_i, we let $x_i^*(u_i)$ be an optimal solution to the problem above. We observe that $\{x_i^*(u_i) : u_i \in \Re_+\}$ includes an optimal solution to problem (25) since problem (26) with $u_i = u_i^*$ is identical to problem (25). In this case, Lemma 9 implies that $\{x_i^*(u_i) : u_i \in \Re_+\}$ includes an optimal solution to problem (20) as well.

The discussion above indicates that if we can characterize the optimal solutions to problem (26) for all $u_i \in \Re_+$, then we can obtain an optimal solution to problem (20) by checking the objective value provided by each one of these solutions and picking the best one. To characterize the optimal solutions to problem (26) for all $u_i \in \Re_+$, we equivalently write this problem as

$$\max_{\substack{x_i \,:\, \sum_{j \in N} w_{ij}\, x_{ij} \le b_i^k \\ 0 \le x_{ij} \le \mathbf{1}(w_{ij} \le b_i^k) \;\; \forall\, j \in N}} \Big\{ \sum_{j \in N} v_{ij}\,(r_{ij} - u_i)\, x_{ij} \Big\}, \tag{27}$$

which is a knapsack problem. We can solve this problem as follows. We only focus on the items with space consumptions not exceeding b_i^k. We order these items according to their utility to space consumption ratios and fill the knapsack by using the items in the order of decreasing utility to space consumption ratios, as long as the sign of the utility to space consumption ratio is positive. Thus, the optimal solution to problem (27) depends only on the ordering and signs of the utility to space consumption ratios. In this case, by using an argument similar to the one in Section 5.2, it is possible to show that there are only $O(n^2)$ different optimal solutions to problem (27) as u_i takes values over $\Re_+$. To see this result, we define the linear functions $\{h_{ij}(\cdot) : j \in N\}$ as $h_{ij}(u_i) = v_{ij}\,(r_{ij} - u_i)/w_{ij}$ to capture the utility to space consumption ratio of item j. Computing the pairwise intersection points of these lines by solving $h_{ij}(u_i) = h_{ij'}(u_i)$ for u_i for all distinct $j, j' \in N$, these points correspond to the values of u_i at which the ordering between the utility to space consumption ratios of the items change. There are $O(n^2)$ such pairwise intersection points. Similarly solving $h_{ij}(u_i) = 0$ for u_i for all $j \in N$, we obtain the values of u_i at which the sign of the utility to space consumption ratios change. There are $O(n)$ such points. So, there are a total of $O(n^2) + O(n) = O(n^2)$ values for u_i at which the ordering or signs of the utility to space consumption ratios change. As long as u_i takes values between these points, the ordering or sign of the utility to space consumption ratios of the items in problem (27) do not change, which implies that the solution to this problem does not change either. Thus, there are only $O(n^2)$ different optimal solutions to problem (27) as u_i takes values over $\Re_+$, as desired.

Following the approach above, we can come up with $O(n^2)$ solutions such that the optimal solution to problem (26) for any $u_i \in \Re_+$ corresponds to one of these solutions. In this case, Lemma 9 implies that we can check the objective value provided by each one of these solutions for problem (20) and pick the best one to obtain an optimal solution to problem (20).

7 Numerical Experiments

By using the approach described in Section 5, we can obtain a 4-approximate solution to problem (1) under a space constraint. Our goal in this section is to numerically test the performance of this approach. Since we can obtain the optimal solution to problem (1) when we have a cardinality constraint, we do not provide numerical experiments under a cardinality constraint.

7.1 Numerical Setup

In our numerical experiments, we randomly generate a large number of problem instances. For each problem instance, we find a 4-approximate solution by using the approach described in Section 5. Furthermore, we compute an upper bound on the optimal expected revenue for each problem instance by using the approach described in Section 6. We check the gap between the expected revenue obtained by the 4-approximate solution and the upper bound on the optimal expected revenue to assess the optimality gap of the 4-approximate solution.

We use the following procedure to generate each problem instance. We set the number of nests as $m = 3$ or $m = 5$ and the number of products in each nest as $n = 15$ or $n = 30$. To come up with the revenues and preference weights of the products, we sample C_{ij} from the uniform distribution over $[0, 1]$. Similarly, we sample X_{ij} and Y_{ij} from the uniform distribution over $[0.75, 1.25]$. In this case, we set the revenue and preference weight of product j in nest i respectively as $r_{ij} = 10 \times C_{ij}^2 \times X_{ij}$ and $v_{ij} = 10 \times (1 - C_{ij}) \times Y_{ij}$. Through C_{ij}, we ensure that the products having larger revenues generally tend to have smaller preference weights, indicating more expensive products tend to be less attractive. Squaring C_{ij} in the expression for r_{ij} skews the distribution of the revenues so that we have a small number products with large revenues, but a large number of products with small revenues. Through X_{ij} and Y_{ij}, we incorporate idiosyncratic noise into the revenues and preference weights so that not all products with large revenues have small preference weights. We sample the dissimilarity parameter γ_i for each nest i from the uniform distribution over $[0.25, 0.75]$. To come up with the preference weight v_0 of the no purchase option, we choose v_0 such that the probability that a customer leaves without making a purchase is 0.4 even when we offer all products in all nests. Finally, we sample the space requirement w_{ij} of product j in nest i from the uniform distribution over $[1, 10]$. We set the capacity availability as $c = \kappa \sum_{i \in M} \sum_{j \in N} w_{ij}$ so that the capacity availability is a κ fraction of the total space consumption of all products in all nests. We use $\kappa = 0.1$, $\kappa = 0.15$ or $\kappa = 0.2$.

In our numerical experiments, we vary (m, n, κ) over $\{3, 5\} \times \{15, 30\} \times \{0.1, 0.15, 0.2\}$, which provides 12 parameter combinations. For each parameter combination, we generate 5,000 individual problem instances by using the approach described in the paragraph above.

7.2 Numerical Results

Our numerical results are given in Table 1. The first column in this table shows the parameter combination in consideration by using the triplet (m, n, κ). We recall that we generate 5,000 individual problem instances for each parameter combination. For each problem instance, we use the approach described in Section 5 to obtain a 4-approximate solution to problem (1). We use Rev^p to denote the expected revenue obtained by the 4-approximate solution for problem instance p. On the other hand, we use the approach described in Section 6 to obtain an upper bound on the optimal expected revenue. When computing the upper bound, the intervals $\{[b_i^{k-1}, b_i^k] : k = 0, \ldots, K\}$ are obtained by partitioning $[0, c]$ into 250 intervals of equal width. We use Bnd^p to denote the upper bound on the optimal total expected revenue for problem instance p. The second column in Table 1 gives the average percent gap between Rev^p and Bnd^p, where average is taken over all 5,000 problem instances in a parameter combination. The third column gives the 95th percentile of the percent gaps between Rev^p and Bnd^p over all 5,000 problem instances in a particular parameter combination. In other words, the second and third columns respectively give the average and 95th percentile of the data $\{100 \times (\mathsf{Bnd}^p - \mathsf{Rev}^p)/\mathsf{Bnd}^p : p = 1, \ldots, 5{,}000\}$. In this way, the second and third columns give an indication of the optimality gaps of the 4-approximate solutions. The fourth column gives the number of products in the 4-approximate solution, averaged over all 5,000 problem instances. The fifth column gives the number of problem instances for which the percent gap between Rev^p and Bnd^p is less than 1%. Similarly, the sixth, seventh, eighth, ninth and tenth columns give the number of problem instances where the percent gap between Rev^p and Bnd^p is

Param. Combin. (m, n, κ)	% Gap btw. Rev^p, Bnd^p		Avg. Assr. Size	No. Problem Instances with Certain % Gap btw. Rev^p, Bnd^p						Uncn. Assr. Size	No. Capac. Prbs.	Spc. to Cap. Ratio
	Avg.	95th		1%	2%	3%	4%	5%	10%			
$(3, 15, 0.20)$	1.52	3.84	9.94	2,069	3,552	4,398	4,796	4,939	5,000	17.29	4,919	0.11
$(3, 15, 0.15)$	2.20	5.49	8.10	1,406	2,728	3,659	4,265	4,638	4,998	17.29	4,986	0.15
$(3, 15, 0.10)$	3.33	8.41	6.16	896	1,899	2,740	3,393	3,882	4,892	17.29	5,000	0.22
$(3, 30, 0.20)$	0.81	1.96	20.14	3,492	4,776	4,978	4,995	4,998	5,000	33.71	4,951	0.06
$(3, 30, 0.15)$	1.19	2.73	16.42	2,462	4,223	4,853	4,981	4,999	5,000	33.71	4,994	0.07
$(3, 30, 0.10)$	1.79	4.09	12.48	1,569	3,204	4,207	4,728	4,912	5,000	33.71	5,000	0.11
$(5, 15, 0.20)$	1.09	2.41	16.88	2,680	4,455	4,929	4,992	4,998	5,000	28.66	4,984	0.07
$(5, 15, 0.15)$	1.56	3.48	13.79	1,722	3,591	4,538	4,895	4,985	5,000	28.66	4,999	0.09
$(5, 15, 0.10)$	2.29	5.23	10.56	1,081	2,546	3,595	4,338	4,690	4,999	28.66	5,000	0.13
$(5, 30, 0.20)$	0.68	1.35	33.92	4,115	4,981	5,000	5,000	5,000	5,000	56.25	4,993	0.03
$(5, 30, 0.15)$	0.95	1.86	27.65	3,068	4,838	4,991	5,000	5,000	5,000	56.25	5,000	0.04
$(5, 30, 0.10)$	1.37	2.76	21.07	1,862	4,082	4,853	4,989	4,999	5,000	56.25	5,000	0.07
Average	1.56	3.64										
Total				26,422	44,875	52,741	56,372	58,040	59,889		59,826	

Table 1: Performance of 4-approximate solutions.

respectively less than 2%, 3%, 4%, 5% and 10%. The eleventh and twelfth columns attempt to give a feel for the tightness of the space constraint. The eleventh column shows the average number of products in the optimal assortment when there are no capacity constraints. The twelfth column shows the number of problem instances for which this unconstrained solution violates the space constraint. Finally, the last column gives the average ratio between the space consumption of a product and the total capacity availability, averaged over all products and all problem instances in a parameter combination. The goal of this column is to give an indication of how many products we can offer if we use all of the available capacity.

Our results indicate that the 4-approximate solutions obtained by our approach perform quite well. Over all problem instances, the average optimality gap of these solutions is no larger than 1.56%. In 58,040 out of 60,000 problem instances that we solve in all parameter combinations, the optimality gaps of the 4-approximate solutions are less than 5%. In nine out of 12 parameter combinations, the 95th percentiles of the optimality gaps do not exceed 5%. As a general trend, we observe that the optimality gaps tend to get larger as κ gets smaller. As κ gets smaller, the capacity availability gets smaller and each product occupies a larger fraction of the available capacity. Thus, problem instances where each product occupies a larger fraction of the available capacity appear to be more difficult to approximate. This observation is well aligned with the intuition that the linear programming relaxation of a knapsack problem becomes looser as each item occupies a larger fraction of the knapsack capacity. In particular, it is known that if each item occupies no larger than a fraction ϵ of the knapsack capacity, then the optimal objective value of the linear programming relaxation exceeds the optimal objective value of a knapsack problem by at most a factor of $1/(1-\epsilon)$. Tightness of the linear programming relaxation deteriorates as ϵ becomes large and each item occupies a larger portion of the knapsack capacity. The most problematic parameter combination in Table 1 is $(3, 15, 0.1)$ corresponding to a small value of κ with $\kappa = 0.1$. Nevertheless, even for this parameter combination, in more than 75% of the problem instances, the optimality gap of the 4-approximate solution is no larger than 5%. It is also worthwhile to note that all optimality gaps we report are, in a sense, pessimistic estimates, since these optimality gaps are obtained by comparing the expected revenue from an assortment with an upper bound on the optimal expected revenue, rather than the optimal expected revenue itself.

The computation times to obtain the 4-approximate solutions and the upper bounds on the optimal expected revenues are reasonable. All of our numerical experiments are carried out on a Intel Xeon CPU with four cores running at 2.00 GHz, using Gurobi 5.1 as our linear programming software. For the largest problem instances with $m = 5$ and $n = 30$, we can obtain a 4-approximate solution in 3.517 seconds on average, with the shortest and longest computation time for an individual problem instance being respectively 1.591 and 5.928 seconds. For these largest problem instances, the average computation time to obtain an upper bound on the optimal expected revenue is 3.558 seconds, with the shortest and longest computation times respectively given by 1.420 and 5.663 seconds. Overall, our numerical results demonstrate that we can efficiently obtain 4-approximate solutions under a space constraint. Furthermore, we can solve a tractable convex program to obtain an upper bound on the optimal expected revenue and use this convex program to assess the optimality gap of the 4-approximate solution. If the optimality gap comes out to be small, then there is no need to look for better solutions. In this way, the convex program can be utilized to test the effectiveness of not only our 4-approximate solutions, but any other heuristic or approximation method for problem (1).

8 Conclusions

We studied assortment optimization problems when there is a constraint that limits the total capacity consumption of all products in the offered assortment. If each product occupies one unit of capacity, then there is a limit on the number of products in the offered assortment and the optimal assortment can be obtained by solving a linear program with $O(m^2n)$ decision variables and $O(m^2n^4)$ constraints. If each product occupies an arbitrary amount of space, then we can obtain a 4-approximate solution by solving a linear program with $O(m)$ decision variables and $O(mn^4)$ constraints.

There are a number of directions to extend our results. In this paper, we consider a capacity constraint of the form $\sum_{i \in M} C_i(S_i) \leq c$, limiting the total capacity consumption of the products offered in all nests. In addition to this capacity constraint, we can incorporate separate capacity constraints on the products that are offered in each nest, so that for each nest i the assortment S_i offered in this nest also satisfies $C_i(S_i) \leq \bar{c}_i$ for some $\bar{c}_i \in \Re_+$. It is simple to modify our approach to incorporate such nest specific capacity constraints. All we need to do is when generating the collections $\mathcal{A}_i$ of candidate assortments for each nest i, we focus our attention to assortments that satisfy this nest specific capacity constraint. Another extension to pursue is based on the understanding that our approach in this paper provides a 4-approximate solution under a space constraint and it does not provide any guidance as to how we can obtain better solutions if we are willing to increase the computational effort. It is worthwhile to develop approximation schemes that can tradeoff approximation guarantee with computational work.

References

Borsch-Supan, A. (1990), 'On the compatibility of nested logit models with utility maximization', *Journal of Econometrics* **43**, 373–388.

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

Davis, J., Gallego, G. and Topaloglu, H. (2013), 'Assortment optimization under variants of the nested logit model', *Operations Research* (to appear).

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

Farias, V. F., Jagabathula, S. and Shah, D. (2012), Sparse choice models, Technical report, MIT, Cambridge, MA.

Farias, V. F., Jagabathula, S. and Shah, D. (2013), 'A non-parametric approach to modeling choice with limited data', *Management Science* **59**(2), 305–322.

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

Gallego, G., Ratliff, R. and Shebalov, S. (2011), A general attraction model and an efficient formulation for the network revenue management problem, Technical report, Columbia University, New York, NY.

Gallego, G. and Topaloglu, H. (2014), 'Constrained assortment optimization for the nested logit model', *Management Science* (to appear).

Gallego, G. and Wang, R. (2011), Multi-product price optimization and competition under the nested attraction model, Technical report, Columbia University.

Kok, A. G., Fisher, M. L. and Vaidyanathan, R. (2008), Assortment planning: Review of literature and industry practice, *in* 'Retail Supply Chain Management', Kluwer Publishers.

Kok, A. G. and Xu, Y. (2011), 'Optimal and competitive assortments with endogenous pricing under hierarchical consumer choice models', *Management Science* **57**(9), 1546–1563.

Kunnumkal, S. (2013), 'Randomization approaches for network revenue management with customer choice behavior', *Production and Operations Management* (to appear).

Kunnumkal, S. and Topaloglu, H. (2008), 'A refined deterministic linear program for the network revenue management problem with customer choice behavior', *Naval Research Logistics Quarterly* **55**(6), 563–580.

Li, G. and Rusmevichientong, P. (2012), Technical note: A simple greedy algorithm for assortment optimization in the two-level nested logit model, Technical report, University of Southern California, Marshall School of Business.

Li, H. and Huh, W. T. (2011), 'Pricing multiple products with the multinomial logit and nested models: Concavity and implications', *Manufacturing & Service Operations Management* **13**(4), 549–563.

Liu, Q. and van Ryzin, G. (2008), 'On the choice-based linear programming model for network revenue management', *Manufacturing & Service Operations Management* **10**(2), 288–310.

Luce, R. D. (1959), *Individual Choice Behavior: A Theoretical Analysis*, Wiley, New York, NY.

McFadden, D. (1974), Conditional logit analysis of qualitative choice behavior, *in* P. Zarembka, ed., '*Frontiers in Economics*', Academic Press, pp. 105–142.

Meissner, J., Strauss, A. and Talluri, K. (2012), 'An enhanced concave program relaxation for choice network revenue management', *Production and Operations Management* **22**(1), 71–87.

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

Natarajan, K., Song, M. and Teo, C.-P. (2009), 'Persistency model and its applications in choice modeling', *Management Science* **55**(3), 453–469.

Puterman, M. L. (1994), *Markov Decision Processes*, John Wiley and Sons, Inc., New York.

Rusmevichientong, P., Shen, Z.-J. M. and Shmoys, D. B. (2009), 'A PTAS for capacitated sum-of-ratios optimization', *Operations Research Letters* **37**(4), 230–238.

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

Rusmevichientong, P., Shmoys, D. B., Tong, C. and Topaloglu, H. (2013), 'Assortment optimization under the multinomial logit model with random choice parameters', *Production and Operations Management* (to appear).

Ruszczynski, A. (2006), *Nonlinear Optimization*, Princeton University Press, Princeton, New Jersey.

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

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

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

Train, K. (2003), *Discrete Choice Methods with Simulation*, Cambridge University Press, Cambridge, UK.

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

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

Williams, H. C. W. L. (1977), 'On the formation of travel demand models and economic evaluation measures of user benefit', *Environment and Planning* **9**(3), 285–344.

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

A Appendix: Omitted Proofs

A.1 Lemma 10

Proof of Lemma 4 uses the next lemma. A similar result is given in Lemma 3 in Davis et al. (2013). For completeness, we provide an alternative proof, which is also simpler.

Lemma 10 *Let $(S_1^*, \ldots, S_m^*)$ be an optimal solution to problem (1) with an objective value of z^*. For any $i \in M$, if $R_i(S_i^*) < z^*$, then we have $S_i^* = \emptyset$.*

Proof. To get a contradiction, we assume that $R_i(S_i^*) < z^*$ and $S_i^* \neq \emptyset$ for some nest i. We define the assortment $(\tilde{S}_1, \ldots, \tilde{S}_m)$ as $\tilde{S}_i = \emptyset$ and $\tilde{S}_l = S_l^*$ for all $l \in M \setminus \{i\}$. Noting that $z^* = \Pi(S_1^*, \ldots, S_m^*) = \sum_{l \in M} V_l(S_l^*)^{\gamma_l} R_l(\hat{S}_l^*)/(v_0 + \sum_{l \in M} V_l(S_l^*)^{\gamma_l})$, arranging the terms in this equality yields $v_0\, z^* = \sum_{l \in M} V_l(S_l^*)^{\gamma_l}\, (R_l(S_l^*) - z^*)$. So, since we have $\tilde{S}_i = \emptyset$ satisfying $V_i(\tilde{S}_i) = 0$, but $S_i^* \neq \emptyset$ satisfying $V_i(S_i^*) > 0$ and $R_i(S_i^*) - z^* < 0$, we obtain $v_0\, z^* = \sum_{l \in M} V_l(S_l^*)^{\gamma_l}\, (R_l(S_l^*) - z^*) < \sum_{l \in M} V_l(\tilde{S}_l)^{\gamma_l}\, (R_l(\tilde{S}_l) - z^*)$. Focusing on the first and last terms in the last chain of inequalities and solving for z^*, we have $z^* < \sum_{l \in M} V_l(\tilde{S}_l)^{\gamma_l} R_l(\tilde{S}_l)/(v_0 + \sum_{l \in M} V_l(\tilde{S}_l)^{\gamma_l}) = \Pi(\tilde{S}_1, \ldots, \tilde{S}_m)$, indicating that the assortment $(\tilde{S}_1, \ldots, \tilde{S}_m)$ provides larger expected revenue than the optimal assortment. Furthermore, since $\sum_{l \in M} C_l(\tilde{S}_l) \leq \sum_{l \in M} C_l(S_l^*) \leq c$, the assortment $(\tilde{S}_1, \ldots, \tilde{S}_m)$ is feasible to problem (1). Thus, we reach a contradiction. □

A.2 Proof of Lemma 7

Associating the dual variables Δ and $y = \{y_i : i \in M\}$ with the two sets of constraints in problem (11)-(14), the dual of this problem is

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

Therefore, the solution $(\hat{\Delta}, \hat{y})$ is feasible to the dual of problem (11)-(14) when we solve this problem with $z = \hat{z}$, which implies that $f^R(\hat{z}) \leq c\,\hat{\Delta} + \sum_{i \in M} \hat{y}_i = v_0\,\hat{z}$, where the last equality follows from the fact that $(\hat{\Delta}, \hat{y}, \hat{z})$ is a feasible solution to problem (15)-(18). To get a contradiction, assume that the last inequality is strict so that $f^R(\hat{z}) < v_0\,\hat{z}$. We let $\tilde{z}$ be such that $f^R(\tilde{z}) = v_0\,\tilde{z}$ and $(\tilde{\Delta}, \tilde{y})$ be an optimal solution to the problem above when we solve this problem with $z = \tilde{z}$. Thus, we get $v_0\,\tilde{z} = f^R(\tilde{z}) = c\,\tilde{\Delta} + \sum_{i \in M} \tilde{y}_i$, which indicates that $(\tilde{\Delta}, \tilde{y}, \tilde{z})$ is a feasible solution to problem (15)-(18). In this case, it follows that $v_0\,\tilde{z} = f^R(\tilde{z}) = c\,\tilde{\Delta} + \sum_{i \in M} \tilde{y}_i \geq c\,\hat{\Delta} + \sum_{i \in M} \hat{y}_i = v_0\,\hat{z} > f^R(\hat{z})$, where the first inequality is by the fact that $(\tilde{\Delta}, \tilde{y}, \tilde{z})$ is a feasible, but not necessarily an optimal, solution to problem (15)-(18). The last chain of inequalities yields $\tilde{z} \geq \hat{z}$ and $f^R(\tilde{z}) > f^R(\hat{z})$, which contradict the fact that $f^R(\cdot)$ is decreasing.

A.3 Proof of Lemma 9

The proof is similar to that of Lemma 4. For notational brevity, we let $V_i^* = V_i(x_i^*)$, $R_i^* = R_i(x_i^*)$, $\hat{V}_i = V_i(\hat{x}_i)$ and $\hat{R}_i = R_i(\hat{x}_i)$. First, assume that $x_i^* \neq 0$. In this case, it follows that $R_i^* \geq z$, otherwise the optimal solution x_i^* would provide a negative objective value for problem (20), whereas the solution zero already provides an objective value of zero for this problem. Since $R_i^* \geq z$, we obtain $u_i^* = \gamma_i\, z + (1-\gamma_i)\, R_i^*$. By the definition of $\hat{x}_i$, we have $\hat{V}_i\,(\hat{R}_i - u_i^*) \geq V_i^*\,(R_i^* - u_i^*)$. Noting that $u_i^* = \gamma_i\, z + (1-\gamma_i)\, R_i^*$, we write this inequality as $\hat{V}_i\,(\hat{R}_i - \gamma_i\, z - (1-\gamma_i)\, R_i^*) \geq \gamma_i\, V_i^*\,(R_i^* - z)$. If $\hat{V}_i = 0$, then since R_i^* is larger than z, the last inequality yields $V_i^*\,(R_i^* - z) = 0$, so we obtain $\hat{V}_i^{\gamma_i}(\hat{R}_i - z) = 0 = (V_i^*)^{\gamma_i}(R_i^* - z)$. If $\hat{V}_i \neq 0$, then we write the last inequality as

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

Using the fact that $g(y) = y^{\gamma_i}$ is a concave function of y with $g(1) = 1$ and $g'(1) = \gamma_i$, we get $y^{\gamma_i} \leq 1 + \gamma_i\,(y-1) = \gamma_i\, y + (1-\gamma_i)$. Using this last inequality with $y = V_i^*/\hat{V}_i$ in the inequality above, we get $\hat{R}_i - z \geq (V_i^*/\hat{V}_i)^{\gamma_i}(R_i^* - z^*)$ and arranging the terms yields $\hat{V}_i^{\gamma_i}(\hat{R}_i - z) \geq (V_i^*)^{\gamma_i}(R_i^* - z)$. So, we obtain $\hat{V}_i^{\gamma_i}(\hat{R}_i - z) \geq (V_i^*)^{\gamma_i}(R_i^* - z)$ under the case $x_i^* \neq 0$.

Second, assume that $x_i^* = 0$. If $\hat{x}_i = 0$ as well, then we get $\hat{V}_i^{\gamma_i}(\hat{R}_i - z) = 0 = (V_i^*)^{\gamma_i}(R_i^* - z)$. In the rest of the proof, assume that $\hat{x}_i \neq 0$. Since $x_i^* = 0$, we have $R_i^* = 0$, in which case, $u_i^* = z$ by the definition of u_i^*. Since $\hat{x}_i$ is an optimal solution to problem (25), we must have $\hat{R}_i \geq u_i^* = z$, otherwise the optimal solution $\hat{x}_i$ would provide a negative objective value for problem (25), whereas the solution zero immediately gives an objective value of zero for this problem. Therefore, we obtain $\hat{V}_i^{\gamma_i}(\hat{R}_i - z) \geq 0 = (V_i^*)^{\gamma_i}(R_i^* - z)$. So, under all cases we consider, it follows that $\hat{V}_i^{\gamma_i}(\hat{R}_i - z) \geq (V_i^*)^{\gamma_i}(R_i^* - z)$, indicating that the objective value provided by the solution $\hat{x}_i$ for problem (20) is at least as large as the objective value provided by the optimal solution x_i^* to this problem. Thus, $\hat{x}_i$ is an optimal solution to problem (20).