Revenue Management under the Markov Chain Choice Model

Jacob B. Feldman,[1] Huseyin Topaloglu[2]

March 29, 2017

Abstract

We consider revenue management problems when customers choose among the offered products according to the Markov chain choice model. In this choice model, a customer arrives into the system to purchase a particular product. If this product is available for purchase, then the customer purchases it. Otherwise, the customer transitions to another product or to the no purchase option, until she reaches an available product or the no purchase option. We consider three classes of problems. First, we study assortment problems, where the goal is to find a set of products to offer to maximize the expected revenue obtained from each customer. We give a linear program to obtain the optimal solution. Second, we study single resource revenue management problems, where the goal is to adjust the set of offered products over a selling horizon when the sale of each product consumes the resource. We show how the optimal set of products to offer changes with the remaining resource inventory. Third, we study network revenue management problems, where the goal is to adjust the set of offered products over a selling horizon when the sale of each product consumes a combination of resources. A standard linear programming approximation of this problem includes one decision variable for each subset of products. We show that this linear program can be reduced to an equivalent one with a substantially smaller size. We give an algorithm to recover the optimal solution to the original linear program from the reduced linear program. The reduced linear program can dramatically improve the solution times for the original linear program.

Incorporating customer choice behavior into revenue management models has been seeing increased attention. Traditional revenue management models assume that each customer arrives into the system with the intention of purchasing a certain product. If this product is available for purchase, then the customer purchases it. Otherwise, the customer leaves without a purchase. In reality, however, there may be multiple products that serve the needs of a customer and a customer may observe the set of available products and make a choice among them. When customers choose among the available products, the demand for a particular product depends on what other products are made available to the customers, creating interactions between the demands for the different products. When such interactions exist between the demands for the different products, finding the right set of products to offer to customers can be a challenging task.

In this paper, we consider revenue management problems when customers choose among the offered products according to the Markov chain choice model. In the Markov chain choice model, a customer arrives into the system to purchase a particular product. If this product is available for purchase, then the customer purchases it. Otherwise, the customer transitions to another product with a certain probability and checks the availability of the other product, or she transitions to the no purchase option with a certain probability and leaves the system without a purchase. In

[1]Olin Business School, Washington University in St. Louis, St. Louis, MO 63130, jbfeldman@wustl.edu.

[2]School of Operations Research and Information Engineering, Cornell Tech, New York, NY 10011, topaloglu@orie.cornell.edu.

this way, the customer transitions between the products until she reaches a product available for purchase or she reaches the no purchase option. We consider three fundamental classes of revenue management problems when customers choose under the Markov chain choice model. In particular, we consider assortment optimization problems, revenue management problems with a single resource and revenue management problems over a network of resources. We proceed to describing our contributions to these three classes of problems.

CONTRIBUTIONS. First, we consider assortment problems. In the assortment setting, there is a revenue for each product. Customers choose among the offered products according to the Markov chain choice model. We want to offer a set of products to maximize the expected revenue from each customer. We relate the probability of purchasing each product under the Markov chain choice model to the extreme points of a polyhedron (Lemma 1). Using this result, we show that the optimal set of products to offer can be obtained by a linear program (Theorem 2). Also, we show that as the revenues of the products increase by the same amount, the optimal subset to offer becomes larger (Lemma 3). This property becomes useful when we study the optimal policy for the single resource revenue management problem. We show that the optimal subset of products to offer is not nested by revenue in general, so that we may offer a product with a smaller revenue, but not offer a product with a larger revenue. We give one sufficient condition on the Markov chain choice model under which it is optimal to offer a nested by revenue subset (Lemma 4).

Second, we consider revenue management problems with a single resource. In this setting, we need to decide which set of products to make available over a selling horizon. At each time period, an arriving customer chooses among the set of available products according to the Markov chain choice model. There is limited inventory of the resource and the sale of a product consumes one unit of the resource. The goal is to decide which set of products to make available at each time period to maximize the total expected revenue. We show that as we have more capacity or as we get closer to the end of the selling horizon, the urgency to liquidate the resource inventory takes over and the optimal policy offers a larger subset of products (Theorem 5). This result implies that as we have less remaining capacity at a time period, we stop offering certain products. In general, we may stop offering a product with a larger revenue before we stop offering a product with a smaller revenue, but we give one sufficient condition under which the order in which we stop offering the products follows the revenue order of the products (Lemma 6).

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

Focusing on the deterministic linear program described above, we show that if the customers choose according to the Markov chain choice model, then the deterministic linear program can immediately be reduced to an equivalent linear program whose numbers of decision variables and constraints increase only linearly with the numbers of products and resources (Theorem 7). We develop an algorithm to recover the optimal solution to the original deterministic linear program by using the optimal solution to the reduced linear program. By using the reduced linear program, we demonstrate that the subsets of products offered by the optimal solution to the deterministic linear program can be ordered such that one subset is included in another one (Theorem 9). This result implies that the optimal solution to the deterministic linear program offers at most $n+1$ different subsets, where n is the number of products. These properties are consequences of the Markov chain choice model and they do not necessarily hold under other choice models. Our computational experiments show that using the reduced linear program can provide remarkable computational savings over solving the original deterministic linear program through column generation. For problem instances with 100 resources and 2000 products, we cannot solve the original deterministic linear program within two hours of run time, while we can use the reduced linear program to obtain the optimal subset offer frequencies within four minutes.

In addition to our contributions to the three classes of problems above, we provide some practical support for the Markov chain choice model by giving computational experiments that demonstrate that the Markov chain choice model may capture the customer choice behavior better when compared with the popular multinomial logit model, while ensuring that the corresponding optimization problems still remain tractable.

Related Literature. Markov chain choice model appears in Zhang and Cooper (2005), where the authors use this choice model to capture the customer choice process in one of their numerical examples. Markov chain choice model was recently analyzed in the paper by Blanchet et al. (2013). This paper shows that the multinomial logit model, which is often used to model customer choices in practice, is a special case of the Markov chain choice model and shows how to solve the assortment problem under the Markov chain choice through a dynamic program. We give an alternative solution approach for the assortment problem that is based on a linear program. Blanchet et al. (2013) do not focus on revenue management problems with limited resources. We show structural properties of the optimal policy for the single resource revenue management problem and show how to reduce the size of the deterministic linear programming formulation for the network revenue management problem. On the other hand, Gallego et al. (2011) show that if the customers choose according to the multinomial logit model, then the deterministic linear program for the network revenue management problem can be reduced to an equivalent linear program whose size grows linearly with the numbers of products and resources. We find that an analogue of this result can be established under the Markov chain choice model, which is, as mentioned above, more general than the multinomial logit model. Obtaining the optimal solution to the original deterministic linear program through the optimal solution to the reduced linear program is substantially more difficult under the Markov chain choice model and our results demonstrate how to accomplish this task. Lastly, Talluri and van Ryzin (2004) study the single resource revenue management problem and derive structural properties of the optimal policy. Our study of the single resource revenue management problem is based on their model.

There is literature on assortment problems without inventory considerations. Rusmevichientong et al. (2010), Wang (2012) and Wang (2013) study assortment problems when customers choose according to variants of the multinomial logit model. Bront et al. (2009), Mendez-Diaz et al. (2010), Desir and Goyal (2013) and Rusmevichientong et al. (2014) focus on assortment problems under a mixture of multinomial logit models. Li and Rusmevichientong (2014), Davis et al. (2014), Gallego and Topaloglu (2014) and Li et al. (2015) study assortments problem under the nested logit model. Jagabathula (2008) and Farias et al. (2013) use a nonparametric choice model to capture customer choices, where each customer arrives with a particular ordering of products in mind and purchases the first available product in her ordering. Li and Huh (2011) and Gallego and Wang (2014) study related pricing problems under the nested logit model.

It is common to formulate deterministic linear programming approximations for network revenue management problems under the assumption that customer choices take on their expected values. Such approximations appear in Gallego et al. (2004), Liu and van Ryzin (2008), Bront et al. (2009), Kunnumkal and Topaloglu (2008), Talluri (2011), Meissner et al. (2012) and Vossen and Zhang (2013). Zhang and Cooper (2005), Zhang and Adelman (2009), Kunnumkal and Talluri (2012) and Meissner and Strauss (2012) provide tractable methods to approximate the dynamic programming formulations of network revenue management problems.

Organization. In Section 1, we formulate the Markov chain choice model. In Section 2, we show how to solve the assortment problem. In Section 3, we give structural properties for the solution to the assortment problem. In Section 4, we consider the single resource revenue management problem and show that we offer a larger subset of products as we have more capacity at a time period or as we get closer to the end of the selling horizon. In Section 5, we show that the deterministic linear program for the network revenue management problem can be reduced to an equivalent linear program whose size increases linearly with the numbers of products and resources. In Section 6, we give an algorithm to recover the optimal solution to the original deterministic linear program by using the optimal solution to the reduced linear program. In Section 7, we give computational experiments to demonstrate the benefits from using the Markov chain choice model instead of simpler choice models. In Section 8, we give computational experiments to demonstrate the benefits from the reduced linear program. In Section 9, we conclude.

1 Markov Chain Choice Model

In the Markov chain choice model, there are n products indexed by $N = \{1, \ldots, n\}$. With probability λ_j, a customer arrives into the system to purchase product j. If this product is available for purchase, then the customer purchases it. Otherwise, the customer transitions to product i with probability $\rho_{j,i}$ and checks whether product i is available for purchase. With probability $1 - \sum_{i \in N} \rho_{j,i}$, the customer transitions to the no purchase option and leaves the system without a purchase. In this way, the customer transitions between the products according to a Markov chain until she visits a product available for purchase or she visits the no purchase option. Given that we offer the subset $S \subset N$ of products, we use $P_{j,S}$ to denote the expected number of times that a customer visits product j that is available for purchase. By definition, we have $P_{j,S} = 0$ for all $j \notin S$. Similarly, given that we offer the subset $S \subset N$ of products, we use $R_{j,S}$ to denote the expected number of times that a customer visits product j that is not available for purchase. By definition, we have $R_{j,S} = 0$ for all $j \in S$. Using the vectors $P_S = (P_{1,S}, \ldots, P_{n,S})$ and $R_S = (R_{1,S}, \ldots, R_{n,S})$, we can obtain (P_S, R_S) be solving the system of equations

$$\begin{aligned} & P_{j,S} \,+\, R_{j,S} \;=\; \lambda_j \,+\, \sum_{i \in N} \rho_{i,j}\, R_{i,S} \quad \forall\, j \in N, \\ & P_{j,S} = 0 \quad \forall\, j \notin S, \qquad\qquad R_{j,S} = 0 \quad \forall\, j \in S. \end{aligned} \tag{Balance}$$

We interpret the Balance equations as follows. On the left side of the Balance equations, $P_{j,S} + R_{j,S}$ corresponds to the expected number of times that a customer visits product j during the course of her choice process. The expected number of times that a customer visits product j as the first product she visits is λ_j. Furthermore, the expected number of times that a customer visits some product i that is not available for purchase is $R_{i,S}$. Each time a customer visits such a product i, she transitions from product i to product j with probability $\rho_{i,j}$. Thus, on the right side of the Balance equations, $\lambda_j + \sum_{i \in N} \rho_{i,j}\, R_{i,S}$ also corresponds to the expected number of times that a customer visits product j during the course of her choice process. The choice process of a customer stops when she visits a product that is available for purchase. Thus, a customer visits a product that is available for purchase at most once, which implies that the expected number of times that a customer visits a product that is available for purchase is the same as the probability that a customer purchases this product. So, once we solve the Balance equations, $P_{j,S}$ gives the probability that a customer purchases product j when we offer the subset S of products.

We can interpret λ_j as the probability that product j is the first choice of a customer and $\rho_{j,i}$ as the probability that the next choice of a customer is product i given that her current choice is product j. Throughout the paper, we assume that $\lambda_j > 0$ and $\sum_{i \in N} \rho_{j,i} < 1$ for all $j \in N$. So, there is a strictly positive probability that a customer arrives into the system to purchase any of the products and a customer can leave the system after she visits any of the products. Since $R_{j,S} = 0$ for all $j \in S$, by the Balance equations, we have $P_{j,S} = \lambda_j + \sum_{j \in N} \rho_{j,i}\, R_{j,S}$ for all $j \in S$. Thus, the assumption that $\lambda_j > 0$ for all $j \in N$ ensures that $P_{j,S} > 0$ for all $j \in S$, indicating there is a strictly positive probability that each product in the offered subset is purchased. Our results continue to hold with some modifications when we have $\lambda_j = 0$ for some $j \in N$, but the proofs get significantly longer without providing additional insight. We point out these modifications in the conclusions section. Also, since $P_{j,S} = 0$ for all $j \notin S$ and $R_{j,S} = 0$ for all $j \in S$, by the BALANCE equations, we have $R_{j,S} = \lambda_j + \sum_{i \notin S} \rho_{i,j} R_{i,S}$ for all $j \notin S$. Using the matrix $\bar{Q} = \{\rho_{j,i} : j, i \notin S\}$ and the vectors $\bar{\lambda} = \{\lambda_j : j \notin S\}$ and $\bar{R}_S = \{R_{j,S} : j \notin S\}$, we write the last equality as $(I - \bar{Q})^\top \bar{R}_S = \bar{\lambda}$, where I is the identity matrix with the appropriate dimension. Since $\sum_{i \in N} \rho_{j,i} < 1$ for all $j \in N$, by Theorem 3.2.c in Puterman (1994), $(I - \bar{Q})^{-1}$ exists and has nonnegative entries. So, noting that $((I - \bar{Q})^\top)^{-1} = ((I - \bar{Q})^{-1})^\top$, we have $\bar{R}_S = ((I - \bar{Q})^{-1})^\top \bar{\lambda}$. Once we compute $\bar{R}_S = \{R_{j,S} : j \notin S\}$ by using the last equality, we can compute $P_{j,S}$ for all $j \in S$ by noting that $P_{j,S} = \lambda_j + \sum_{i \notin S} \rho_{i,j} R_{i,S}$ for all $j \in S$. So, the assumption that $\sum_{i \in N} \rho_{j,i} < 1$ for all $j \in N$ ensures that we have a unique and nonnegative solution to the BALANCE equations for any subset $S \subset N$.

The Markov chain choice model may not always reflect the thought process of a customer when making a purchase, but the important point is that if the purchase probabilities $(P_{1,S}, \ldots, P_{n,S})$ computed through the Markov chain choice model can accurately reflect the probabilities that a customer purchases different products when we offer the subset S of products, then it is not necessary to insist that the Markov chain choice model reflects the thought process of a customer. It is not difficult to see why the Markov chain choice model may not always reflect the thought process of a customer. For example, the Markov chain choice model implies that the substitution behavior of a customer is memoryless. Given that the current choice of a customer is product j, the probability that her next choice is product i is given by $\rho_{j,i}$, which does not depend on her previous choices. Also, making transitions over the full set of products N implies that a customer has the knowledge of all of the products that can possibly be offered for purchase. In reality, a customer usually has the knowledge of only the products that are actually offered for purchase, possibly with the addition of a few other products that can possibly be offered. When a customer has the knowledge of only the products that are actually offered for purchase, we may have to model the transition probability $\rho_{j,i}$ as a quantity that depends on the subset of the offered products, but this extension dramatically complicates the analysis. Finally, the Markov chain choice model implies that the thought process of a customer may visit a product that is not available for purchase multiple times.

One approach for constructing choice models is based on random utility maximization, where each customer associates utilities with the products and the no purchase option. These utilities are known to the customer and follow a particular distribution. The customer chooses the available option that provides the largest utility. Section 9.4 in Zhang and Cooper (2005) shows that the Markov chain choice model is compatible with the random utility maximization approach.

2 Assortment Optimization

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

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

Even computing the objective value of the problem above for a certain subset S is not trivial, since computing $P_{j,S}$ requires solving the Balance equations, but we show that we can solve the Assortment problem through a linear program. To show this result, we exploit a connection between (P_S, R_S) and the extreme points of a certain polyhedron. Consider the polyhedron

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

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

Lemma 1 *For an extreme point $(\hat{x}, \hat{z})$ of $\mathcal{H}$, define $S_{\hat{x}} = \{ j \in N : \hat{x}_j > 0 \}$. Then, we have $P_{j,S_{\hat{x}}} = \hat{x}_j$ and $R_{j,S_{\hat{x}}} = \hat{z}_j$ for all $j \in N$.*

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

An important implication of Lemma 1 is that we can obtain the optimal objective value of the Assortment problem by solving the linear program

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

To see this result, we use $\hat{S}$ to denote the optimal solution to the Assortment problem. Since $(P_{\hat{S}}, R_{\hat{S}})$ satisfies the Balance equations with $S = \hat{S}$, it follows that $(P_{\hat{S}}, R_{\hat{S}})$ is a feasible solution to the linear program above providing the objective value $\sum_{j\in N} r_j\, P_{j,\hat{S}}$. Therefore, there exists a feasible solution to the linear program above, which provides an objective value that is equal to the optimal objective value of the ASSORTMENT problem, indicating that the optimal objective value of the linear program above is at least as large as the optimal objective value of the ASSORTMENT problem. On the other hand, letting $(\hat{x}, \hat{z})$ be the optimal solution to the linear program above, define the subset $S_{\hat{x}} = \{j \in N : \hat{x}_j > 0\}$. Without loss of generality, we assume that $(\hat{x}, \hat{z})$ is an extreme point solution, in which case, Lemma 1 implies that $P_{j,S_{\hat{x}}} = \hat{x}_j$ for all $j \in N$. Therefore, the subset $S_{\hat{x}}$ provides an objective value of $\sum_{j\in N} r_j\, P_{j,S_{\hat{x}}} = \sum_{j\in N} r_j\, \hat{x}_j$ for the ASSORTMENT problem, indicating that there exists a feasible solution to the ASSORTMENT problem, which provides an objective value that is equal to the optimal objective value of the linear program above. Thus, the optimal objective value of the ASSORTMENT problem is at least as large as the optimal objective value of the linear program above, establishing the desired result. Naturally, we can obtain the optimal objective value of the linear program above by using its dual, which is given by

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

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

Theorem 2 *Letting $\hat{v}$ be the optimal solution to the* DUAL *problem, define $\hat{S} = \big\{j \in N : \hat{v}_j = r_j\big\}$. Then, $\hat{S}$ is the optimal solution to the* ASSORTMENT *problem.*

Proof. We use $\hat{Z}$ to denote the optimal objective value of the DUAL problem. By the discussion right before the theorem, $\hat{Z}$ also corresponds to the optimal objective value of the ASSORTMENT problem. We note that for each $j \in N$, we have $\hat{v}_j = r_j$ or $\hat{v}_j = \sum_{i\in N} \rho_{j,i}\, \hat{v}_i$. In particular, if we have $\hat{v}_j > r_j$ and $\hat{v}_j > \sum_{i\in N} \rho_{j,i}\, \hat{v}_i$ for some $j \in N$, then we can decrease the value of the decision variable $\hat{v}_j$ by a small amount, while keeping the feasibility of the solution $\hat{v}$ for the DUAL problem. Since $\lambda_j > 0$, the solution obtained in this fashion provides a strictly smaller objective value than the optimal solution, which is a contradiction. Since we have $\hat{v}_j = r_j$ or $\hat{v}_j = \sum_{i\in N} \rho_{j,i}\, \hat{v}_i$ for all $j \in N$, by the definition of $\hat{S}$, it holds that $\hat{v}_j = r_j$ for all $j \in \hat{S}$ and $\hat{v}_j = \sum_{i\in N} \rho_{j,i}\, \hat{v}_i$ for all $j \notin \hat{S}$. By the BALANCE equations, we also have $P_{j,\hat{S}} = 0$ for all $j \notin \hat{S}$ and $R_{j,\hat{S}} = 0$ for all $j \in \hat{S}$. Therefore, we obtain $P_{j,\hat{S}}\, \hat{v}_j = P_{j,\hat{S}}\, r_j$ for all $j \in N$ and $R_{j,\hat{S}}\, \hat{v}_j = \sum_{i\in N} R_{j,\hat{S}}\, \rho_{j,i}\, \hat{v}_i$ for all $j \in N$. Adding the last two equalities over all $j \in N$, it follows that $\sum_{j\in N} P_{j,\hat{S}}\, \hat{v}_j + \sum_{j\in N} R_{j,\hat{S}}\, \hat{v}_j = \sum_{j\in N} P_{j,\hat{S}}\, r_j + \sum_{j\in N} \sum_{i\in N} R_{j,\hat{S}}\, \rho_{j,i}\, \hat{v}_i$. If we arrange the terms in this equality, then we obtain

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

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

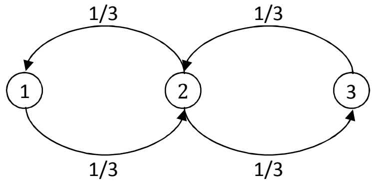
Thus, by Theorem 2, we can obtain the optimal solution to the Assortment problem by solving the Dual problem, which indicates that the Assortment problem is tractable.

3 Properties of the Optimal Assortment

In this section, we show several structural properties of the optimal solution to the Assortment problem. In the next lemma, we show that the optimal solution to the Assortment problem becomes a smaller subset when we decrease the revenues associated with all of the products by the same positive amount. In particular, for $\eta \geq 0$, we let $\hat{v}^\eta$ be the optimal solution to the Dual problem when we decrease the revenues of all of the products by η. By Theorem 2, $\{j \in N : \hat{v}_j^\eta = r_j - \eta\}$ is the optimal solution to the Assortment problem when we decrease the revenues associated with all of the products by η. In the next lemma, we show that $\{j \in N : \hat{v}_j^\eta = r_j - \eta\} \subset \{j \in N : \hat{v}_j^0 = r_j\}$, which implies that if we decrease the revenues of all of the products by the same positive amount η, then the optimal solution to the Assortment problem becomes a smaller subset. This result becomes useful when we show the structural properties of the optimal policy for the single resource revenue management problem.

Lemma 3 *For $\eta \geq 0$, let $\hat{v}^\eta$ be the optimal solution to the* Dual *problem when we decrease the revenues of all products by η. Then, we have $\{j \in N : \hat{v}_j^\eta = r_j - \eta\} \subset \{j \in N : \hat{v}_j^0 = r_j\}$.*

Proof. By the same argument in the proof of Theorem 2, we have $\hat{v}_j^0 = r_j$ or $\hat{v}_j^0 = \sum_{i\in N} \rho_{j,i}\, \hat{v}_i^0$ for all $j \in N$, in which case, noting the constraints in the Dual problem, we obtain $\hat{v}_j^0 = \max\{r_j, \sum_{i\in N} \rho_{j,i}\, \hat{v}_i^0\}$ for all $j \in N$. A similar argument yields $\hat{v}_j^\eta = \max\{r_j - \eta, \sum_{i\in N} \rho_{j,i}\, \hat{v}_i^\eta\}$ for all $j \in N$. We define $\tilde{v} = (\tilde{v}_1, \ldots, \tilde{v}_n)$ as $\tilde{v}_j = \min\{\hat{v}_j^0, \hat{v}_j^\eta + \eta\}$ for all $j \in N$. We claim that $\tilde{v}$ is a feasible solution to the Dual problem. To see the claim, by the definition of $\tilde{v}$, we have $\tilde{v}_j \leq \hat{v}_j^0$ and $\tilde{v}_j \leq \hat{v}_j^\eta + \eta$ for all $j \in N$. In this case, using the fact that $\hat{v}_j^0 = \max\{r_j, \sum_{i\in N} \rho_{j,i}\, \hat{v}_i^0\}$, we obtain $\max\{r_j, \sum_{i\in N} \rho_{j,i}\, \tilde{v}_i\} \leq \max\{r_j, \sum_{i\in N} \rho_{j,i}\, \hat{v}_i^0\} = \hat{v}_j^0$ for all $j \in N$. Similarly, using the fact that $\hat{v}_j^\eta = \max\{r_j - \eta, \sum_{i\in N} \rho_{j,i}\, \hat{v}_i^\eta\}$, we have $\max\{r_j, \sum_{i\in N} \rho_{j,i}\, \tilde{v}_i\} \leq \max\{r_j, \sum_{i\in N} \rho_{j,i}\, (\hat{v}_i^\eta + \eta)\} \leq \max\{r_j - \eta, \sum_{i\in N} \rho_{j,i}\, \hat{v}_i^\eta\} + \eta = \hat{v}_j^\eta + \eta$ for all $j \in N$. Therefore, the discussion so far shows that $\max\{r_j, \sum_{i\in N} \rho_{j,i}\, \tilde{v}_i\} \leq \hat{v}_j^0$ and $\max\{r_j, \sum_{i\in N} \rho_{j,i}\, \tilde{v}_i\} \leq \hat{v}_j^\eta + \eta$, indicating that $\max\{r_j, \sum_{i\in N} \rho_{j,i}\, \tilde{v}_i\} \leq \min\{\hat{v}_j^0, \hat{v}_j^\eta + \eta\} = \tilde{v}_j$ for all $j \in N$, where the last equality is by the definition of $\tilde{v}$. Having $\max\{r_j, \sum_{i\in N} \rho_{j,i}\, \tilde{v}_i\} \leq \tilde{v}_j$ for all $j \in N$ implies that $\tilde{v}$ is a feasible solution to the Dual problem and the claim follows. To get a contradiction to the result that we want to show, we assume that there exists some $j \in N$ such that $j \in \{i \in N : \hat{v}_i^\eta = r_i - \eta\}$, but $j \notin \{i \in N : \hat{v}_i^0 = r_i\}$. So, we have $\hat{v}_j^\eta + \eta = r_j < \hat{v}_j^0$, in which case, noting that $\tilde{v}_j = \min\{\hat{v}_j^0, \hat{v}_j^\eta + \eta\}$, we get $\tilde{v}_j < \hat{v}_j^0$. Since $\tilde{v}_i \leq \hat{v}_i^0$ for all $i \in N$ by the definition of $\tilde{v}$, it follows that the components of $\tilde{v}$ are no larger than the corresponding components of $\hat{v}^0$ and there exists some $j \in N$ such that $\tilde{v}_j$



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

Table 1: A counterexample where a nested by revenue subset is not optimal.

is strictly smaller than $\hat{v}_j^0$. Since $\tilde{v}$ is a feasible solution to the Dual problem, the last observation contradicts the fact that $\hat{v}^0$ is the optimal solution to the Dual problem. □

If customers choose according to the multinomial logit model, then Talluri and van Ryzin (2004) show that the optimal solution to the Assortment problem includes a certain number of products with the largest revenues. In other words, if the products are indexed such that $r_1 \geq \ldots \geq r_n$, then a subset of the form $\{1, \ldots, j\}$ for some $j \in N$ is optimal. We refer to a subset that includes a certain number of products with the largest revenues as a nested by revenue subset. A natural question is whether a nested by revenue subset is optimal under the Markov chain choice model. We give a counterexample to show that a nested by revenue subset is not necessarily optimal under the Markov chain choice model. We consider a problem instance with three products. The revenues of the products are $(r_1, r_2, r_3) = (720, 225, 180)$. So, nested by revenue subsets for this problem instance are $\{1\}$, $\{1,2\}$ and $\{1,2,3\}$. The probabilities that a customer arrives into the system to purchase each one of the three products are $(\lambda_1, \lambda_2, \lambda_3) = (1/3, 1/3, 1/3)$. The transition probabilities are $\rho_{1,2} = \rho_{2,1} = \rho_{2,3} = \rho_{3,2} = 1/3$. The other transition probabilities in $\{\rho_{j,i} : j, i \in N\}$ are zero. So, for example, if a customer visits product 2 and product 2 is not available, then she transitions to the no purchase option with probability 1/3. For this problem instance, the left side of Table 1 shows the transition probabilities. The right side of Table 1 shows the purchase probabilities $(P_{1,S}, P_{2,S}, P_{3,S})$ when we offer each subset S and the expected revenue $\sum_{j=1}^{3} P_{j,S}\, r_j$ from each subset S. The purchase probabilities $(P_{1,S}, P_{2,S}, P_{3,S})$ are obtained by solving the Balance equations for $(P_{1,S}, P_{2,S}, P_{3,S})$ and $(R_{1,S}, R_{2,S}, R_{3,S})$. The results in Table 1 indicate the subset $\{1,3\}$ provides an expected revenue of 400, whereas the expected revenue from the best nested by revenue subset is 375. Therefore, a nested by revenue subset is not optimal. In Online Appendix A, we give a more involved counterexample to show that if there are n products, then the expected revenue provided by the best nested by revenue subset can deviate from the optimal expected revenue by a factor arbitrarily close to $n/2$, indicating that nested by revenue subsets can perform arbitrarily poorly under the Markov chain choice model.

It is tempting to come up with sufficient conditions to ensure that a nested by revenue subset is optimal under the Markov chain choice model. General sufficient conditions appear to be elusive. As shown in the previous paragraph, a nested by revenue subset is not necessarily optimal even when the Markov chain choice model follows a simple birth and death process. Blanchet et al. (2013) show that if the rows of the transition probability matrix $\{\rho_{j,i} : j, i \in N\}$ are equal to each other and each row is given by $(\lambda_1, \ldots, \lambda_n)$, then this Markov chain choice model is equivalent to a multinomial logit model. Since a nested by revenue subset is optimal under the multinomial logit model, it immediately follows that a nested by revenue subset is optimal under such a Markov chain choice model. In the next lemma, we give another sufficient condition to ensure that a nested by revenue subset is optimal. The proof of this lemma is deferred to Online Appendix B.

Lemma 4 *Assume that the products are indexed such that $r_1 \geq \ldots \geq r_n$ and the transition probabilities satisfy $\rho_{j,i} \leq \rho_{j+1,i}$ for all $j = 1, \ldots, n-1$, $i \in N$. Then, a nested by revenue subset is optimal for the* Assortment *problem.*

The rows of the transition probability matrix in Lemma 4 do not have to be equal to each other and the probabilities $(\lambda_1, \ldots, \lambda_n)$ can be arbitrary. Thus, this transition probability matrix can be quite different from the one that yields a choice model that is equivalent to the multinomial logit model, indicating that the purchase probabilities provided by the transition probability matrix in Lemma 4 can be quite different from the purchase probabilities under the multinomial logit model. It is difficult to give a behavioral justification for the sufficient condition in Lemma 4 but one implication of this sufficient condition is as follows. If product j is not available, then the probability that a customer transitions from product j to the no purchase option is given by $1 - \sum_{i \in N} \rho_{j,i}$. The sufficient condition in Lemma 4 implies that $1 - \sum_{i \in N} \rho_{j,i} \geq 1 - \sum_{j \in N} \rho_{j+1,i}$. Thus, if a customer visits a product with a larger revenue and this product is not available for purchase, then she has a larger probability of transitioning to the no purchase option.

4 Single Resource Revenue Management

In the single resource revenue management setting, we manage one resource with a limited amount of capacity. At each time period in the selling horizon, we need to decide which subset of products to offer to customers. Customers arrive into the system one by one and choose among the offered products according to the Markov chain choice model. When we sell a product, we generate a revenue and consume one unit of the resource. The goal is to find a policy to dynamically decide which subsets of products to offer over the selling horizon so as to maximize the total expected revenue. Single resource revenue management problems arise when airlines control the availability of different fare classes on a single flight leg. Different fare classes correspond to different products and the seats on the flight leg correspond to the resource. Talluri and van Ryzin (2004) consider such single resource revenue management problems when customers choose under a general choice model. In this section, we give structural properties of the optimal policy when customers choose under the Markov chain choice model. Similar to our notation in the previous section, we index the products by $N = \{1, \ldots, n\}$ and denote the revenue associated with product j by r_j. If we offer the subset S of products, then a customer purchases product j with probability $P_{j,S}$. We have T time periods in the selling horizon. Customers arrive one by one at each time period. We have c units of resource available at the beginning of the selling horizon. We let $V_t(x)$ be the optimal total expected revenue over the time periods $t, \ldots, T$, given that we have x units of remaining capacity at the beginning of time period t. We can compute $\{V_t(x) : x = 0, \ldots, c,\ t = 1, \ldots, T\}$ by solving the dynamic program

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

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

We let $\hat{S}_t(x)$ be the optimal subset of products to offer given that we have x units of remaining capacity at the beginning of time period t, in which case, $\hat{S}_t(x)$ is given by the optimal solution to the problem on the right side of the Single Resource dynamic program. In this section, we show that there exists an optimal policy that satisfies the properties $\hat{S}_t(x-1) \subset \hat{S}_t(x)$ and $\hat{S}_{t-1}(x) \subset \hat{S}_t(x)$. The first property implies that if we have fewer units of remaining capacity at a particular time period, then the optimal subset of products to offer becomes smaller. The second property implies that if we have more time periods left in the selling horizon with a particular number of units of remaining capacity, then the optimal subset of products to offer becomes smaller. The first property has an important implication when implementing the optimal policy. Since the optimal subset of products to offer becomes smaller when we have fewer units of remaining capacity at a time period, we let $\bar{x}_{jt}$ be the smallest value of the remaining capacity such that it is still optimal to offer product j at time period t. In this case, if we have x units of remaining capacity at the beginning of time period t and $x \geq \bar{x}_{jt}$, then it is optimal to offer product j. Otherwise, it is optimal not to offer product j. So, we can associate a threshold value $\bar{x}_{jt}$ for each product j and time period t such that we can decide whether it is optimal to offer product j at time period t by comparing the remaining resource capacity with the threshold value. The threshold value $\bar{x}_{jt}$ is referred to as the protection level for product j at time period t and the resulting policy is referred to as a protection level policy. Due to the optimality of a protection level policy, we can separately decide whether to offer each product by comparing the remaining resource capacity with the threshold value of the product. In the next theorem, we show that there exists an optimal policy that satisfies the properties described at the beginning of this paragraph.

Theorem 5 *There exists an optimal policy for the* Single Resource *dynamic program such that* $\hat{S}_t(x-1) \subset \hat{S}_t(x)$ *and* $\hat{S}_{t-1}(x) \subset \hat{S}_t(x)$.

Proof. It is a standard result that the first differences of the value functions computed through the Single Resource dynamic program increase as we have fewer units of remaining capacity or as we have more time periods left in the selling horizon. In particular, letting $\Delta V_t(x) = V_t(x) - V_t(x-1)$, Talluri and van Ryzin (2004) show that $\Delta V_{t+1}(x) \leq \Delta V_{t+1}(x-1)$ and $\Delta V_{t+1}(x) \leq \Delta V_t(x)$ under any choice model. Letting $r_{jt}(x) = r_j - \Delta V_{t+1}(x)$, by definition, $\hat{S}_t(x)$ is the optimal solution to the problem $\max_{S \subset N} \sum_{j \in N} P_{j,S} \, (r_j - \Delta V_{t+1}(x)) = \max_{S \subset N} \sum_{j \in N} P_{j,S} \, r_{jt}(x)$, whereas $\hat{S}_t(x-1)$ is the optimal solution to the problem $\max_{S \subset N} \sum_{j \in N} P_{j,S} \, (r_j - \Delta V_{t+1}(x-1)) = \max_{S \subset N} \sum_{j \in N} P_{j,S} \, (r_{jt}(x) - (\Delta V_{t+1}(x-1) - \Delta V_{t+1}(x)))$. Identifying $r_{jt}(x)$ with the revenue of product j in the ASSORTMENT problem, the problem that computes $\hat{S}_t(x)$ has the same form as the ASSORTMENT problem. The problem that computes $\hat{S}_t(x-1)$ also has the same form as the ASSORTMENT problem as long as we identify $r_{jt}(x) - (\Delta V_{t+1}(x-1) - \Delta V_{t+1}(x))$ with the revenue of product j in the ASSORTMENT problem. Thus, the revenue of each product in the problem that computes $\hat{S}_t(x-1)$ is obtained by subtracting $\Delta V_{t+1}(x-1) - \Delta V_{t+1}(x)$ from the revenue of the corresponding product in the problem that computes $\hat{S}_t(x)$. By the discussion at the beginning of the proof, we have $\Delta V_{t+1}(x-1) - \Delta V_{t+1}(x) \geq 0$ and Lemma 3 implies that if we decrease the revenue of each product by the same positive amount, then the optimal solution to the ASSORTMENT problem becomes a smaller subset. Thus, it follows that $\hat{S}_t(x-1) \subset \hat{S}_t(x)$. Following the same argument but using the fact that the first differences of the value functions increase as we have more time periods left in the selling horizon, we can show that $\hat{S}_{t-1}(x) \subset \hat{S}_t(x)$. □

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

For this problem instance, Table 2 shows the purchase probabilities $(P_{1,S}, P_{2,S}, P_{3,S})$ along with the expected revenues $\sum_{j \in N} P_{j,S} \, r_j$ and $\sum_{j \in N} P_{j,S} \, (r_j - V_2(1))$ for each subset S. Since there are two time periods in the selling horizon, we have $V_3(x) = 0$ for all $x = 0, 1, 2$ and $V_t(0) = 0$ for all $t = 1, 2$. So, the SINGLE RESOURCE dynamic program implies that $V_2(x) = \max_{S \subset N} \sum_{j \in N} P_{j,S} \, r_j$ for all $x = 1, 2$. Noting Table 2, we have $V_2(x) = \max_{S \subset N} \sum_{j \in N} P_{j,S} \, r_j = 140$ for all $x = 1, 2$. Consider the optimal subset of products to offer at time period one. If we have two units of remaining capacity at time period one, then noting that $V_2(2) = V_2(1) = 140$, we solve the problem

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

Table 2: A counterexample where we stop offering a product with larger revenue before we stop offering a product with smaller revenue as we have fewer units of remaining capacity.

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

By Theorem 5, the optimal subsets of products to offer with different units of remaining capacities are nested so that $\hat{S}_t(0) \subset \hat{S}_t(1) \subset \ldots \subset \hat{S}_t(c)$, but our counterexample shows that the nesting order of the subsets may not follow the revenue order of the products. We give a sufficient condition on the Markov chain choice model to ensure that the nesting order of the subsets $\{\hat{S}_t(x) : x = 0,\ldots,c\}$ follows the revenue order of the products. We index the products such that $r_1 \geq \ldots \geq r_n$. In the next lemma, we show that if the sufficient condition in Lemma 4 holds, then $\hat{S}_t(x)$ is of the form $\{1,\ldots,j_t(x)\}$, where $j_t(x)$ is increasing in x. Since $j_t(x)$ is increasing in x, we have $\{1,\ldots,j_t(x-1)\} \subset \{1,\ldots,j_t(x)\}$. So, if we have fewer units of remaining capacity at time period t, then we stop offering the products with smaller revenues before we stop offering the products with larger revenues. We defer the proof to Online Appendix C.

Lemma 6 *Assume that the products are indexed such that $r_1 \geq \ldots \geq r_n$ and the transition probabilities satisfy $\rho_{j,i} \leq \rho_{j+1,i}$ for all $j = 1,\ldots,n-1$, $i \in N$. Then, we have $\hat{S}_t(x) = \{1,\ldots,j_t(x)\}$ for some $j_t(x) \in N \cup \{0\}$, where $j_t(x) = 0$ corresponds to the case with $\hat{S}_t(x) = \varnothing$. Furthermore, $j_t(x)$ is increasing in x and t.*

Since $j_t(x)$ is increasing in t, Lemma 6 also implies that if we have more time periods left in the selling horizon with x units of remaining capacity, then we stop offering the products with smaller revenues before we stop offering the products with larger revenues.

5 Network Revenue Management

In the network revenue management setting, we manage a network of resources, each of which has a limited amount of capacity. At each time period in the selling horizon, we need to decide which subset of products to offer to customers. Customers arrive into the system one by one and choose among the offered products according to the Markov chain choice model. Each product uses a certain combination of resources. When we sell a product, we generate a revenue and consume the capacities of the resources used by the product. The goal is to find a policy to dynamically decide which subsets of products to make available over the selling horizon so as to maximize the total expected revenue. Network revenue management problems model the situation faced by an airline making itinerary availability decisions over a network of flight legs. Different itineraries correspond to different products and seats on different flight legs correspond to different resources. Gallego et al. (2004) and Liu and van Ryzin (2008) formulate a linear programming approximation to the network revenue management problem under the assumption that the customer choices take on their expected values. In this section, we show that the size of the linear program can be drastically reduced when customers choose under the Markov chain choice model.

There are m resources indexed by $M = \{1, \ldots, m\}$. Similar to the earlier sections, we index the products by $N = \{1, \ldots, n\}$. Each customer chooses among the offered products according to the Markov chain choice model. There are T time periods in the selling horizon. For simplicity of notation, we assume that at most one customer arrives at each time period. We have c_q units of resource q available at the beginning of the selling horizon. If we sell one unit of product j, then we generate a revenue of r_j and consume $a_{q,j}$ units of resource q. Using the decision variable u_S to capture the probability of offering subset S of products at a time period, we consider a linear programming approximation to the network revenue management problem given by

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

Noting the definition of the decision variable u_S, the expression $\sum_{S \subset N} T\, P_{j,S}\, u_S$ corresponds to the total expected number of sales for product j over the selling horizon. Thus, the objective function of the problem above computes the total expected revenue over the selling horizon. The first constraint ensures that the total expected capacity of resource q consumed over the selling horizon does not exceed the capacity of this resource. The second constraint ensures that we offer a subset of products with probability one at each time period, but this subset can be the empty set. Liu and van Ryzin (2008) propose approximate policies that use the solution to the Choice Based linear program to decide which subset of products to offer at each time period.

In our formulation of the Choice Based linear program, the probabilities (P_S, R_S) do not depend on the time period. If the customers arriving at different time periods choose according to choice models with different parameters, then we can use the decision variable $u_{S,t}$, which corresponds to the probability of offering subset S of products at time period t. Since the Choice Based linear program has one decision variable for each subset of products, its number of decision variables grows exponentially with the number of products. Therefore, a common approach for solving this problem is to use column generation. In this section, we show that if customers choose according to the Markov chain choice model, then the Choice Based linear program can equivalently be formulated as a linear program with only $2n$ decision variables and $m+n$ constraints. To give the equivalent formulation, we use the decision variable x_j to capture the expected number of times that a customer visits product j that is available for purchase. Also, we use the decision variable z_j to capture the expected number of times that a customer visits product j that is unavailable for purchase. Our main result in this section shows that the Choice Based linear program is equivalent to the linear program

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

Whenever a customer visits an available product, she purchases it. So, the decision variable x_j also captures the fraction of customers that purchase product j, in which case, $T x_j$ corresponds to the total expected number of sales for product j over the selling horizon. Thus, the objective function of the problem above computes the total expected revenue over the selling horizon. The first constraint ensures that the total expected capacity of resource q consumed over the selling horizon does not exceed the capacity of this resource. The second constraint is similar to the Balance equations. Noting that $x_j + z_j$ is the expected number of times a customer visits product j, the interpretation of this constraint is identical to the interpretation of the Balance equations given in Section 1. Note that there are $2n$ decision variables and $m+n$ constraints in the Reduced linear program. In the next theorem, we show that the Reduced linear program is equivalent to the Choice Based linear program in the sense that we can use the optimal solution to the Reduced linear program to obtain the optimal solution to the Choice Based one.

Theorem 7 *Letting $(\hat{x}, \hat{z})$ be the optimal solution to the* Reduced *linear program, there exist subsets $S^1, \ldots, S^K$ and positive scalars $\gamma^1, \ldots, \gamma^K$ summing to one such that $\hat{x} = \sum_{k=1}^K \gamma^k P_{S^k}$ and $\hat{z} = \sum_{k=1}^K \gamma^k R_{S^k}$. In this case, the optimal objective values of the* Reduced *and* Choice Based *linear programs are the same and the solution $\hat{u}$ obtained by letting $\hat{u}_{S^k} = \gamma^k$ for all $k = 1, \ldots, K$ and $\hat{u}_S = 0$ for all $S \notin \{S^1, \ldots, S^K\}$ is optimal to the* Choice Based *linear program.*

Proof. Noting the definition of $\mathcal{H}$ in Section 2, since $(\hat{x}, \hat{z})$ is a feasible solution to the Reduced linear program, we have $(\hat{x}, \hat{z}) \in \mathcal{H}$. Furthermore, Lemma 10 in Online Appendix D shows that $\mathcal{H}$ is bounded. Thus, there exist extreme points $(x^1, z^1), \ldots, (x^K, z^K)$ of $\mathcal{H}$ and positive scalars

$\gamma^1, \ldots, \gamma^K$ summing to one such that $\hat{x} = \sum_{k=1}^K \gamma^k \, x^k$ and $\hat{z} = \sum_{k=1}^K \gamma^k \, z^k$. For all $k = 1, \ldots, K$, define the subset $S^k = \{ j \in N : x_j^k > 0 \}$. By Lemma 1, we have $P_{j,S^k} = x_j^k$ and $R_{j,S^k} = z_j^k$ for all $j \in N$, which implies that $\hat{x} = \sum_{k=1}^K \gamma^k \, x^k = \sum_{k=1}^K \gamma^k \, P_{S^k}$ and $\hat{z} = \sum_{k=1}^K \gamma^k \, z^k = \sum_{k=1}^K \gamma^k \, R_{S^k}$, establishing the first part of the theorem. To show the second part of the theorem, we use $\hat{\zeta}_C$ and $\hat{\zeta}_R$ to respectively denote the optimal objective values of the Choice Based linear program and the Reduced linear program. First, we show that $\hat{\zeta}_C \geq \hat{\zeta}_R$. By the definition of $\hat{u}$ in the theorem, we have $\sum_{S \subset N} P_{j,S} \, \hat{u}_S = \sum_{k=1}^K P_{j,S^k} \, \gamma^k = \hat{x}_j$. Thus, we obtain $\sum_{j \in N} \sum_{S \subset N} T \, a_{q,j} \, P_{j,S} \, \hat{u}_S = \sum_{j \in N} T \, a_{q,j} \, \hat{x}_j \leq c_q$, where the inequality uses the fact that $(\hat{x}, \hat{z})$ is a feasible solution to the Reduced linear program. Furthermore, we have $\sum_{S \subset N} \hat{u}_S = \sum_{k=1}^K \gamma^k = 1$. Therefore, $\hat{u}$ is a feasible solution to the Choice Based linear program. Since $\sum_{S \subset N} P_{j,S} \, \hat{u}_S = \hat{x}_j$, we obtain $\hat{\zeta}_R = \sum_{j \in N} T \, r_j \, \hat{x}_j = \sum_{S \subset N} \sum_{j \in N} T \, r_j \, P_{j,S} \, \hat{u}_S \leq \hat{\zeta}_C$, where the inequality uses the fact that $\hat{u}$ is a feasible solution to the Choice Based linear program.

Second, we show that $\hat{\zeta}_C \leq \hat{\zeta}_R$. Let $\tilde{u}$ be the optimal solution to the Choice Based linear program. Define $(\tilde{x}, \tilde{z})$ as $\tilde{x}_j = \sum_{S \subset N} P_{j,S} \, \tilde{u}_S$ and $\tilde{z}_j = \sum_{S \subset N} R_{j,S} \, \tilde{u}_S$ for all $j \in N$. We have $\sum_{j \in N} T \, a_{q,j} \, \tilde{x}_j = \sum_{S \subset N} \sum_{j \in N} T \, a_{q,j} \, P_{j,S} \, \tilde{u}_S \leq c_q$, where we use the fact that $\tilde{u}$ satisfies the first constraint in the Choice Based linear program. Since (P_S, R_S) satisfies the Balance equations, we have $P_{j,S} + R_{j,S} = \lambda_j + \sum_{i \in N} \rho_{i,j} \, R_{i,S}$ for all $S \subset N$. Multiplying this equality by $\tilde{u}_S$, adding over all $S \subset N$ and noting that $\tilde{u}$ is a feasible solution to the Choice Based linear program satisfying $\sum_{S \subset N} \tilde{u}_S = 1$, we get $\tilde{x}_j + \tilde{z}_j = \sum_{S \subset N} P_{j,S} \, \tilde{u}_S + \sum_{S \subset N} R_{j,S} \, \tilde{u}_S = \lambda_j + \sum_{S \subset N} \sum_{i \in N} \rho_{i,j} \, R_{i,S} \, \tilde{u}_S = \lambda_j + \sum_{i \in N} \rho_{i,j} \, \tilde{z}_i$, where the first and last equalities use the definitions of $\tilde{x}$ and $\tilde{z}$. Thus, $(\tilde{x}, \tilde{z})$ is a feasible solution to the Reduced linear program. Since $\tilde{x}_j = \sum_{S \subset N} P_{j,S} \, \tilde{u}_S$, we obtain $\hat{\zeta}_C = \sum_{S \subset N} \sum_{j \in N} T \, r_j \, P_{j,S} \, \tilde{u}_S = \sum_{j \in N} T \, r_j \, \tilde{x}_j \leq \hat{\zeta}_R$, where the inequality uses the fact that $(\tilde{x}, \tilde{z})$ is a feasible solution to the Reduced linear program. Therefore, we have $\hat{\zeta}_C = \hat{\zeta}_R$ and the solution $\hat{u}$ defined in the theorem must be optimal to the Choice Based linear program. □

By Theorem 7, we have a general approach for recovering the optimal solution to the Choice Based linear program by using the optimal solution to the Reduced linear program. Letting $(\hat{x}, \hat{z})$ be the optimal solution to the Reduced linear program, the key is to find subsets $S^1, \ldots, S^K$ and positive scalars $\gamma^1, \ldots, \gamma^K$ summing to one such that $(\hat{x}, \hat{z})$ can be expressed as $\hat{x} = \sum_{k=1}^K \gamma^k \, P_{S^k}$ and $\hat{z} = \sum_{k=1}^K \gamma^k \, R_{S^k}$. In the next section, we give a tractable algorithm for this purpose. This algorithm also demonstrates that the number of subsets $S^1, \ldots, S^K$ satisfies $K \leq n + 1$.

6 Recovering the Optimal Solution

In this section, we consider the question of recovering the optimal solution to the Choice Based linear program by using the optimal solution to the Reduced linear program. Noting Theorem 7 and the discussion at the end of the previous section, letting $(\hat{x}, \hat{z})$ be the optimal solution to the Reduced linear program, it is enough to find subsets $S^1, \ldots, S^K$ and positive scalars $\gamma^1, \ldots, \gamma^K$ summing to one such that $(\hat{x}, \hat{z})$ can be expressed as $\hat{x} = \sum_{k=1}^K \gamma^k \, P_{S^k}$ and $\hat{z} = \sum_{k=1}^K \gamma^k \, R_{S^k}$. One possible approach to find the subsets $S^1, \ldots, S^K$ and the scalars $\gamma^1, \ldots, \gamma^K$ is to express $(\hat{x}, \hat{z})$

as $\hat{x} = \sum_{S \subset N} \gamma_S \, P_S$ and $\hat{z} = \sum_{S \subset N} \gamma_S \, R_S$ for unknown positive scalars $\{\gamma_S : S \subset N\}$ satisfying $\sum_{S \subset N} \gamma_S = 1$ and solve these equations for the unknown scalars. Since there are $2n+1$ equations, there exists a solution where at most $2n+1$ of the unknown scalars take strictly positive values, but solving these equations directly is not tractable since the number of unknown scalars grows exponentially with the number of products. We can use an idea similar to column generation, where we focus on a small subset of the unknown scalars $\{\gamma_S : S \subset N\}$ and iteratively extend this subset of scalars, but this approach can be as computationally intensive as solving the Choice Based linear program directly by using column generation.

In this section, we give a tractable dimension reduction approach to find subsets $S^1, \ldots, S^K$ and positive scalars $\gamma^1, \ldots, \gamma^K$ summing to one such that the optimal solution $(\hat{x}, \hat{z})$ to the Reduced linear program can be expressed as $\hat{x} = \sum_{k=1}^K \gamma^k \, P_{S^k}$ and $\hat{z} = \sum_{k=1}^K \gamma^k \, R_{S^k}$. Our dimension reduction approach uses the following recursive idea. Since $(\hat{x}, \hat{z})$ is a feasible solution to the Reduced linear program, we have $(\hat{x}, \hat{z}) \in \mathcal{H}$. We define $S_{\hat{x}} = \{j \in N : \hat{x}_j > 0\}$ to capture the nonzero components of $\hat{x}$ and consider three cases. In the first case, we assume that $S_{\hat{x}} = \varnothing$, implying that $\hat{x}_j = 0$ for all $j \in N$. Since $(\hat{x}, \hat{z}) \in \mathcal{H}$ and $\hat{x}_j = 0$ for all $j \in N$, $(\hat{x}, \hat{z})$ satisfies the Balance equations with $S = \varnothing$, so that we must have $(\hat{x}, \hat{z}) = (P_\varnothing, R_\varnothing)$. Thus, we can express $(\hat{x}, \hat{z})$ simply as $\hat{x} = P_\varnothing$ and $\hat{z} = R_\varnothing$ and we are done. In the rest of the discussion, we assume that $S_{\hat{x}} \neq \varnothing$. We define $\alpha_{\hat{x}} = \min\{\hat{x}_j \,/ P_{j,S_{\hat{x}}} : j \in S_{\hat{x}}\}$. Using the fact that $(\hat{x}, \hat{z}) \in \mathcal{H}$, if we add the equalities in the definition of $\mathcal{H}$ over all $j \in N$ and arrange the terms, then we get $\sum_{j \in N} \hat{x}_j + \sum_{j \in N} (1 - \sum_{i \in N} \rho_{j,i}) \, \hat{z}_j = \sum_{j \in N} \lambda_j$. Adding the Balance equations over all $j \in N$, we also get $\sum_{j \in N} \lambda_j = \sum_{j \in N} P_{j,S_{\hat{x}}} + \sum_{j \in N} (1 - \sum_{i \in N} \rho_{j,i}) \, R_{j,S_{\hat{x}}}$. Noting that $\hat{x}_j = 0 = P_{j,S_{\hat{x}}}$ for all $j \notin S_{\hat{x}}$, the last two equalities yield $\sum_{j \in S_{\hat{x}}} \hat{x}_j + \sum_{j \in N} (1 - \sum_{i \in N} \rho_{j,i}) \, \hat{z}_j = \sum_{j \in S_{\hat{x}}} P_{j,S_{\hat{x}}} + \sum_{j \in N} (1 - \sum_{i \in N} \rho_{j,i}) \, R_{j,S_{\hat{x}}}$. Since $(\hat{x}, \hat{z}) \in \mathcal{H}$, Lemma 11 in Online Appendix E implies that $\hat{z}_j \geq R_{j,S_{\hat{x}}}$ for all $j \in N$, in which case, the last equality implies that there exists some $j \in S_{\hat{x}}$ such $\hat{x}_j \leq P_{j,S_{\hat{x}}}$. Therefore, we must have $\alpha_{\hat{x}} = \min\{\hat{x}_j \,/ P_{j,S_{\hat{x}}} : j \in S_{\hat{x}}\} \leq 1$.

In the second case, we assume that $\alpha_{\hat{x}} = 1$. Having $\alpha_{\hat{x}} = 1$ implies that $\hat{x}_j \geq P_{j,S_{\hat{x}}}$ for all $j \in S_{\hat{x}}$. By the discussion at the end of the previous paragraph, we observe that $\hat{z}_j \geq R_{j,S_{\hat{x}}}$ for all $j \in N$ and $\sum_{j \in S_{\hat{x}}} \hat{x}_j + \sum_{j \in N} (1 - \sum_{i \in N} \rho_{j,i}) \, \hat{z}_j = \sum_{j \in S_{\hat{x}}} P_{j,S_{\hat{x}}} + \sum_{j \in N} (1 - \sum_{i \in N} \rho_{j,i}) \, R_{j,S_{\hat{x}}}$. Noting that we have $\hat{x}_j \geq P_{j,S_{\hat{x}}}$ for all $j \in S_{\hat{x}}$ and $\hat{z}_j \geq R_{j,S_{\hat{x}}}$ for all $j \in N$, the last equality implies that $\hat{x}_j = P_{j,S_{\hat{x}}}$ for all $j \in S_{\hat{x}}$ and $\hat{z}_j = R_{j,S_{\hat{x}}}$ for all $j \in N$. Furthermore, by the definition of $S_{\hat{x}}$ and the Balance equations, we also have $\hat{x}_j = 0 = P_{j,S_{\hat{x}}}$ for all $j \notin S_{\hat{x}}$. Thus, we have $\hat{x}_j = P_{j,S_{\hat{x}}}$ for all $j \in N$. Since we have $\hat{x}_j = P_{j,S_{\hat{x}}}$ and $\hat{z}_j = R_{j,S_{\hat{x}}}$ for all $j \in N$, we can express $(\hat{x}, \hat{z})$ simply as $\hat{x} = P_{S_{\hat{x}}}$ and $\hat{z} = R_{S_{\hat{x}}}$ and we are done.

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

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

Lemma 8 *For $(\hat{x}, \hat{z}) \in \mathcal{H}$, define $S_{\hat{x}} = \{j \in N : \hat{x}_j > 0\}$, $\alpha_{\hat{x}} = \min\{\hat{x}_j \,/ P_{j,S_{\hat{x}}} : j \in S_{\hat{x}}\}$ and $j_{\hat{x}} = \arg\min\{\hat{x}_j \,/ P_{j,S_{\hat{x}}} : j \in S_{\hat{x}}\}$. Assuming that $\alpha_{\hat{x}} < 1$, define $(\hat{u}, \hat{v})$ as*

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

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

Proof. First, we show that $(\hat{u}, \hat{v}) \in \mathcal{H}$. We observe that $(\hat{u}, \hat{v})$ is obtained by multiplying $(\hat{x}, \hat{z})$ by $1/(1 - \alpha_{\hat{x}})$ and $(P_{S_{\hat{x}}}, R_{S_{\hat{x}}})$ by $-\alpha_{\hat{x}}/(1 - \alpha_{\hat{x}})$ and adding them up. Using the fact that $(\hat{x}, \hat{z}) \in \mathcal{H}$ and noting the Balance equations, we have $\hat{x}_j + \hat{z}_j = \lambda_j + \sum_{i \in N} \rho_{i,j} \, \hat{z}_i$ and $P_{j,S_{\hat{x}}} + R_{j,S_{\hat{x}}} = \lambda_j + \sum_{i \in N} \rho_{i,j} \, R_{i,S_{\hat{x}}}$ for all $j \in N$. Multiplying these two equations respectively by $1/(1 - \alpha_{\hat{x}})$ and $-\alpha_{\hat{x}}/(1 - \alpha_{\hat{x}})$ and adding, it follows that $(\hat{u}, \hat{v})$ satisfies $\hat{u}_j + \hat{v}_j = \lambda_j + \sum_{i \in N} \rho_{i,j} \, \hat{v}_i$ for all $j \in N$. Thus, to show that $(\hat{u}, \hat{v}) \in \mathcal{H}$, it is enough to check that $\hat{u}_j \geq 0$ and $\hat{v}_j \geq 0$ for all $j \in N$. Since $\alpha_{\hat{x}} \leq \hat{x}_j \,/ P_{j,S_{\hat{x}}}$ for all $j \in S_{\hat{x}}$, we have $\hat{u}_j = (\hat{x}_j - \alpha_{\hat{x}} \, P_{j,S_{\hat{x}}})/(1 - \alpha_{\hat{x}}) \geq 0$ for all $j \in S_{\hat{x}}$. By definition, we have $\hat{x}_j = 0 = P_{j,S_{\hat{x}}}$ for all $j \notin S_{\hat{x}}$. Thus, we also have $\hat{u}_j = (\hat{x}_j - \alpha_{\hat{x}} \, P_{j,S_{\hat{x}}})/(1 - \alpha_{\hat{x}}) = 0$ for all $j \notin S_{\hat{x}}$. By Lemma 11 in Online Appendix E, we have $\hat{z}_j \geq R_{j,S_{\hat{x}}}$ for all $j \in N$. Using the assumption that $\alpha_{\hat{x}} < 1$, we have $\hat{v}_j = (\hat{z}_j - \alpha_{\hat{x}}\, R_{j,S_{\hat{x}}})/(1-\alpha_{\hat{x}}) \geq 0$ for all $j \in N$ and the first part of the lemma follows. Second, we show that $S_{\hat{u}} \subset S_{\hat{x}} \setminus \{j_{\hat{x}}\}$. Consider $j \in S_{\hat{u}}$ so that $\hat{u}_j > 0$. Since $\hat{u}_j = (\hat{x}_j - \alpha_{\hat{x}}\, P_{j,S_{\hat{x}}})/(1-\alpha_{\hat{x}}) > 0$, it must be the case that $\hat{x}_j > 0$, indicating that $j \in S_{\hat{x}}$. So, we get $S_{\hat{u}} \subset S_{\hat{x}}$. Furthermore, we have $j_{\hat{x}} \notin S_{\hat{u}}$ since $u_{j_{\hat{x}}} = (\hat{x}_{j_{\hat{x}}} - \alpha_{\hat{x}}\, P_{j_{\hat{x}},S_{\hat{x}}})/(1-\alpha_{\hat{x}}) = 0$, where the second equality uses the fact that $\alpha_{\hat{x}} = \hat{x}_{j_{\hat{x}}} / P_{j_{\hat{x}},S_{\hat{x}}}$. Thus, we obtain $S_{\hat{u}} \subset S_{\hat{x}} \setminus \{j_{\hat{x}}\}$. □

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

Dimension Reduction

Step 0. Using $(\hat{x}, \hat{z})$ to denote the optimal solution to the Reduced linear program, set $(x^1, z^1) = (\hat{x}, \hat{z})$. Initialize the iteration counter by setting $k = 1$.

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

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

Step 3. Set (x^{k+1}, z^{k+1}) as

$$x_j^{k+1} = \frac{x_j^k - \alpha^k\, P_{j,S^k}}{1-\alpha^k} \quad \text{and} \quad z_j^{k+1} = \frac{z_j^k - \alpha^k\, R_{j,S^k}}{1-\alpha^k}$$

for all $j \in N$.

Step 4. Increase k by one and go to Step 1.

Steps 1, 2 and 3 in the Dimension Reduction algorithm respectively correspond to the first, second and third cases considered at the beginning of this section. We can use induction over the iterations of the algorithm to show that $(x^k, z^k) \in \mathcal{H}$ at each iteration k. In particular, since $(\hat{x}, \hat{z})$ is a feasible solution to the Reduced linear program, we have $(x^1, z^1) \in \mathcal{H}$. Assuming that $(x^k, z^k) \in \mathcal{H}$, if we identify (x^k, z^k) and (x^{k+1}, z^{k+1}) in the algorithm respectively with $(\hat{x}, \hat{z})$ and $(\hat{u}, \hat{v})$ in Lemma 8, then this lemma implies that $(x^{k+1}, z^{k+1}) \in \mathcal{H}$, completing the induction. Letting $j^k = \arg\min\{x_j^k / P_{j,S^k} : j \in S^k\}$ and noting that $(x^k, z^k) \in \mathcal{H}$, if we identify S^k and S^{k+1} in the algorithm with $S_{\hat{x}}$ and $S_{\hat{u}}$ in Lemma 8, then this lemma also implies that $S^{k+1} \subset S^k \setminus \{j^k\}$. In the next theorem, we show that the Dimension Reduction algorithm achieves its goal.

Theorem 9 *Assume that the* Dimension Reduction *algorithm stops at iteration K, generating the subsets $S^1, \ldots, S^K$ and the scalars $\alpha^1, \ldots, \alpha^K$. Then, letting $\gamma^k = (1-\alpha^1) \ldots (1-\alpha^{k-1})\, \alpha^k$ for all $k = 1, \ldots, K$, we have $\hat{x} = \sum_{k=1}^K \gamma^k\, P_{S^k}$, $\hat{z} = \sum_{k=1}^K \gamma^k\, R_{S^k}$ and $\sum_{k=1}^K \gamma^k = 1$. Furthermore, the* Dimension Reduction *algorithm stops after at most $n+1$ iterations and the subsets generated by this algorithm satisfy $S^1 \supset S^2 \supset \ldots \supset S^K$.*

Proof. We use induction over the iterations of the Dimension Reduction algorithm to show that $\hat{x} = \gamma^1\, P_{S^1} + \ldots + \gamma^{k-1}\, P_{S^{k-1}} + (\gamma^k/\alpha^k)\, x^k$ for all $k = 1, \ldots, K$. Since $\gamma^1 = \alpha^1$, we have

$\hat{x} = x^1 = (\gamma^1/\alpha^1)\, x^1$, so that the result holds at the first iteration. We assume that the result holds at iteration k. Noting Step 3 of the algorithm, we have $x^k = \alpha^k\, P_{S^k} + (1-\alpha^k)\, x^{k+1}$. In this case, since $\hat{x} = \gamma^1\, P_{S^1} + \ldots + \gamma^{k-1}\, P_{S^{k-1}} + (\gamma^k/\alpha^k)\, x^k$ by the induction assumption, we have

$$\hat{x} = \gamma^1\, P_{S^1} + \ldots + \gamma^{k-1}\, P_{S^{k-1}} + \frac{\gamma^k}{\alpha^k}\,(\alpha^k\, P_{S^k} + (1-\alpha^k)\, x^{k+1}) \\ = \gamma^1\, P_{S^1} + \ldots + \gamma^k\, P_{S^k} + \frac{\gamma^{k+1}}{\alpha^{k+1}}\, x^{k+1},$$

where the second equality uses the fact that $(\gamma^k/\alpha^k)\,(1-\alpha^k) = \gamma^{k+1}/\alpha^{k+1}$ by the definition of γ^k. Therefore, the result holds at iteration $k+1$ and the induction is complete. A similar argument also shows that $\hat{z} = \gamma^1\, R_{S^1} + \ldots + \gamma^{k-1}\, R_{S^{k-1}} + (\gamma^k/\alpha^k)\, z^k$. On the other hand, we note that the algorithm stops in either Step 1 or Step 2. If the algorithm stops in Step 1, then $S^K = \varnothing$ and $\alpha^K = 1$. By the discussion that follows the description of the Dimension Reduction algorithm, we have $(x^K, z^K) \in \mathcal{H}$, in which case, we can follow the same argument in the first case at the beginning of this section to conclude that $(x^K, z^K) = (P_\varnothing, R_\varnothing)$. Thus, we obtain $(x^K, z^K) = (P_{S^K}, R_{S^K})$. If the algorithm stops in Step 2, then $\alpha^K = 1$ and we can follow the same argument in the second case at the beginning of this section to conclude that $(x^K, z^K) = (P_{S^K}, R_{S^K})$. Therefore, we always have $(x^K, z^K) = (P_{S^K}, R_{S^K})$ and $\alpha^K = 1$ when the Dimension Reduction algorithm stops. In this case, using the equalities $\hat{x} = \gamma^1\, P_{S^1} + \ldots + \gamma^{k-1}\, P_{S^{k-1}} + (\gamma^k/\alpha^k)\, x^k$ and $\hat{z} = \gamma^1\, R_{S^1} + \ldots + \gamma^{k-1}\, R_{S^{k-1}} + (\gamma^k/\alpha^k)\, z^k$ with $k = K$, we obtain $\hat{x} = \sum_{k=1}^K \gamma^k\, P_{S^k}$ and $\hat{z} = \sum_{k=1}^K \gamma^k\, R_{S^k}$. Finally, noting that $\alpha^K = 1$ and using the definition of γ^k in the theorem, it follows that $\gamma^{K-1} + \gamma^K = (1-\alpha^1)\ldots(1-\alpha^{K-2})$. Repeating the argument recursively, we obtain $\gamma^2 + \ldots + \gamma^K = (1-\alpha^1)$, in which case, since $\gamma^1 = \alpha^1$ by the definition of γ^k, we get $\gamma^1 + \gamma^2 + \ldots + \gamma^K = 1$, establishing the first part of the theorem. By the discussion right before the theorem, we have $S^{k+1} \subset S^k \setminus \{j^k\}$, which implies that $|S^{k+1}| < |S^k|$. Since $|S^1| \le n$, the algorithm must stop after generating at most $n+1$ subsets and the subsets generated by the algorithm satisfy $S^1 \supset S^2 \supset \ldots \supset S^K$. $\square$

Theorem 9 shows that we can use the Dimension Reduction algorithm to find subsets $S^1, \ldots, S^K$ and positive scalars $\gamma^1, \ldots, \gamma^K$ summing to one such that the optimal solution $(\hat{x}, \hat{z})$ to the Reduced linear program can be expressed as $\hat{x} = \sum_{k=1}^K \gamma^k\, P_{S^k}$ and $\hat{z} = \sum_{k=1}^K \gamma^k\, R_{S^k}$. In this case, Theorem 7 implies that we only offer the subsets $S^1, \ldots, S^K$ of products in the optimal solution to the Choice Based linear program. Noting that the Dimension Reduction algorithm stops after at most $n+1$ iterations, we have $K \le n+1$, which implies that there exists an optimal solution $\hat{u}$ to the Choice Based linear program, where at most $n+1$ of the decision variables can take on nonzero values. Furthermore, letting $\hat{u}_{S^1}, \ldots, \hat{u}_{S^{n+1}}$ be these decision variables, we have $S^1 \supset \ldots \supset S^{n+1}$, since the subsets generated by the Dimension Reduction algorithm are nested. Therefore, a practical implication of Theorem 9 is that the optimal solution to the Choice Based linear program randomizes over offering at most $n+1$ subsets and the offered subsets are related in the sense that one subset is included in another one. Lastly, it turns out that the Dimension Reduction algorithm has an appealing geometric interpretation and we elaborate on this interpretation in Online Appendix F.

7 Capturing Customer Choices through the Markov Chain Choice Model

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

7.1 Experimental Setup

In our computational experiments, we generate the past purchase history of the customers under the assumption that the customers choose according to a complicated ground choice model that does not comply with the Markov chain choice model or the multinomial logit model. We fit both a Markov chain choice model and a multinomial logit model to the past purchase history of the customers and compare the two fitted choice models. In the ground choice model that governs the customer choices, there are m customer types. A customer of a particular type considers a particular subset of products, ranks these products according to a particular order and purchases the offered product that has the highest ranking in her order. If none of the products considered by the customer is offered, then the customer leaves without a purchase. We use β^g to denote the probability that a customer of type g arrives into the system. To denote the subset of products that a customer of type g considers and the rankings of these products, we use the tuple $(j^g(1), \ldots, j^g(k^g))$ such that a customer of type g considers the subset of products $\{j^g(1), \ldots, j^g(k^g)\} \subset N$ and product $j^g(1)$ is her highest ranking product, whereas product $j^g(k^g)$ is her lowest ranking product.

In our computational experiments, we use the following approach to generate the tuple $(j^g(1), \ldots, j^g(k^g))$ for each customer type g. We assume that there is an overarching order between the quality levels of the products. The products that have higher quality levels also have higher prices. Customers of a particular type have a maximum willingness to pay amount and a minimum acceptable quality level such that they only consider the products whose prices are below their maximum willingness to pay amount and whose quality levels are above their minimum acceptable quality level. To introduce some idiosyncrasy in the choice behavior of customers, we assume that customers of a particular type drop some of the products from consideration, even if the prices of these products are below their maximum willingness to pay amount and the quality levels of these products are above their minimum acceptable quality level. Furthermore, we assume that customers of a particular type occasionally flip the order of some of the products in the overarching quality order so that they occasionally deviate from the overarching quality order. Among the products that they consider, customers of a particular type prefer a product with a higher quality level to a product with a lower quality level.

To be specific, we index the products such that product 1 has the highest quality level and the highest price, whereas product n has the lowest quality level and the lowest price. To generate the tuple $(j^g(1), \ldots, j^g(k^g))$ for customer type g, we sample product ℓ_L^g from the uniform distribution

$\{1, \dots, n\}$ and product ℓ_U^g from the uniform distribution over $\{\ell_L^g, \dots, n\}$. Customers of type g do not consider products whose prices are higher than the price of product ℓ_L^g and whose quality levels are lower than the quality level of product ℓ_U^g. We drop each one of the products in the subset $\{\ell_L^g, \dots, \ell_U^g\}$ with probability 0.1 to obtain the subset of products $\{i^g(1), \dots, i^g(k^g)\}$. We assume that the products $\{i^g(1), \dots, i^g(k^g)\}$ are indexed such that product $i^g(1)$ has the highest quality level and the highest price, whereas product $i^g(k^g)$ has the lowest quality level and the lowest price. With probability 0.5, we set the tuple $(j^g(1), \dots, j^g(k^g))$ to be $(i^g(1), \dots, i^g(k^g))$. With probability 0.5, we randomly choose one product in the set $\{i^g(1), \dots, i^g(k^g - 1)\}$ and flip its ordering in the tuple $(i^g(1), \dots, i^g(k^g))$ with its successor to obtain the tuple $(j^g(1), \dots, j^g(k^g))$. If the tuple $(j^g(1), \dots, j^g(k^g))$ is identical to the tuple for an another customer type, then we regenerate the tuple to ensure that the tuples for the different customer types are different.

To generate the arrival probabilities $(\beta^1, \dots, \beta^m)$, we set $\beta_g = 1/m$ so that customers of each type arrive with equal probability. In all of our computational experiments, we set the number of products to $n = 10$ and the number of customer types to $m = 100$. Once we generate the ground choice model, we generate the past purchase history of the customers under the assumption that the customers choose among the offered products according to this ground choice model. We capture the past purchase history of τ customers by using $\{(S_t, j_t) : t = 1, \dots, \tau\}$, where S_t is the subset of products offered to customer t and j_t is the product purchased by customer t. If customer t leaves without a purchase, then we have $j_t = 0$. To generate the past purchase history of the customers, we sample the subset S_t such that each product is included in this subset with probability 0.5. Using g_t to denote the type of customer t in the past purchase history, we sample g_t by using the probabilities $(\beta^1, \dots, \beta^m)$. In this case, the product j_t purchased by customer t is given by the first product in the tuple $(j^{g_t}(1), \dots, j^{g_t}(k^{g_t}))$ that is also in the subset S_t. If none of the products in the tuple $(j^{g_t}(1), \dots, j^{g_t}(k^{g_t}))$ is in the subset S_t, then the customer leaves without a purchase.

7.2 Fitting Choice Models

We fit a Markov chain choice model and a multinomial logit model to the past purchase history of the customers and compare the two fitted choice models. We follow four steps. (1) We generate the ground choice model that governs the choice process of the customers by using the approach described in the previous section. Throughout our computational experiments, the ground choice model that governs the choice process of the customers is fixed. (2) Using the approach described in the previous section, we generate the past purchase history of τ customers when the customers choose according to the ground choice model. We use H_τ to denote the past purchase history of τ customers. We generate three past purchase histories by varying the number of customers over $\tau \in \{1000, 1750, 2500\}$, corresponding to three different levels of data availability when we fit a Markov chain choice model and a multinomial logit model. We refer to each one of these past purchase histories as a training data set. (3) We use maximum likelihood estimation to fit a Markov chain choice model and a multinomial logit model to each one of the training data sets. We use MC_τ

and ML_τ to respectively denote the Markov chain choice model and the multinomial logit model that are fitted to the training data set H_τ. (4) We compare the fitted choice models MC_τ and ML_τ for $\tau \in \{1000, 1750, 2500\}$ through two performance measures. First, we check how well the two fitted choice models allow us to predict the purchases of the customers that are not in the past purchase history. Second, we compute the optimal subsets in the Assortment problem under the assumption that the customers choose according to either one of the two fitted choice models and check the expected revenues obtained by offering these subsets.

Given that customers choose according to the Markov chain choice model characterized by the parameters $\lambda = (\lambda_1, \ldots, \lambda_n)$ and $\rho = \{\rho_{j,i} : j, i \in N\}$, we use $P_{j,S}(\lambda, \rho)$ to denote the probability that a customer purchases product j when we offer the subset S of products. As a function of the parameters λ and ρ, the log likelihood of the training data set $H_\tau = \{(S_t, j_t) : t = 1, \ldots, \tau\}$ can be written as $\mathcal{L}(\lambda, \rho \,|\, H_\tau) = \sum_{t=1}^{\tau} \log P_{j_t, S_t}(\lambda, \rho)$. To obtain the maximum likelihood estimators of the parameters λ and ρ, we maximize the log likelihood $\mathcal{L}(\lambda, \rho \,|\, H_\tau)$. One approach to maximize the log likelihood is to compute the derivatives of $P_{j_t, S_t}(\lambda, \rho)$ with respect to λ_j and $\rho_{j,i}$ for all $j, i \in N$ and use gradient search. In particular, using the vectors $P_S(\lambda, \rho) = (P_{1,S}(\lambda, \rho), \ldots, P_{n,S}(\lambda, \rho))$ and $R_S(\lambda, \rho) = (R_{1,S}(\lambda, \rho), \ldots, R_{n,S}(\lambda, \rho))$, we let $(P_S(\lambda, \rho), R_S(\lambda, \rho))$ be the solution to the Balance equations, where we make the dependence of the solution on λ and ρ explicit. Differentiating both sides of the Balance equations with respect to λ_j and $\rho_{j,i}$ provides systems of equations that should be satisfied by the derivatives of $P_{j,S}(\lambda, \rho)$ and $R_{j,S}(\lambda, \rho)$ with respect to λ_j and $\rho_{j,i}$. By using the same approach in Section 1, we can show that if $\sum_{i \in N} \rho_{j,i} < 1$ for all $j \in N$, then these systems of equations have unique solutions. We elaborate on this approach in Online Appendix G. Another approach to maximize the log likelihood function is to use an optimization routine that only needs evaluating the value of the objective function, rather than its derivatives. In our computational experiments, we use `fmincon` routine in `Matlab`, which is an optimization routine that only needs evaluating the value of the objective function. This approach turns out to be adequate to demonstrate the benefits from fitting a Markov chain choice model, instead of a simpler choice model such as the multinomial logit model. Our main focus in this paper is on solving various revenue management problems under the Markov chain choice model and sophisticated methods for obtaining the maximum likelihood estimators of the parameters of the Markov chain choice model is an important research question that remains open for further investigation.

7.3 Predicting Customer Purchases

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

Grn. Cho. Mod.	$\tau = 1000$			$\tau = 1750$			$\tau = 2500$		
	Like. MC_τ	Like. ML_τ	Perc. Gap	Like. MC_τ	Like. ML_τ	Perc. Gap	Like. MC_τ	Like. ML_τ	Perc. Gap
1	−3966	−4061	2.40	−3910	−4053	3.65	−3892	−4047	3.98
2	−4130	−4231	2.45	−4082	−4226	3.52	−4074	−4221	3.61
3	−4123	−4206	1.99	−4097	−4199	2.48	−4072	−4194	3.01
4	−3963	−4081	2.98	−3956	−4084	3.23	−3946	−4081	3.43
5	−4027	−4130	2.55	−4014	−4126	2.79	−3999	−4126	3.18
6	−4049	−4152	2.54	−4016	−4141	3.10	−3998	−4139	3.52
7	−4005	−4123	2.94	−4008	−4116	2.68	−3978	−4118	3.51
8	−4003	−4122	2.96	−4005	−4117	2.78	−3993	−4116	3.08
9	−4047	−4205	3.91	−4052	−4203	3.72	−4024	−4197	4.30
10	−4041	−4113	1.77	−4015	−4103	2.18	−3991	−4101	2.76
Avg.	−4035	−4142	2.65	−4016	−4137	3.01	−3997	−4134	3.44

Table 3: Log likelihoods of the testing data set under the fitted choice models.

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

Table 3 shows our computational results. Each row in Table 3 shows the results for each one of the 10 ground choice models. There are three blocks of three consecutive columns in Table 3. Each block corresponds to one of the three values for $\tau \in \{1000, 1750, 2500\}$, capturing different levels of data availability. Given that each block corresponds to one value of τ, the first column in each block shows the log likelihood of the testing data set under the assumption that the customers choose according to the Markov chain choice model MC_τ estimated from the past purchase history H_τ, whereas the second column in each block shows the log likelihood of the testing data set under the assumption that the customers choose according to the multinomial logit model ML_τ estimated from the past purchase history H_τ. In particular, for any choice model $\mathsf{CM} \in \{\mathsf{MC}_\tau, \mathsf{ML}_\tau\}$, we let $P_{j,S}(\mathsf{CM})$ be the probability that a customer purchases product j when we offer the subset S of products, given that customers choose under the choice model CM. Also, we use $\{(S_t, j_t) : t = 1, \ldots, 2500\}$ to denote the subsets of offered products and the products purchased by the customers in the testing data set. In this case, the first and second columns in each block show the log likelihoods $\sum_{t=1}^{2500} \log P_{j_t,S_t}(\mathsf{MC}_\tau)$ and $\sum_{t=1}^{2500} \log P_{j_t,S_t}(\mathsf{ML}_\tau)$. Finally, the third column in each block shows the percent gap between the first and second columns.

The Markov chain choice model has $O(n^2)$ parameters, corresponding to $(\lambda_1, \ldots, \lambda_n)$ and $\{\rho_{j,i} : j, i \in N\}$, whereas the multinomial logit model has $O(n)$ parameters, corresponding to the mean utilities of the products. Since the Markov chain choice model has a larger number of parameters, it provides more flexibility for modeling the choice behavior of customers. However, due to its large number of parameters, the Markov chain choice model may overfit to the training data set especially when there are too few customers in the training data set, in which case, it may not provide satisfactory performance on the testing data set. In Table 3, the log likelihoods of the testing data set under the fitted Markov chain choice model are larger than those under the fitted multinomial logit model. Thus, overfitting does not seem to be a concern for the Markov chain choice model when we have as few as 1000 customers in the training data set and as many as about 100 parameters to be estimated. Nevertheless, it is useful to keep in mind that overfitting is a potential concern for the Markov chain choice model and the multinomial logit model may provide better performance when we have too few customers in the training data set and we have too many parameters to be estimated. Section 1.1 in Bishop (2006) explains that overfitting may be a concern when working with a model with a large number of parameters, but this concern goes away when there are abundant training data. Therefore, the Markov chain choice model may be preferable when the training data are abundant, but the multinomial logit model may be a more viable option when the training data are scarce. Also, we observe as the data availability in the training data set increases, the log likelihood of the testing data set under the fitted Markov chain choice model increases more noticeably when compared with that under the fitted multinomial logit model. For example, the average log likelihood under the fitted Markov chain choice model increases from -4035 to -3997 as τ increases from 1000 to 2500, but the average log likelihood under the fitted multinomial logit model increases from -4142 to -4134. Thus, the Markov chain choice model seems to benefit from additional data availability more. Lastly, since the Markov chain choice model has more parameters than the multinomial logit model and we maximize the log likelihood of the training data set when fitting the two choice models, we expect that the Markov chain choice model provides a larger log likelihood on the training data set when compared with the multinomial logit model. However, since the Markov chain choice model may overfit to the training data set, it is not guaranteed that it provides a larger log likelihood on the testing data set. So, the larger number of parameters in the Markov chain choice model is expected to translate into a larger log likelihood on the training data set, but not necessarily on the testing data set. Beside the concern for overfitting when we have too few customers in the training data set, there are other factors that may make the multinomial logit model preferable to the Markov chain choice model. We explain these factors as we discuss our conclusions in Section 9.

For each ground choice model and for each value of $\tau \in \{1000, 1750, 2500\}$, the results in Table 3 consider one training data set. To check that our results do not change considerably from one training data set to another, we repeated our computational experiments for 10 different training data sets. We give the results of these computational experiments in Online Appendix H. Also, we repeated our computational experiments under censored demands as in Vulcano et al. (2012). The Markov chain choice model continued to provide improvements for the levels of data availability and the numbers of parameters to be estimated in our test problems.

7.4 Finding Good Assortments

Another approach for comparing the Markov chain choice model MC_τ and the multinomial logit model ML_τ that are fitted to the training data set H_τ is based on checking how well the two fitted choice models allow us to solve the ASSORTMENT problem. For this purpose, we generate 100 samples of the product revenues. We use $\{(r_1^k, \ldots, r_n^k) : k = 1, \ldots, 100\}$ to denote the samples of the product revenues, where we sample r_j^k from the uniform distribution over $[0, 100]$. For each sample of the product revenues $(r_1^k, \ldots, r_n^k)$, we compute the optimal subset in the ASSORTMENT problem under the assumption the customers choose according to either one of the two fitted choice models. In particular, we let $S^k(\mathsf{MC}_\tau) = \arg\max_{S \subset N} \sum_{j \in N} P_{j,S}(\mathsf{MC}_\tau)\, r_j^k$ and $S^k(\mathsf{ML}_\tau) = \arg\max_{S \subset N} \sum_{j \in N} P_{j,S}(\mathsf{ML}_\tau)\, r_j^k$. In the last two problems, for any choice model $\mathsf{CM} \in \{\mathsf{MC}_\tau, \mathsf{ML}_\tau\}$, the purchase probability $P_{j,S}(\mathsf{CM})$ is as defined in the previous section. Since the customers actually choose according to the ground choice model, if we offer the subset $S^k(\mathsf{MC}_\tau)$ or $S^k(\mathsf{ML}_\tau)$ of products, then we obtain an actual expected revenue of $\mathsf{Rev}^k(\mathsf{MC}_\tau) = \sum_{j \in N} P_{j,S^k(\mathsf{MC}_\tau)}(\mathsf{GC})\, r_j^k$ or $\mathsf{Rev}^k(\mathsf{ML}_\tau) = \sum_{j \in N} P_{j,S^k(\mathsf{ML}_\tau)}(\mathsf{GC})\, r_j^k$, where we use $P_{j,S}(\mathsf{GC})$ to denote the probability that a customer purchases product j when we offer the subset S of products, given that customers choose under the ground choice model. Our goal is to compare the actual expected revenues $\{\mathsf{Rev}^k(\mathsf{MC}_\tau) : k = 1, \ldots, 100\}$ and $\{\mathsf{Rev}^k(\mathsf{ML}_\tau) : k = 1, \ldots, 100\}$ obtained under 100 samples of the product revenues. Similar to our results in the previous section, we repeat our computational experiments for 10 different ground choice models we generate.

Table 4 shows our computational results. Each row in Table 4 shows the results for each one of the 10 ground choice models. There are three blocks of five consecutive columns in Table 4 and each of these three blocks corresponds to one of the three values for $\tau \in \{1000, 1750, 2500\}$. The first column in each block shows the average of $\{\mathsf{Rev}^k(\mathsf{MC}_\tau) : k = 1, \ldots, 100\}$, which is the average expected revenue over all product revenue samples when we use the optimal subset that is computed under the assumption that the customers choose according to the Markov chain choice model MC_τ fitted to the past purchase history. The second column in each block shows the average of $\{\mathsf{Rev}^k(\mathsf{ML}_\tau) : k = 1, \ldots, 100\}$. The third column in each block shows the percent gap between the first and second columns. The fourth column in each block shows the number of samples of product revenues for which $\mathsf{Rev}^k(\mathsf{MC}_\tau) > \mathsf{Rev}^k(\mathsf{ML}_\tau)$. That is, this column shows $\sum_{k=1}^{100} \mathbf{1}(\mathsf{Rev}^k(\mathsf{MC}_\tau) > \mathsf{Rev}^k(\mathsf{ML}_\tau))$, where $\mathbf{1}(\cdot)$ is the indicator function. The fifth column in each block shows the number of samples of product revenues for which $\mathsf{Rev}^k(\mathsf{MC}_\tau) < \mathsf{Rev}^k(\mathsf{ML}_\tau)$.

In Table 4, for the levels of data availability and the numbers of parameters to be estimated in our test problems, the expected revenues from the subsets that are computed under the assumption that the customers choose according to the fitted Markov chain choice model are often larger than those computed under the assumption that the customers choose according to the fitted multinomial logit model. When we have as few as 1000 customers in the training data set and as many as about 100 parameters to be estimated, the subsets obtained under the fitted Markov chain choice model provide an expected revenue improvement of 6.39% on average, but if the training data are too

Grn. Cho. Mod.	$\tau = 1000$ MC$_\tau$ Exp. Rev.	ML$_\tau$ Exp. Rev.	Per. Gap.	MC$_\tau$ $\succ$ ML$_\tau$	MC$_\tau$ $\prec$ ML$_\tau$	$\tau = 1750$ MC$_\tau$ Exp. Rev.	ML$_\tau$ Exp. Rev.	Per. Gap.	MC$_\tau$ $\succ$ ML$_\tau$	MC$_\tau$ $\prec$ ML$_\tau$	$\tau = 2500$ MC$_\tau$ Exp. Rev.	ML$_\tau$ Exp. Rev.	Per. Gap.	MC$_\tau$ $\succ$ ML$_\tau$	MC$_\tau$ $\prec$ ML$_\tau$
1	78.49	72.71	7.37	65	15	78.68	72.76	7.51	65	14	79.06	72.82	7.89	66	8
2	75.87	71.42	5.87	67	13	76.18	71.56	6.06	70	16	76.36	71.45	6.43	70	10
3	72.76	69.99	3.82	50	34	72.42	69.94	3.42	51	33	73.33	70.44	3.93	58	23
4	77.78	73.69	5.26	55	15	78.18	73.52	5.96	57	8	78.19	73.52	5.96	62	7
5	74.92	69.44	7.32	66	27	75.42	69.56	7.77	72	13	75.63	69.13	8.59	74	8
6	76.18	71.65	5.96	65	19	76.85	71.87	6.49	72	17	77.50	71.76	7.40	75	9
7	74.40	70.44	5.31	69	16	74.83	69.95	6.52	75	10	74.99	69.39	7.47	81	7
8	74.42	68.10	8.50	63	10	75.07	68.25	9.09	72	10	75.05	68.57	8.64	73	8
9	77.11	71.49	7.29	61	15	77.68	71.54	7.90	66	10	78.11	71.61	8.32	74	5
10	74.39	69.04	7.19	59	20	74.82	69.18	7.54	60	6	74.87	69.19	7.58	57	7
Avg.	75.63	70.79	6.39	62	18	76.01	70.81	6.83	66	13	76.31	70.79	7.22	69	9

Table 4: Expected revenues from the optimal subsets computed under the fitted choice models.

scarce, then the Markov chain choice model may overfit to the training data set and the subsets obtained under the fitted the multinomial logit model may provide larger expected revenues. For each ground choice model and for each value of $\tau \in \{1000, 1750, 2500\}$, the results in Table 4 consider one training data set. We repeated our computational experiments for 10 different training data sets. We give the results of these computational experiments in Online Appendix I.

8 Performance Improvements for Network Revenue Management

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

8.1 Experimental Setup

In our computational experiments, we generate a number of network revenue management problem instances. For each problem instance, we use two strategies to obtain the optimal solution to the Choice Based linear program. The first strategy solves the Choice Based linear program directly by using column generation. We refer to this strategy as CG, standing for column generation. The second strategy solves the Reduced linear program and carries out the Dimension Reduction algorithm to recover the optimal solution to the Choice Based linear program by using the optimal solution to the Reduced linear program. We refer to this strategy as DR, standing for dimension reduction. Our objective is to compare the performances of CG and DR. To generate our test problems, we sample β_j from the uniform distribution over $[0, 1]$ and set $\lambda_j = \beta_j / \sum_{i \in N} \beta_i$. Similarly, we sample $\zeta_{j,i}$ from the uniform distribution over $[0, 1]$ and set $\rho_{j,i} = (1 - P_0)\, \zeta_{j,i} / \sum_{k \in N} \zeta_{j,k}$, where P_0 is a parameter that we vary. In this case, if a customer visits product j during the course of her choice process and product j is not available, then she leaves without making a purchase with probability $1-\sum_{i\in N}\rho_{j,i} = 1-(1-P_0)\sum_{i\in N}\zeta_{j,i}/\sum_{k\in N}\zeta_{j,k} = P_0$. Thus, the parameter P_0 controls the tendency of the customers to leave without a purchase.

In all of our problem instances, we normalize the length of the selling horizon to $T = 100$. We sample the revenue r_j of product j from the uniform distribution over $[200, 600]$. For each product j, we randomly choose a resource q_j and set $a_{q_j,j} = 1$. For the other resources, we set $a_{q,j} = 1$ with probability ξ and $a_{q,j} = 0$ with probability $1-\xi$ for all $q \in M \setminus \{q_j\}$, where ξ is a parameter that we vary. Therefore, the expected number of resources used by a product is $1+(m-1)\,\xi$ and ξ controls the expected number of resources used by a product. To generate the capacities of the resources, we let S^* be the optimal solution to the problem $\max_{S\subset N}\sum_{j\in N} r_j\, P_{j,S}$, so that S^* is the subset of products that maximizes the expected revenue when we have unlimited resource capacities. We set the capacity of resource q as $c_q = \kappa\sum_{j\in N} T\, a_{q,j}\, P_{j,S^*}$, where κ is another parameter that we vary. We note that $\sum_{j\in N} T\, a_{q,j}\, P_{j,S^*}$ is the total expected capacity consumption of resource q when we offer the subset S^* of products. So, the capacity of resource q is obtained by multiplying this total expected capacity consumption by κ and the parameter κ controls the tightness of the resource capacities. We vary (m, n, P_0, ξ, κ) over $\{25, 50\} \times \{250, 500\} \times \{0.1, 0.3\} \times \{0.02, 0.2\} \times \{0.6, 0.8\}$, where m is the number of resources, n is the number of products and the parameters P_0, ξ and κ are as defined in this section. This setup yields 32 problem instances.

8.2 Computational Results

Table 5 summarizes our main computational results. The first column in this table shows the problem instances by using the tuple (m, n, P_0, ξ, κ). The second column shows the CPU seconds for CG. The third column shows the CPU seconds for DR. The fourth column shows the ratio of the CPU seconds for CG and DR. The fifth column shows what portion of the CPU seconds for DR is spent on solving the Reduced linear program, whereas the sixth column shows what portion of the CPU seconds for DR is spent on carrying out the Dimension Reduction algorithm. The results in Table 5 indicate that DR can provide remarkable performance improvements over CG. Over all of our test problems, DR improves the CPU seconds for CG by an average factor of 1835. There are test problems where the CPU seconds for CG exceed two hours of run time. For these test problems, DR can obtain the optimal solution to the Choice Based linear program within only 2.5 seconds. About 80% of the CPU seconds for DR is spent on solving the Reduced linear program and the rest is spent on carrying out the Dimension Reduction algorithm. In Table 5, we observe two trends in the performance improvements provided by DR. First, as the number of resources and the number of products increase and the size of the problem becomes larger, the performance improvements provided by DR become more pronounced. For the smallest test problems with $m = 25$ and $n = 250$, DR improves the CPU seconds for CG by an average factor of 503, whereas for the largest test problems with $m = 50$ and $n = 500$, DR improves the CPU seconds for CG by an average factor of 4229. Second, as κ decreases and the capacities of the resources become tighter, the performance improvement provided by DR also tends to increase. For the test problems

Test Problem (m, n, P_0, ξ, κ)	CPU Secs. CG	CPU Secs. DR	CPU Secs. Ratio	Red. Lin.	Dim. Red.
(25, 250, 0.1, 0.02, 0.8)	162	0.23	722	0.18	0.05
(25, 250, 0.1, 0.02, 0.6)	200	0.35	571	0.27	0.08
(25, 250, 0.1, 0.2, 0.8)	112	0.25	447	0.20	0.05
(25, 250, 0.1, 0.2, 0.6)	170	0.22	772	0.18	0.04
(25, 250, 0.3, 0.02, 0.8)	59	0.22	270	0.17	0.05
(25, 250, 0.3, 0.02, 0.6)	137	0.19	722	0.15	0.04
(25, 250, 0.3, 0.2, 0.8)	36	0.26	139	0.22	0.04
(25, 250, 0.1, 0.2, 0.6)	104	0.27	384	0.23	0.04
(25, 500, 0.1, 0.02, 0.8)	1575	1.58	997	1.22	0.36
(25, 500, 0.1, 0.02, 0.6)	1957	1.36	1439	0.98	0.38
(25, 500, 0.1, 0.2, 0.8)	1772	2.00	886	1.63	0.37
(25, 500, 0.1, 0.2, 0.6)	2487	1.63	1526	1.24	0.39
(25, 500, 0.3, 0.02, 0.8)	556	1.61	345	1.31	0.30
(25, 500, 0.3, 0.02, 0.6)	1174	1.52	772	1.18	0.34
(25, 500, 0.3, 0.2, 0.8)	559	1.99	281	1.67	0.32
(25, 500, 0.1, 0.2, 0.6)	1383	2.01	688	1.66	0.35
Avg.	778	0.98	685	0.78	0.20

Test Problem (m, n, P_0, ξ, κ)	CPU Secs. CG	CPU Secs. DR	CPU Secs. Ratio	Red. Lin.	Dim. Red.
(50, 250, 0.1, 0.02, 0.8)	410	0.22	1831	0.18	0.05
(50, 250, 0.1, 0.02, 0.6)	926	0.22	4208	0.16	0.06
(50, 250, 0.1, 0.2, 0.8)	398	0.29	1374	0.24	0.05
(50, 250, 0.1, 0.2, 0.6)	468	0.25	1901	0.20	0.05
(50, 250, 0.3, 0.02, 0.8)	213	0.26	820	0.22	0.04
(50, 250, 0.3, 0.02, 0.6)	387	0.22	1758	0.18	0.04
(50, 250, 0.3, 0.2, 0.8)	155	0.28	552	0.24	0.04
(50, 250, 0.3, 0.2, 0.6)	389	0.26	1495	0.21	0.05
(50, 500, 0.1, 0.02, 0.8)	6628	1.75	3788	1.40	0.35
(50, 500, 0.1, 0.02, 0.6)	15500	1.50	10333	1.11	0.39
(50, 500, 0.1, 0.2, 0.8)	7739	2.12	3651	1.74	0.38
(50, 500, 0.1, 0.2, 0.6)	10659	1.90	5610	1.50	0.40
(50, 500, 0.3, 0.02, 0.8)	3008	1.84	1635	1.54	0.30
(50, 500, 0.3, 0.02, 0.6)	6780	1.64	4134	1.29	0.35
(50, 500, 0.3, 0.2, 0.8)	2392	2.32	1031	1.99	0.33
(50, 500, 0.3, 0.2, 0.6)	7600	2.08	3654	1.71	0.37
Avg.	3,978	1.07	2,986	0.96	0.11

Table 5: CPU seconds for CG and DR.

with $\kappa = 0.8$, DR improves the CPU seconds for CG by an average factor of 1173, whereas for the test problems with $\kappa = 0.6$, DR improves the CPU seconds for CG by an average factor of 2498. Overall, DR can be faster than CG by orders of magnitude.

Column generation approaches can be fast in obtaining a near optimal solution, but they can also be slow in ultimately reaching the optimal solution, which can be a factor in the relatively poor performance of CG in Table 5. We checked the CPU seconds for CG to obtain a solution that provides an objective value within 1% of the optimal objective value of the Choice Based linear program. We provide a few summary statistics on the performance of CG and defer the details to Online Appendix J. If we stop CG when it obtains a solution within 1% of the optimal objective value, but we continue using DR to obtain the optimal solution to the Choice Based linear program, then, on average, the CPU seconds for DR improve the CPU seconds for CG by a factor of 286. There are test problems, where DR improves the CPU seconds for CG by factors exceeding 1000. Thus, DR continues to provide substantial improvements over CG even when we terminate CG once it obtains a solution within 1% of the optimal objective value.

Table 6 provides computational experiments for test problems with 100 resources and 2000 products. For these test problems, we vary (P_0, ξ, κ) over $\{0.1, 0.3\} \times \{0.02, 0.2\} \times \{0.6, 0.8\}$. We note that these test problems are significantly larger than the earlier ones. For these test problems, two hours of run time is not enough for CG to provide a solution within 1% of the optimal objective value of the Choice Based linear program. The first column in Table 6 lists the test problems by using the tuple (m, n, P_0, ξ, κ). The second column shows the optimality gaps obtained by CG after two hours of run time. The third column shows the CPU seconds for DR to obtain the optimal solution to the Choice Based linear program. On average, DR obtains the optimal solution to

Test Problem (m, n, P_0, ξ, κ)	CG % Opt. Gap	DR CPU Secs.
$(100, 2000, 0.1, 0.02, 0.8)$	5.87	158.58
$(100, 2000, 0.1, 0.02, 0.6)$	12.53	144.50
$(100, 2000, 0.1, \ \ 0.2, 0.8)$	5.38	194.69
$(100, 2000, 0.1, \ \ 0.2, 0.6)$	10.27	165.95
Avg.	8.51	165.93

Test Problem (m, n, P_0, ξ, κ)	CG % Opt. Gap	DR CPU Secs.
$(100, 2000, 0.3, 0.02, 0.8)$	4.89	194.75
$(100, 2000, 0.3, 0.02, 0.6)$	9.69	224.79
$(100, 2000, 0.3, \ \ 0.2, 0.8)$	2.83	173.70
$(100, 2000, 0.3, \ \ 0.2, 0.6)$	6.13	178.08
Avg.	5.89	192.83

Table 6: Optimality gaps for CG when we terminate it after two hours of run time and CPU seconds for DR to obtain the optimal solution.

the Choice Based linear program in less than three minutes. Over all of our test problems, the average optimality gap of the solutions obtained by CG even after two hours of run time is 7.2%. There are test problems where CG terminates with more than 10% optimality gap after two hours of run time. In contrast, the largest CPU seconds for DR is about 225. Our results indicate that DR continues to provide substantial improvements over CG for larger test problems.

9 Conclusions

We studied assortment optimization, single resource revenue management and network revenue management problems under the Markov chain choice model. Our results gave a tractable solution method for the assortment optimization problem, provided structural properties of the optimal policy for the single resource revenue management problem and showed how to reduce the size of a deterministic approximation for the network revenue management problem. As mentioned in Section 1, the assumption that $\lambda_j > 0$ for all $j \in N$ allows us provide succinct proofs for our results, but our results continue to hold with some modifications when we have $\lambda_j = 0$ for some $j \in N$. We give the details of these modifications in Online Appendix K. In our computational experiments, we discussed that the Markov chain choice model may suffer from overfitting especially when there are too few customers in the training data set and we have too many parameters to be estimated. For the levels of data availability and the numbers of parameters to be estimated in our test problems, overfitting did not seem to be a concern for the Markov chain choice model, but the multinomial logit model may be preferable to the Markov chain choice model when the training data are scarce. Below, we discuss two other factors that may make the multinomial logit model preferable.

First, an important feature of the multinomial logit model is that one can estimate the mean utility of a product as a function of its attributes. In the airline setting, the products take the form of tickets for different itineraries and the attributes of a product could be its price, number of connections, total layover duration and change fees. The advantage of using features is that if a new product is introduced into the market or if the attributes of an existing product changes, then one can compute the mean utility of a product without having to reestimate the parameters of the choice model. In addition, knowing how the mean utility of a product depends on its attributes sheds light into the most critical product attributes. Markov chain choice model and the ranking based choice model that we used as the ground choice model in our computational experiments do not take advantage of the attributes of the products. The transition probabilities in the Markov chain choice model, to some extent, capture the degree of substitution between the products and it would be useful to come up with representations of the Markov chain choice model that can take advantage of the attributes of the products.

Second, both the multinomial logit model and the Markov chain choice model are compatible with the random utility maximization approach, but the distribution of the utilities of the products under the multinomial logit model is much better understood. Under the multinomial logit model, the utilities of the products are independent of each other and they have Gumbel distribution. There are also other choice models whose distributions of utilities are well understood, while allowing different dependence structures for the utilities. Under the nested logit model, the products are partitioned into nests and the utilities of the products in the same nest are dependent on each other, but the utilities of the products in different nests are independent. This feature allows us to capture the situation where if a customer likes a certain product, then she is more likely to like another similar product. Under the paired combinatorial logit model, there is a nest for each pair of products, which allows the utilities of any pair of products to be dependent on each other. Under a mixture of multinomial logit models, there are multiple customer types and customers of different types choose according to different multinomial logit models with different parameters. Train (2003) discusses these choice models in detail. If we work with the multinomial logit, nested logit, paired combinatorial logit or a mixture of multinomial logit models, then some knowledge about the customer population and the products among which the customers choose may help us pick a reasonable choice model. Also, since the distributions of the utilities under these choice models are well understood, after estimating the parameters of these choice models, we can check whether the parameter estimates are sensible. As mentioned in the previous paragraph, the transition probabilities in the Markov chain choice model may give a feel for the degree of substitution between the products, but after estimating the parameters of this choice model, it is still more difficult to check whether the parameter estimates are sensible.

One can extend the Markov chain choice model to limit the number of transitions of a customer so that the customer leaves the system without a purchase when the number of transitions reaches a certain limit. Computing the purchase probabilities of the products under this version of the Markov chain choice model is not difficult. In particular, we let $P^k_{j,S}$ be the probability that a customer visits the available product j in transition k, whereas we let $R^k_{j,S}$ be the probability that a customer visits the unavailable product j in transition k. Using m to denote the limit on the number of transitions, these probabilities satisfy a version of the BALANCE equations given by $P^1_{j,S} + R^1_{j,S} = \lambda_j$ for all $j \in N$, $P^k_{j,S} + R^k_{j,S} = \sum_{i \in N} \rho_{i,j} \, R^{k-1}_{i,S}$ for all $j \in N$, $k = 2, \ldots, m$, $P^k_{j,S} = 0$ for all $j \notin S$, $k = 1, \ldots, m$ and $R^k_{j,S} = 0$ for all $j \in S$, $k = 1, \ldots, m$. For example, to see the interpretation of the second one of these equations, we observe that $P^k_{j,S} + R^k_{j,S}$ on the left side corresponds to the probability that a customer visits product j in transition k. To visit product j in transition k, the customer should visit some unavailable product i in transition $k-1$ and transition from product i to product j, yielding $\sum_{i \in N} \rho_{i,j} \, R^{k-1}_{i,S}$ on the right side. However, finding the optimal subset of products to offer in the Assortment problem is difficult. In particular, Lemma 1, which is critical to all of our results in the paper, does not hold when we have a limit on the number of transitions. We can verify that if we are allowed to offer different subsets to customers in different transitions, then Lemma 1 has a natural extension to the case where we have a limit on the number of transitions, but this extension assumes that there are subsets $S^1, \ldots, S^m$ such that we offer subset S^k to a customer in transition k. This assumption is not realistic since we cannot change the subset of offered products during the choice process of a customer. Therefore, if we have a limit on the number of transitions, then the main difficulty in the Assortment problem is due to the fact that the same subset of products should be offered to customers in different transitions. It may be possible to come up with an integer programming formulation of the Assortment problem to ensure that the same subset of products should be offered to customers in different transitions. Such an integer programming formulation may allow solving the Assortment problem when we have a limit on the number of transitions, but extensions to the single resource and network revenue management problems are still difficult since the use of duality theory on the linear programming formulation of the Assortment problem plays a critical role in our results for the single resource and network revenue management problems. More work is needed in this direction.

We used maximum likelihood estimation to estimate the parameters of the Markov chain choice model. We also checked the performance of a regression method that estimates the parameters by minimizing the distance between the empirical purchase probabilities and the purchase probabilities predicted by the choice model, but this approach did not perform well. Finding effective parameter estimation approaches is another important research direction.

Acknowledgments and Supplementary Files. We thank the area editor, the associate editor and three anonymous referees whose comments improved the paper in many ways. The data files and the source code used in our computational experiments are available as supplementary files.

References

Bishop, C. (2006), *Pattern Recognition and Machine Learning*, Springe-Verlang, New York, NY.

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

Bront, J. J. M., Diaz, I. M. 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. (2014), 'Assortment optimization under variants of the nested logit model', *Operations Research* **62**(2), 250–273.

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. (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* **60**(10), 2583–2601.

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

Jagabathula, S. (2008), Nonparametric Choice Modeling: Applications to Operations Management, PhD thesis, Massachusetts Institute of Technology, Cambridge, MA.

Kunnumkal, S. and Talluri, K. (2012), A new compact linear programming formulation for choice network revenue management, Technical report, Universitat Pompeu Fabra, Barcelona, Spain.

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. (2014), 'A greedy algorithm for assortment optimization in the two-level nested logit model', *Operations Research Letters* **42**(5), 319–324.

Li, G., Rusmevichientong, P. and Topaloglu, H. (2015), 'The d-level nested logit model: Assortment and price optimization problems', *Operations Research* **62**(2), 325–342.

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.

Meissner, J. and Strauss, A. K. (2012), 'Network revenue management with inventory-sensitive bid prices and customer choice', *European Journal of Operational Research* **216**, 459–468.

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.

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

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. (2014), 'Assortment optimization under the multinomial logit model with random choice parameters', *Production and Operations Management* **23**(11), 2023–2039.

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.

Vossen, T. W. M. and Zhang, D. (2013), Reductions of approximate linear programs for network revenue management, Technical report, University of Colorado at Boulder, Boulder, CO.

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

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.

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

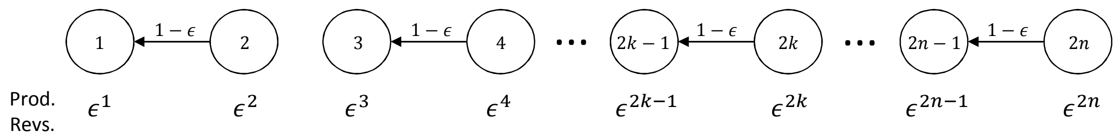
Zhang, D. and Cooper, W. L. (2005), 'Revenue management for parallel flights with customer choice behavior', *Operations Research* **53**(3), 415–431.

A Online Appendix: Poor Performance of Nested by Revenue Subsets

In this section, we show that nested by revenue subsets can perform arbitrarily poorly for the ASSORTMENT problem when customers choose according to the Markov chain choice model. In particular, we consider a problem instance with $2n$ products and show that the expected revenue provided by the best nested by revenue subset can deviate from the optimal expected revenue by a factor arbitrarily close to n. In our problem instance, we index the products by $N = \{1, \ldots, 2n\}$. For some $\epsilon \in (0, 1)$, the revenue of product j is given by ϵ^j. The probability that a customer arrives into the system to purchase product j is given by $\lambda_j = \epsilon^{2n-j+1}$. We assume that ϵ is small enough that $\sum_{j \in N} \lambda_j = \sum_{j \in N} \epsilon^{2n-j+1} \leq 1$. With probability $1 - \sum_{j \in N} \lambda_j$, there is no customer arrival. The assumption that we may not have a customer arrival is without loss of generality. In particular, if we want to ensure that there is a customer arrival with probability one, then we can scale the probabilities $(\lambda_1, \ldots, \lambda_{2n})$ with the same constant to ensure that $\sum_{j \in N} \lambda_j = 1$ without changing the optimal solution to the ASSORTMENT problem. The transition probabilities are given by $\rho_{2k,2k-1} = 1 - \epsilon$ for all $k = 1, \ldots, n$. The other transition probabilities in $\big\{\rho_{j,i} : j, i \in N\big\}$ are zero. The top portion of Table 7 shows the transition probabilities. Thus, if a customer visits product $2k$ for some $k = 1, \ldots, n$ and product $2k$ is not available, then she transitions to product $2k - 1$ with probability $1 - \epsilon$. With probability ϵ, she transitions to the no purchase option. If a customer visits product $2k - 1$ for some $k = 1, \ldots, n$ and product $2k - 1$ is not available, then she transitions to the no purchase option with probability one.

Since $\epsilon \in (0, 1)$ and $r_j = \epsilon^j$, the revenues of the products satisfy $r_1 \geq \ldots \geq r_{2n}$. Thus, a nested by revenue subset for this problem instance is of the form $\big\{1, \ldots, j\big\}$ for some $j \in N$. The first row in the bottom portion of Table 7 shows the purchase probabilities $(P_{1,S}, \ldots, P_{2n,S})$ when we offer the nested by revenue subset $S = \big\{1, \ldots, 2k\big\}$ for some $k = 1, \ldots, n$. Similarly, the second row in the bottom portion of Table 7 shows the purchase probabilities $(P_{1,S}, \ldots, P_{2n,S})$ when we offer the nested by revenue subset $S = \big\{1, \ldots, 2k - 1\big\}$ for some $k = 1, \ldots, n$. Finally, the third row in the bottom portion of Table 7 shows the purchase probabilities $(P_{1,S}, \ldots, P_{2n,S})$ when we offer the subset $S = \big\{1, 3, 5, \ldots, 2k - 1\big\}$ for some $k = 1, \ldots, n$, which is not a nested by revenue subset. All of these purchase probabilities are obtained by solving the BALANCE equations for (P_S, R_S). To give an idea, we briefly verify some of the purchase probabilities $(P_{1,S}, \ldots, P_{2n,S})$ when we offer the subset $S = \big\{1, 3, 5, \ldots, 2k - 1\big\}$. Since $1 \in S$ and $2 \notin S$, by the BALANCE equations, we have $R_{1,S} = 0$ and $P_{2,S} = 0$. Thus, the BALANCE equations imply that $P_{1,S} = \lambda_1 + \rho_{2,1} R_{2,S}$ and $R_{2,S} = \lambda_2$, in which case, we obtain $R_{2,S} = \epsilon^{2n-1}$ and $P_{1,S} = \epsilon^{2n} + (1 - \epsilon)\, \epsilon^{2n-1} = \epsilon^{2n-1}$. Similarly, since $3 \in S$ and $4 \notin S$, by the BALANCE equations, we $R_{3,S} = 0$ and $P_{4,S} = 0$. Therefore, the BALANCE equations imply that $P_{3,S} = \lambda_3 + \rho_{4,3} R_{4,S}$ and $R_{4,S} = \lambda_4$, in which case, we obtain $R_{4,S} = \epsilon^{2n-3}$ and $P_{3,S} = \epsilon^{2n-2} + (1 - \epsilon)\, \epsilon^{2n-3} = \epsilon^{2n-3}$. All of the other purchase probabilities in the bottom portion of Table 7 can be verified in a similar fashion.

Focusing on the first row in the bottom portion of Table 7, if we offer the nested by revenue subset $S = \big\{1, \ldots, 2k\big\}$ for some $k = 1, \ldots, n$, then we obtain an expected revenue of $\sum_{j \in N} P_{j,S}\, r_j =$



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

Table 7: A problem instance where the optimal expected revenue exceeds the expected revenue from the best nested by revenue subset by a factor arbitrarily close to n.

$\epsilon^{2n}\,\epsilon + \epsilon^{2n-1}\,\epsilon^2 + \ldots + \epsilon^{2n-2k+2}\,\epsilon^{2k-1} + \epsilon^{2n-2k+1}\,\epsilon^{2k} = 2k\,\epsilon^{2n+1}$. This expected revenue is increasing in k. Thus, if we want to obtain the largest expected revenue by offering a nested by revenue subset of the form $\{1,\ldots,2k\}$ for some $k = 1,\ldots,n$, then we set $k = n$, yielding an expected revenue of $2n\,\epsilon^{2n+1}$. Focusing on the second row in the bottom portion of Table 7, if we offer the nested by revenue subset $S = \{1,\ldots,2k-1\}$ for some $k = 1,\ldots,n$, then we obtain an expected revenue of $\sum_{j\in N} P_{j,S}\,r_j = \epsilon^{2n}\,\epsilon + \epsilon^{2n-1}\,\epsilon^2 + \ldots + \epsilon^{2n-2k+3}\,\epsilon^{2k-2} + \epsilon^{2n-2k+1}\,\epsilon^{2k-1} = (2k-2)\,\epsilon^{2n+1} + \epsilon^{2n}$. This expected revenue is also increasing in k. Thus, if we want to obtain the largest expected revenue by offering a nested by revenue subset of the form $\{1,\ldots,2k-1\}$ for some $k = 1,\ldots,n$, then we set $k = n$, yielding an expected revenue of $(2n-2)\,\epsilon^{2n+1} + \epsilon^{2n}$. Putting the results in this paragraph together, if we offer the nested by revenue subset $\{1,\ldots,2k\}$ for some $k = 1,\ldots,n$, then the largest expected revenue is $2n\,\epsilon^{2n+1}$. If we offer the nested by revenue subset $\{1,\ldots,2k-1\}$ for some $k = 1,\ldots,n$, then the largest expected revenue is $(2n-2)\,\epsilon^{2n+1} + \epsilon^{2n}$. Thus, the expected revenue from any nested by revenue subset is no larger than $2n\,\epsilon^{2n+1} + \epsilon^{2n}$.

Focusing on the third row in the bottom portion of Table 7, if we offer the subset $S = \{1,3,5,\ldots,2k-1\}$ for some $k = 1,\ldots,n$, then we obtain an expected revenue of $\sum_{j\in N} P_{j,S}\,r_j = \epsilon^{2n-1}\,\epsilon + \epsilon^{2n-3}\,\epsilon^3 + \ldots + \epsilon^{2n-2k+1}\,\epsilon^{2k-1} = k\,\epsilon^{2n}$. This expected revenue is also increasing in k. Thus, if we want to obtain the largest expected revenue by offering a subset of the form $\{1,3,5,\ldots,2k-1\}$ for some $k = 1,\ldots,n$, then we set $k = n$, yielding an expected revenue of $n\,\epsilon^{2n}$. At the end of the previous paragraph, we observe that the expected revenue from any nested by revenue subset is no larger than $2n\,\epsilon^{2n+1} + \epsilon^{2n}$, but the expected revenue from the subset $\{1,3,5,\ldots,2n-1\}$ is $n\,\epsilon^{2n}$. Therefore, the ratio between the optimal expected revenue and the expected revenue from the best nested by revenue subset is at least $n\,\epsilon^{2n}/(2n\,\epsilon^{2n+1} + \epsilon^{2n}) = n/(2n\,\epsilon + 1)$. Since we have $\lim_{\epsilon\to 0} n/(2n\,\epsilon + 1) = n$, by choosing ϵ sufficiently small, we obtain a problem instance, where the optimal expected revenue exceeds the expected revenue from the best nested by revenue subset by a factor arbitrarily close to n. Thus, as the number of products gets larger, n gets larger and nested by revenue subsets perform arbitrarily poorly for this problem instance.

B Online Appendix: Optimality of Nested by Revenue Subsets

Below is the proof of Lemma 4.

Proof of Lemma 4. Without loss of generality, we assume that the revenues of the products are positive, since we can drop the products with negative revenues without changing the optimal solution to the ASSORTMENT problem. Letting $\hat{v}$ be the optimal solution to the DUAL problem, since $\hat{v}$ is a feasible solution to the DUAL problem, $\hat{v}_j \geq r_j \geq 0$ for all $j \in N$. Defining the subset $\hat{S} = \{j \in N : \hat{v}_j = r_j\}$, by Theorem 2, $\hat{S}$ is the optimal solution to the ASSORTMENT problem. To get a contradiction, assume that $\hat{S}$ is not a nested by revenue subset so that $j_1 \notin \hat{S}$ and $j_2 \in \hat{S}$ for some $j_1, j_2 \in N$ with $j_1 < j_2$. Since $\hat{v}$ is a feasible solution to the DUAL problem, we have $v_{j_1} \geq r_{j_1}$ and $v_{j_1} \geq \sum_{i \in N} \rho_{j_1,i}\, \hat{v}_i$. By the same argument in the proof of Theorem 2, we have $\hat{v}_j = r_j$ or $\hat{v}_j = \sum_{i \in N} \rho_{j,i}\, \hat{v}_i$ for all $j \in N$. Thus, since $j_1 \notin \hat{S}$, we have $\hat{v}_{j_1} \neq r_{j_1}$, which implies that $\sum_{i \in N} \rho_{j_1,i}\, \hat{v}_i = \hat{v}_{j_1} > r_{j_1}$. Furthermore, we have $r_{j_2} = v_{j_2} \geq \sum_{i \in N} \rho_{j_2,i}\, \hat{v}_i$, where the equality uses the fact that $j_2 \in \hat{S}$ and the inequality uses the fact that $\hat{v}$ is a feasible solution to the DUAL problem. Noting that $j_1 < j_2$ and the products are indexed such that $r_1 \geq \ldots \geq r_n$, it follows that $\sum_{i \in N} \rho_{j_1,i}\, \hat{v}_i = \hat{v}_{j_1} > r_{j_1} \geq r_{j_2} = v_{j_2} \geq \sum_{i \in N} \rho_{j_2,i} \hat{v}_i$, indicating that $\sum_{i \in N} \rho_{j_1,i}\, \hat{v}_i > \sum_{i \in N} \rho_{j_2,i}\, \hat{v}_i$. However, since $j_1 < j_2$, by the assumption in the lemma, we have $\rho_{j_1,i} \leq \rho_{j_2,i}$ for all $i \in N$. Therefore, we cannot have $\sum_{i \in N} \rho_{j_1,i}\, \hat{v}_i > \sum_{i \in N} \rho_{j_2,i}\, \hat{v}_i$ and we reach a contradiction. □

C Online Appendix: Nesting Order of Optimal Subsets

Below is the proof of Lemma 6, which follows as a corollary to Lemma 4 and Theorem 5.

Proof of Lemma 6. We observe that $\hat{S}_t(x)$ is given by the optimal solution to the problem $\max_{S \subset N} \sum_{j \in N} P_{j,S}\,(r_j + V_{t+1}(x-1) - V_{t+1}(x))$, which has the same form as the ASSORTMENT problem with the revenue of product j given by $r_j + V_{t+1}(x-1) - V_{t+1}(x)$. Since $r_1 \geq \ldots \geq r_n$, we have $r_1 + V_{t+1}(x-1) - V_{t+1}(x) \geq \ldots \geq r_n + V_{t+1}(x-1) - V_{t+1}(x)$. In this case, by Lemma 4, the optimal solution to the problem $\max_{S \subset N} \sum_{j \in N} P_{j,S}\,(r_j + V_{t+1}(x-1) - V_{t+1}(x))$ is a nested by revenue subset of the form $\{1, \ldots, j_t(x)\}$ for some $j_t(x) \in N \cup \{0\}$. Thus, $\hat{S}_t(x) = \{1, \ldots, j_t(x)\}$ for some $j_t(x) \in N \cup \{0\}$. By Theorem 5, we have $\hat{S}_t(x-1) \subset \hat{S}_t(x)$ and $\hat{S}_{t-1}(x) \subset \hat{S}_t(x)$, which implies that $j_t(x)$ is increasing in x and increasing in t. □

D Online Appendix: Boundedness of the Feasible Set

In the next lemma, we show that the polyhedron $\mathcal{H}$ is bounded.

Lemma 10 *There exists $U < \infty$ such that if $(x, z) \in \mathcal{H}$, then $x_j \leq U$ and $z_j \leq U$ for all $j \in N$.*

Proof. We define the polyhedron $\mathcal{G} = \{z \in \Re^n_+ : z_j \leq \lambda_j + \sum_{i \in N} \rho_{i,j}\, z_i \;\forall\, j \in N\}$. We claim that there exists $L < \infty$ such that if $z \in \mathcal{G}$, then $z_j \leq L$ for all $j \in N$. Using the matrix $Q =$

$\{\rho_{j,i} : j, i \in N\}$ and the vector $\lambda = (\lambda_1, \ldots, \lambda_n)$, by the definition of $\mathcal{G}$, if $z \in \mathcal{G}$, then we have $z \leq \lambda + Q^\top z$, which can be written as $(I - Q)^\top z \leq \lambda$, where I is the identity matrix with the appropriate dimension. By the same argument used in Section 1, since $\sum_{i \in N} \rho_{j,i} < 1$ for all $j \in N$, $(I - Q)^{-1}$ exists and has nonnegative entries. In this case, multiplying both sides of the inequality $(I - Q)^\top z \leq \lambda$ with $((I - Q)^{-1})^\top$ and noting that $((I - Q)^{-1})^\top = ((I - Q)^\top)^{-1}$, it follows that if $z \in \mathcal{G}$, then we have $z \leq ((I - Q)^{-1})^\top \lambda$. Thus, letting L be the largest entry of the matrix $(I - Q)^{-1}$, since $\sum_{j \in N} \lambda_j \leq 1$, the last inequality implies that if $z \in \mathcal{G}$, then we have $z_j \leq L$ for all $j \in N$, establishing the claim. By the definition of $\mathcal{H}$, if $(x, z) \in \mathcal{H}$, then we have

$$z_j \leq \lambda_j + \sum_{i \in N} \rho_{i,j}\, z_i \quad \text{and} \quad x_j \leq \lambda_j + \sum_{i \in N} \rho_{i,j}\, z_i$$

for all $j \in N$. By the first inequality above, it follows that if $(x, z) \in \mathcal{H}$, then $z \in \mathcal{G}$, in which case, we have $z_j \leq L$ for all $j \in N$. In this case, by the second inequality, it follows that if $(x, z) \in \mathcal{H}$, then $x_j \leq \lambda_j + \sum_{i \in N} \rho_{i,j}\, z_i \leq \lambda_j + n\, L \leq 1 + n\, L$. Therefore, letting $U = 1 + n\, L$, if $(x, z) \in \mathcal{H}$, then we have $x_j \leq U$ and $z_j \leq U$ for all $j \in N$. ☐

E Online Appendix: Comparison of Expected Number of Visits to a Product

For $(\hat{x}, \hat{z}) \in \mathcal{H}$, we define $S_{\hat{x}} = \{j \in N : \hat{x}_j > 0\}$. In the next lemma, we show that $\hat{z}_j \geq R_{j, S_{\hat{x}}}$ for all $j \in N$. This result is used in several places throughout Section 6.

Lemma 11 *For $(\hat{x}, \hat{z}) \in \mathcal{H}$, define $S_{\hat{x}} = \{j \in N : \hat{x}_j > 0\}$. Then, we have $\hat{z}_j \geq R_{j, S_{\hat{x}}}$ for all $j \in N$.*

Proof. Since $(\hat{x}, \hat{z}) \in \mathcal{H}$ and $\hat{x}_j = 0$ for all $j \notin S_{\hat{x}}$, we have $\hat{z}_j = \lambda_j + \sum_{i \in N} \rho_{i,j}\, \hat{z}_i$ for all $j \notin S_{\hat{x}}$. Also, since $P_{j, S_{\hat{x}}} = 0$ for all $j \notin S_{\hat{x}}$ and $R_{j, S_{\hat{x}}} = 0$ for all $j \in S_{\hat{x}}$, by the Balance equations, we have $R_{j, S_{\hat{x}}} = \lambda_j + \sum_{i \notin S_{\hat{x}}} \rho_{i,j}\, R_{i, S_{\hat{x}}}$ for all $j \notin S_{\hat{x}}$. Subtracting the last two equalities, we obtain $\hat{z}_j - R_{j, S_{\hat{x}}} = \sum_{i \in S_{\hat{x}}} \rho_{i,j}\, \hat{z}_i + \sum_{i \notin S_{\hat{x}}} \rho_{i,j}\, (\hat{z}_i - R_{i, S_{\hat{x}}})$, which implies that $\hat{z}_j - R_{j, S_{\hat{x}}} \geq \sum_{i \notin S_{\hat{x}}} \rho_{i,j}\, (\hat{z}_i - R_{i, S_{\hat{x}}})$ for all $j \notin S_{\hat{x}}$. We let $\eta_j = \hat{z}_j - R_{j, S_{\hat{x}}}$ for all $j \notin S_{\hat{x}}$. Using the matrix $\bar{Q} = \{\rho_{j,i} : j, i \notin S_{\hat{x}}\}$ and the vector $\bar{\eta} = \{\eta_j : j \notin S_{\hat{x}}\}$, we can write the last inequality as $(I - \bar{Q})^\top \bar{\eta} \geq 0$. By the same argument in Section 1, $(I - \bar{Q})^{-1}$ exists and has nonnegative entries. So, noting that $((I - \bar{Q})^\top)^{-1} = ((I - \bar{Q})^{-1})^\top$ and multiplying the last inequality with $((I - \bar{Q})^\top)^{-1}$, we obtain $\bar{\eta} \geq 0$. Thus, we have $\eta_j = \hat{z}_j - R_{j, S_{\hat{x}}} \geq 0$ for all $j \notin S_{\hat{x}}$. Also, we have $\hat{z}_j \geq 0 = R_{j, S_{\hat{x}}}$ for all $j \in S_{\hat{x}}$, where the inequality uses the fact that $(\hat{x}, \hat{z}) \in \mathcal{H}$. Thus, we obtain $\hat{z}_j \geq R_{j, S_{\hat{x}}}$ for all $j \in N$. ☐

F Online Appendix: Geometric Interpretation of Dimension Reduction

In this section, we give a geometric interpretation for the Dimension Reduction algorithm. We recall that Lemma 1 gives a connection between (P_S, R_S) and the extreme points of $\mathcal{H}$. In particular, for any extreme point $(\hat{x}, \hat{z})$ of $\mathcal{H}$, if we let $S_{\hat{x}} = \{j \in N : \hat{x}_j > 0\}$, then we have $P_{S_{\hat{x}}} = \hat{x}$ and $R_{S_{\hat{x}}} = \hat{z}$. Treating $\{x_j : j \in N\}$ as the slack variables, we write $\mathcal{H}$ equivalently

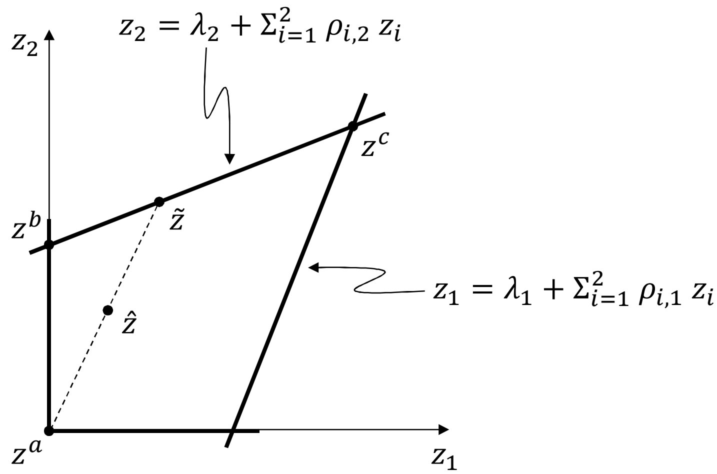


Figure 1: The polyhedron $\mathcal{G} = \{z \in \Re^n_+ : z_j \leq \lambda_j + \sum_{i \in N} \rho_{i,j}\, z_i \ \forall\, j \in N\}$ for the case where $n = 2$.

as $\mathcal{G} = \{z \in \Re^n_+ : z_j \leq \lambda_j + \sum_{i \in N} \rho_{i,j}\, z_i \ \forall\, j \in N\}$. The polyhedra $\mathcal{H}$ and $\mathcal{G}$ are equivalent in the sense that for any $z \in \mathcal{G}$, if we define $x \in \Re^n_+$ as $x_j = \lambda_j + \sum_{i \in N} \rho_{i,j}\, z_i - z_j$ for all $j \in N$, then we have $(x, z) \in \mathcal{H}$. To give an interpretation for the Dimension Reduction algorithm, we focus on the case where $n = 2$ so that $N = \{1, 2\}$. For the case where $n = 2$, we show the polyhedron $\mathcal{G}$ in Figure 1. For the extreme point $z^a = (z^a_1, z^a_2)$ of $\mathcal{G}$ in Figure 1, we have $z^a_1 = z^a_2 = 0$. Thus, we obtain $x^a_j = \lambda_j + \sum_{i \in N} \rho_{i,j}\, z^a_i - z^a_j > 0$ for all $j = 1, 2$, where we use the fact that $\lambda_j > 0$ and $z^a_j = 0$ for all $j = 1, 2$. In this case, defining the subset $S^a = \{j \in N : x^a_j > 0\}$, we have $S^a = \{j \in N : x^a_j > 0\} = \{1, 2\}$. Since z^a is an extreme point of $\mathcal{G}$, one can check that (x^a, z^a) is an extreme point of $\mathcal{H}$, in which case, Lemma 1 implies that $P_{S^a} = x^a$ and $R_{S^a} = z^a$. Similarly, for the extreme point $z^b = (z^b_1, z^b_2)$ in Figure 1, we have $z^b_1 = 0$ and $z^b_2 = \lambda_2 + \sum_{i \in N} \rho_{i,2}\, z^b_i$. So, we obtain $x^b_1 = \lambda_1 + \sum_{i \in N} \rho_{i,1}\, z^b_i - z^b_1 > 0$ and $x^b_2 = \lambda_2 + \sum_{i \in N} \rho_{i,2}\, z^b_i - z^b_2 = 0$. In this case, defining the subset $S^b = \{j \in N : x^b_j > 0\}$, we have $S^b = \{j \in N : x^b_j > 0\} = \{1\}$ and Lemma 1 implies that $P_{S^b} = x^b$ and $R_{S^b} = z^b$. Finally, for the extreme point $z^c = (z^c_1, z^c_2)$ in Figure 1, we have $z^c_j = \lambda_j + \sum_{i \in N} \rho_{i,j}\, z^c_i$ for all $j = 1, 2$. Therefore, we obtain $x^c_j = \lambda_j + \sum_{i \in N} \rho_{i,j}\, z^c_i - z^c_j = 0$ for all $j = 1, 2$. In this case, if we define the subset $S^c = \{j \in N : x^c_j > 0\}$, then we have $S^c = \{j \in N : x^c_j > 0\} = \varnothing$ and Lemma 1 implies that $P_{S^c} = x^c$ and $R_{S^c} = z^c$.

The discussion above shows that $S^a = \{1, 2\} \supset S^b = \{1\} \supset S^c = \varnothing$. Furthermore, we have $R_{S^a} = z^a$, $R_{S^b} = z^b$ and $R_{S^c} = z^c$. Letting $(\hat{x}, \hat{z})$ be the optimal solution to the Reduced linear program, the second constraint in this problem implies that $\hat{z} \in \mathcal{G}$. We show the point $\hat{z}$ in Figure 1. We observe that $\hat{z}$ can be expressed as $\hat{z} = \alpha^a\, z^a + (1 - \alpha^a)\, \tilde{z}$ for some $\alpha^a \in [0, 1]$ and $\tilde{z} \in \mathcal{G}$, where the point $\tilde{z}$ is as shown in Figure 1. On the other hand, $\tilde{z}$ can be expressed as $\tilde{z} = \alpha^b\, z^b + (1 - \alpha^b)\, z^c$ for some $\alpha^b \in [0, 1]$. Therefore, $\hat{z}$ can be expressed as $\hat{z} = \alpha^a\, z^a + (1 - \alpha^a)\, (\alpha^b\, z^b + (1 - \alpha^b)\, z^c) = \alpha^a\, R_{S^a} + (1 - \alpha^a)\, \alpha^b\, R_{S^b} + (1 - \alpha^a)\, (1 - \alpha^b)\, R_{S^c}$. Thus, letting $\gamma^a = \alpha^a$, $\gamma^b = (1 - \alpha^a)\, \alpha^b$ and $\gamma^c = (1 - \alpha^a)\, (1 - \alpha^b)$, it follows that $\gamma^a + \gamma^b + \gamma^c = 1$ and $\hat{z} = \gamma^a\, R_{S^a} + \gamma^b\, R_{S^b} + \gamma^c\, R_{S^c}$, indicating that $\hat{z}$ can be expressed as a convex combination of R_{S^a}, R_{S^b} and R_{S^c}. To sum up, by connecting the points z^a and $\hat{z}$ through a line segment and finding the intersection point of this line segment with a face of $\mathcal{G}$, we can identify the point $\tilde{z}$ so that we can express $\hat{z}$ as a convex combination of z^a and $\tilde{z}$. Since $S^a = \{1,2\}$, expressing $\hat{z}$ as a convex combination of z^a and $\tilde{z}$ implies that the optimal solution to the Reduced linear program offers the subset $\{1,2\}$, along with some other subsets. From this point on, we focus on a smaller dimensional space characterized by only one face of $\mathcal{G}$ that is given by $\lambda_2 + \sum_{i \in N} \rho_{i,2}\, z_i - z_2 = 0$. Since we have $x_2 = \lambda_2 + \sum_{i \in N} \rho_{i,2}\, z_i - z_2 = 0$ on this face, we can drop product 2 from consideration and only focus on offering other subsets that do not include product 2. Following a similar argument, we can show that if $(\hat{x}, \hat{z})$ is the optimal solution to the Reduced linear program, then $\hat{x}$ can also be expressed as a convex combination of P_{S^a}, P_{S^b} and P_{S^c} as well. The Dimension Reduction algorithm follows the same approach to express $(\hat{x}, \hat{z})$ as a convex combination of the extreme points of $\mathcal{H}$. If we let $j^k = \arg\min\{x^k_j / P_{j,S^k} : j \in S^k\}$ in Step 2 of the Dimension Reduction algorithm, then we drop product j^k from consideration at iteration k and focus on offering other subsets that do not include product j^k. In this way, the Dimension Reduction algorithm finds nested subsets $S^1 \supset \ldots \supset S^K$ and positive scalars $\gamma^1, \ldots, \gamma^K$ summing to one such that the optimal solution $(\hat{x}, \hat{z})$ to the Reduced linear program can be expressed as $\hat{x} = \sum_{k=1}^K \gamma^k\, P_{S^k}$ and $\hat{z} = \sum_{k=1}^K \gamma^k\, R_{S^k}$.

G Online Appendix: Derivatives of Purchase Probabilities

In this section, we discuss how we can compute the derivatives of the purchase probabilities with respect to the parameters of the Markov chain choice model. We use $\{P_{j,S}(\lambda,\rho) : j \in N\}$ and $\{R_{j,S}(\lambda,\rho) : j \in N\}$ to denote the solution to the Balance equations, where we make the dependence of the solution on the parameters of the Markov chain choice model explicit. We focus on computing $\partial P_{j,S}(\lambda,\rho)/\partial \rho_{k,\ell}$ and $\partial R_{j,S}(\lambda,\rho)/\partial \rho_{k,\ell}$ for all $j,k,\ell \in N$. By using a similar approach, we can also compute $\partial P_{j,S}(\lambda,\rho)/\partial \lambda_k$ and $\partial R_{j,S}(\lambda,\rho)/\partial \lambda_k$ for all $j,k \in N$. Since $P_{j,S}(\lambda,\rho) = 0$ for all $j \notin S$ and $R_{j,S}(\lambda,\rho) = 0$ for all $j \in S$, we have $\partial P_{j,S}(\lambda,\rho)/\partial \rho_{k,\ell} = 0$ for all $j \notin S$ and $\partial R_{j,S}(\lambda,\rho)/\partial \rho_{k,\ell} = 0$ for all $j \in S$. For $j \notin S$, we write the Balance equations as $R_{j,S}(\lambda,\rho) = \lambda_j + \sum_{i \notin S} \rho_{i,j} R_{i,S}(\lambda,\rho)$. Differentiating this expression with respect to $\rho_{k,\ell}$, we get

$$\frac{\partial R_{j,S}(\lambda,\rho)}{\partial \rho_{k,\ell}} = \mathbf{1}(k \notin S, \ell = j)\, R_{k,S} + \sum_{i \notin S} \rho_{i,j}\, \frac{\partial R_{i,S}(\lambda,\rho)}{\partial \rho_{k,\ell}} \quad \forall\, j \notin S.$$

There are $n - |S|$ equations in the system of equation above. We can solve this system of equations for the $n - |S|$ unknowns $\{\partial R_{j,S}(\lambda,\rho)/\partial \rho_{k,\ell} : j \notin S\}$ to compute $\partial R_{j,S}(\lambda,\rho)/\partial \rho_{k,\ell}$ for all $j \notin S$. Furthermore, for $j \in S$, we write the Balance equations as $P_{j,S}(\lambda,\rho) = \lambda_j + \sum_{i \notin S} \rho_{i,j}\, R_{i,S}(\lambda,\rho)$. Differentiating this expression with respect to $\rho_{k,\ell}$, we obtain

$$\frac{\partial P_{j,S}(\lambda,\rho)}{\partial \rho_{k,\ell}} = \mathbf{1}(k \notin S, \ell = j)\, R_{k,S} + \sum_{i \notin S} \rho_{i,j}\, \frac{\partial R_{i,S}(\lambda,\rho)}{\partial \rho_{k,\ell}} \quad \forall\, j \in S.$$

Once we compute $\{\partial R_{j,S}(\lambda,\rho)/\partial \rho_{k,\ell} : j \notin S\}$, we can use these values on the right side above to compute $\{\partial P_{j,S}(\lambda,\rho)/\partial \rho_{k,\ell} : j \in S\}$. Since $\sum_{i \in N} \rho_{j,i} < 1$ for all $j \in N$, we can use the same argument in Section 1 to show that there exists a unique solution to the first system of equations above. This solution provides the derivatives $\{\partial R_{j,S}(\lambda, \rho)/\partial \rho_{k,\ell} : j \notin S\}$.

H Online Appendix: Robustness of Predicting Customer Purchases

To make sure that our results do not change considerably when we use different training data sets to fit the choice models, we repeated the computational experiments in Section 7.3 for 10 different training data sets that are generated from the same ground choice model. In this way, we check how much the fitted choice models change from one training data set to another. In Section 7.3, we have 10 different ground choice models. For each ground choice model, we generate 10 different training data sets. To each training data set, we fit a Markov chain choice model and a multinomial logit model. We check the out of sample log likelihoods of these fitted choice models by using a testing data set. For economy of space, we only provide the percent gaps between the out of sample log likelihoods of the fitted Markov chain choice model and the multinomial logit model. This performance measure is identical to the one in the third column of each block in Table 3. Table 8 shows our results. The three portions of this table correspond to the cases where we have $\tau = 1000$, $\tau = 1750$ and $\tau = 2500$ data points in the training data set, corresponding to three levels of data availability. Each row corresponds to a different ground choice model. Each column corresponds to a different training data set. The first training data set in Table 8 always corresponds to the one that is used in Section 7.3. Thus, the entries in the first columns in each of the three portions of Table 8 are identical to the entries in the third column in each of the three blocks in Table 3. The results in Table 8 indicate that the fitted Markov chain choice model improves the out of sample log likelihoods of the fitted multinomial logit model for the levels of data availability and the numbers of parameters to be estimated in our test problems.

I Online Appendix: Robustness of Finding Good Assortments

To check how the expected revenues obtained by using the two fitted choice models change from one training data set to another, we repeated the computational experiments in Section 7.4 for 10 different training data sets that are generated from the same ground choice model. Our experimental setup is similar to the one in in Section 7.4 and Online Appendix H. In Section 7.4, we have 10 different ground choice models. For each ground choice model, we generate 10 different training data sets. To each training data set, we fit a Markov chain choice model and a multinomial logit model. To check the expected revenue obtained by using the two fitted choice models, we sample 100 different sets of product revenues. Using $\{(r_1^k, \ldots, r_n^k) : k = 1, \ldots, 100\}$ to denote these samples, we solve the Assortment problem under the assumption that the revenues of the products are $(r_1^k, \ldots, r_n^k)$ and the customers choose according to the fitted Markov chain choice model. Similarly, we solve the Assortment problem under the assumption that the revenues of the products are $(r_1^k, \ldots, r_n^k)$ and the customers choose according to the fitted multinomial logit model. This approach provides two solutions to the Assortment problem. The first one is

$\tau = 1000$

Grn. Cho. Mod.	Training Data Set 1	2	3	4	5	6	7	8	9	10
1	2.40	3.40	2.47	3.12	2.68	2.94	3.41	2.33	3.94	3.03
2	2.45	1.38	1.93	2.53	1.39	2.38	1.85	2.67	1.67	0.97
3	1.99	2.27	2.67	2.06	3.02	3.40	3.44	2.56	2.50	0.79
4	2.98	3.13	2.50	3.53	3.18	2.38	3.42	2.46	2.57	2.44
5	2.55	3.86	1.15	3.27	2.85	2.98	2.35	2.37	3.14	3.71
6	2.54	2.86	2.72	2.47	1.80	3.18	1.80	2.41	1.42	2.60
7	2.94	2.57	2.46	2.49	3.19	2.81	2.10	2.97	3.22	2.22
8	2.96	2.73	2.33	1.53	2.39	2.45	3.48	2.82	2.88	2.38
9	3.91	2.39	3.85	3.99	3.49	2.02	2.41	3.00	3.29	3.30
10	1.77	2.44	0.58	3.36	1.66	2.07	2.53	2.32	2.53	1.91

$\tau = 1750$

Grn. Cho. Mod.	Training Data Set 1	2	3	4	5	6	7	8	9	10
1	3.65	3.92	3.56	4.28	3.02	3.44	3.70	3.30	3.78	3.21
2	3.52	2.95	3.00	2.65	2.21	2.89	2.63	3.23	2.42	2.21
3	2.48	2.48	2.85	2.85	2.84	3.63	3.88	3.05	2.37	2.71
4	3.23	3.91	3.44	4.01	3.57	2.86	3.54	3.85	4.08	3.10
5	2.79	4.67	3.43	3.65	3.35	3.32	3.42	2.97	3.70	3.87
6	3.10	2.98	2.72	2.94	2.58	3.27	2.52	3.71	3.45	3.09
7	2.68	3.56	3.10	3.33	3.45	2.88	3.03	3.64	3.73	3.00
8	2.78	2.89	3.15	2.31	2.90	3.14	3.87	3.11	3.04	3.04
9	3.72	3.26	3.84	3.80	4.02	3.65	3.05	4.35	3.95	3.64
10	2.18	2.93	2.15	3.96	2.49	2.21	3.06	3.27	3.13	2.65

$\tau = 2500$

Grn. Cho. Mod.	Training Data Set 1	2	3	4	5	6	7	8	9	10
1	3.98	3.89	3.98	4.43	3.05	3.82	3.83	3.37	4.01	3.40
2	3.61	3.03	3.15	2.92	2.54	2.96	2.99	3.69	2.82	2.78
3	3.01	2.76	3.09	2.74	3.14	3.72	3.87	3.41	2.39	3.14
4	3.43	4.08	4.05	4.28	3.86	2.92	3.67	3.66	4.05	3.71
5	3.18	4.75	3.60	3.82	3.59	3.98	3.74	3.13	3.90	3.84
6	3.52	2.75	2.92	2.82	2.70	3.74	2.94	3.78	3.91	2.92
7	3.51	3.47	3.16	3.93	3.51	2.83	3.24	3.84	3.91	3.24
8	3.08	3.26	3.29	2.27	3.31	3.51	3.81	3.27	3.26	3.36
9	4.30	3.20	4.26	4.24	4.31	3.92	3.36	4.59	4.24	4.12
10	2.76	3.29	2.31	4.04	2.49	2.33	3.10	3.38	3.23	3.31

Table 8: Percent gaps in the log likelihoods of the testing data set under the two fitted choice models.

computed under the assumption that the customers choose according to the fitted Markov chain choice model, whereas the second one is computed under the assumption that the customers choose according to the fitted multinomial logit model. We compute the actual expected revenues that are obtained by these two assortments when the customers choose according to the ground choice model that actually governs their choice process. We only provide the average percent gap between the two actual expected revenues, where the average is computed over all 100 samples of the product revenues $\{(r_1^k, \ldots, r_n^k) : k = 1, \ldots, 100\}$. This performance measure is identical to the one in the third column of each block in Table 4. Table 9 shows our results. The three portions of the table correspond to the cases where we have $\tau = 1000$, $\tau = 1750$ and $\tau = 2500$ data points in the training data set. Each row corresponds to a different ground choice model. Each column corresponds to a different training data set. The first training data set in Table 9 always corresponds to the one that is used in Section 7.4. Thus, the entries in the first columns in each of the three portions of Table 9 are identical to the entries in the third column in each of the three blocks in Table 4. In Table 9, for the levels of data availability and the numbers of parameters to be estimated in our test problems, the expected revenues obtained by using the fitted Markov chain choice model are larger than those obtained by using the fitted multinomial logit model.

J Online Appendix: Performance of Column Generation

As discussed in Section 8.2, column generation approaches can be fast in obtaining a near optimal solution, but they can be slow in ultimately reaching the optimal solution. To ensure that this concern is not a significant factor in our computational experiments, Table 10 shows the CPU seconds for CG to obtain a solution that provides an objective value within 1% of the optimal objective value of the Choice Based linear program. Similar to Table 5, the first column in this table shows the problem instances. The second column shows the CPU seconds for CG to obtain a solution within 1% of the optimal objective value. For comparison purposes, the third column shows the CPU seconds for DR to obtain the optimal solution to the Choice Based linear program. Thus, the entries in the third column are identical to those in the third column of Table 5. The fourth column in Table 10 shows the ratios of the CPU seconds in the second and third columns. The results indicate that DR continues to provide substantial improvements over CG even when we terminate CG once it obtains a solution within 1% of the optimal objective value. There are problem instances, where DR improves the CPU seconds for CG by factors exceeding 1000. In Table 10, DR improves the performance of CG by an average factor of 286. The improvements provided by DR become more pronounced as the size of the problem, measured by the numbers of resources and products, increases, or as the capacities on the flight legs become tighter.

K Online Appendix: Zero Arrival Probabilities

Throughout the paper, we assume that $\lambda_j > 0$ for all $j \in N$. We use this assumption in the proofs of Lemma 1, Theorem 2 and Lemma 3. Having $\lambda_j > 0$ for all $j \in N$ allows us to avoid some

$\tau = 1000$

Grn. Cho. Mod.	Training Data Set 1	2	3	4	5	6	7	8	9	10
1	7.37	8.83	6.69	6.68	7.54	6.88	6.89	5.27	5.21	9.73
2	5.87	6.70	4.97	5.97	9.79	6.93	3.92	3.96	4.89	6.08
3	3.82	3.02	6.26	5.81	5.03	4.10	5.95	4.43	5.53	6.17
4	5.26	7.43	5.00	4.24	7.14	6.12	9.15	7.86	5.76	4.48
5	7.32	5.44	6.86	5.77	6.99	5.34	5.85	5.77	7.15	4.91
6	5.96	5.49	5.23	6.62	7.42	7.07	6.25	6.51	6.34	4.12
7	5.31	5.18	7.28	7.02	7.25	5.50	5.73	4.91	7.72	8.53
8	8.50	8.20	3.98	5.53	7.74	4.05	9.49	6.48	9.10	6.16
9	7.29	5.49	8.39	5.83	7.00	7.01	8.03	6.79	5.50	5.02
10	7.19	7.33	1.79	8.60	6.21	6.75	5.73	4.51	3.64	3.93

$\tau = 1750$

Grn. Cho. Mod.	Training Data Set 1	2	3	4	5	6	7	8	9	10
1	7.51	9.31	8.02	7.82	7.05	7.92	6.97	5.14	6.00	9.45
2	6.06	8.87	5.67	6.26	9.68	7.19	6.34	4.58	5.64	6.41
3	3.42	4.69	6.00	6.31	6.92	4.56	6.70	6.12	5.29	7.55
4	5.96	6.66	5.76	4.57	7.00	6.93	9.63	9.35	6.10	4.96
5	7.77	5.62	7.93	6.08	6.87	6.21	6.31	6.14	7.68	6.01
6	6.49	5.89	6.01	6.87	7.72	6.90	6.93	6.73	6.16	4.90
7	6.52	6.58	7.38	7.55	8.39	5.40	6.45	5.81	8.09	7.86
8	9.09	9.11	4.81	7.98	8.10	5.36	9.53	7.08	8.64	7.98
9	7.90	6.20	8.54	5.88	7.71	7.40	8.91	7.69	6.51	6.31
10	7.54	6.66	5.45	8.13	6.70	6.67	5.11	4.40	4.28	4.82

$\tau = 2500$

Grn. Cho. Mod.	Training Data Set 1	2	3	4	5	6	7	8	9	10
1	7.89	9.42	7.12	8.09	7.6	7.85	7.3	5.51	6.24	9.25
2	6.43	8.65	5.48	7.62	9.37	7.69	6.84	5.34	6.14	6.52
3	3.93	5.12	6.20	5.87	7.20	5.30	6.79	5.77	5.37	7.79
4	5.96	6.94	5.87	5.26	7.02	7.02	8.74	8.55	5.96	5.17
5	8.59	5.82	8.27	6.39	7.68	6.49	6.87	7.08	7.79	5.97
6	7.40	5.54	6.05	6.66	8.16	7.48	7.37	6.74	7.19	3.38
7	7.47	6.62	7.73	6.77	8.48	5.28	6.08	6.94	8.52	7.78
8	8.64	8.34	5.22	8.21	8.05	5.69	9.53	7.70	8.71	7.76
9	8.32	5.96	9.37	5.64	8.34	8.35	8.60	7.86	6.92	7.16
10	7.58	6.77	6.02	8.29	7.29	6.54	5.43	5.18	4.54	5.24

Table 9: Percent gaps in the expected revenues obtained under the assumption that the customers choose according to the two fitted choice models.

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

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

Table 10: CPU seconds for CG and DR when we terminate CG once it obtains a solution within 1% of the optimal objective value.

degenerate cases in our proofs, but Lemma 1, Theorem 2 and Lemma 3 continue to hold with some modifications when we have $\lambda_j = 0$ for some $j \in N$. In this section, we show how to extend our results to the case where we have $\lambda_j = 0$ for some $j \in N$.

Extension of Lemma 1. We proceed to discussing the modifications needed in the proof of Lemma 1 when we have $\lambda_j = 0$ for some $j \in N$. Letting $(\hat{x}, \hat{z}) \in \Re^{2n}_+$ be an extreme point of $\mathcal{H}$, define $S_{\hat{x}} = \{j \in N : \hat{x}_j > 0\}$. We want to show that $P_{j,S_{\hat{x}}} = \hat{x}_j$ and $R_{j,S_{\hat{x}}} = \hat{z}_j$ for all $j \in N$. Considering the extreme point $(\hat{x}, \hat{z}) \in \Re^{2n}_+$, we define the set of products $M = \{j \in N : \hat{x}_j + \hat{z}_j > 0\}$. Noting that $\hat{x} \geq 0$ and $\hat{z} \geq 0$, we have $\hat{x}_j = 0$ and $\hat{z}_j = 0$ for all $j \in N \setminus M$. Letting $m = |M|$ for notational brevity, we define the polyhedron

$$\mathcal{F} = \left\{ (y, w) \in \Re^{2m}_+ \ : \ y_j + w_j = \lambda_j + \sum_{i \in M} \rho_{i,j}\, w_i \quad \forall\, j \in M \right\},$$

where we use the vectors $y = \{y_j : j \in M\}$ and $w = \{w_j : j \in M\}$. For the extreme point $(\hat{x}, \hat{z}) \in \Re^{2n}_+$, we define the vector $(\hat{y}, \hat{w}) \in \Re^{2m}_+$ such that $\hat{y}_j = \hat{x}_j$ and $\hat{w}_j = \hat{z}_j$ for all $j \in M$. Thus, the vectors $\hat{y}$ and $\hat{w}$ are obtained by dropping the zero components in the vectors $\hat{x}$ and $\hat{z}$ corresponding to the products in $N \setminus M$. For all $j \in M$, we have $\hat{y}_j + \hat{w}_j = \hat{x}_j + \hat{z}_j = \lambda_j + \sum_{i \in N} \rho_{i,j}\, \hat{z}_i = \lambda_j + \sum_{i \in M} \rho_{i,j}\, \hat{z}_i = \lambda_j + \sum_{i \in M} \rho_{i,j}\, \hat{w}_i$, where the second equality is by the fact that $(\hat{x}, \hat{z}) \in \mathcal{H}$ and third equality is by the fact that $\hat{z}_j = 0$ for all $j \in N \setminus M$. So, we have $(\hat{y}, \hat{w}) \in \mathcal{F}$.

We claim that $(\hat{y}, \hat{w})$ is an extreme point of $\mathcal{F}$. To get a contradiction, assume that $(\hat{y}, \hat{w})$ is not an extreme point of $\mathcal{F}$. By using the same argument in the proof of Lemma 10, we can show that the polyhedron $\mathcal{F}$ is bounded. In this case, there exist vectors $(y^1, w^1) \in \mathcal{F}$ and $(y^2, w^2) \in \mathcal{F}$

such that $\hat{y} = \alpha\, y^1 + (1-\alpha)\, y^2$ and $\hat{w} = \alpha\, w^1 + (1-\alpha)\, w^2$ for some $\alpha \in (0,1)$. We define the vector $(x^1, z^1) \in \Re^{2n}_+$ such that $x^1_j = y^1_j$ for all $j \in M$ and $x^1_j = 0$ for all $j \in N \setminus M$ and $z^1_j = w^1_j$ for all $j \in M$ and $z^1_j = 0$ for all $j \in N \setminus M$. Thus, the vectors x^1 and z^1 are obtained by adding zero components into the vectors y^1 and w^1 corresponding to the products in $N \setminus M$. Similarly, we define the vector $(x^2, z^2) \in \Re^{2n}_+$ such that $x^2_j = y^2_j$ for all $j \in M$ and $x^2_j = 0$ for all $j \in N \setminus M$ and $z^2_j = w^2_j$ for all $j \in M$ and $z^2_j = 0$ for all $j \in N \setminus M$. The vector $\hat{x}$ includes zero components for the products in $N \setminus M$ and includes the same components as the vector $\hat{y}$ for the products in N. Therefore, since $\hat{y} = \alpha\, y^1 + (1-\alpha)\, y^2$, we obtain $\hat{x} = \alpha\, x^1 + (1-\alpha)\, x^2$. Noting that $\hat{w} = \alpha\, w^1 + (1-\alpha)\, w^2$ and following a similar reasoning, we obtain $\hat{z} = \alpha\, z^1 + (1-\alpha)\, z^2$.

Next, we verify that $(x^1, z^1) \in \mathcal{H}$. Since $\hat{x}_j = \hat{z}_j = 0$ for all $j \in N \setminus M$ and $(\hat{x}, \hat{z}) \in \mathcal{H}$, we have $0 = \hat{x}_j + \hat{z}_j = \lambda_j + \sum_{i \in N} \rho_{i,j}\, \hat{z}_i = \lambda_j + \sum_{i \in M} \rho_{i,j}\, \hat{z}_i$ for all $j \in N \setminus M$. The last chain of equalities yields two results. First, we have $\lambda_j = 0$ for all $j \in N \setminus M$. Second, if $\hat{z}_i > 0$ for some $i \in M$, then we have $\rho_{i,j} = 0$ for all $j \in N \setminus M$. By the discussion in the previous paragraph, we have $\hat{z} = \alpha\, z^1 + (1-\alpha)\, z^2$. Thus, if $z^1_i > 0$ for some $i \in M$, then we have $\hat{z}_i > 0$, in which case, it follows that $\rho_{i,j} = 0$ for all $j \in N \setminus M$. By the definition of (x^1, z^1), for all $j \in N \setminus M$, we have $x^1_j = z^1_j = 0$, in which case, for all $j \in N \setminus M$, we have $x^1_j + z^1_j = 0 = \lambda_j + \sum_{i \in M} \rho_{i,j}\, z^1_i = \lambda_j + \sum_{i \in N} \rho_{i,j}\, z^1_i$, where the second equality uses the fact that $\lambda_j = 0$ for all $j \in N \setminus M$ and $\rho_{i,j} = 0$ for all $j \in N \setminus M$ whenever $z^1_i > 0$ for some $i \in M$, whereas the third equality uses the fact that $z^1_j = 0$ for all $j \in N \setminus M$. Thus, we have $x^1_j + z^1_j = \lambda_j + \sum_{i \in N} \rho_{i,j}\, z^1_i$ for all $j \in N \setminus M$. By the definition of x^1, the vectors x^1 and y^1 have the same components corresponding to the products in M. Similarly, by the definition of z^1, the vectors z^1 and w^1 have the same components corresponding to the products in M. In this case, for all $j \in M$, we have $x^1_j + z^1_j = y^1_j + w^1_j = \lambda_j + \sum_{i \in M} \rho_{i,j}\, w^1_i = \lambda_j + \sum_{i \in M} \rho_{i,j}\, z^1_i = \lambda_j + \sum_{i \in N} \rho_{i,j}\, z^1_i$, where the second equality uses the fact that $(y^1, w^1) \in \mathcal{F}$ and the fourth equality uses the fact that $\hat{z}^1_j = 0$ for all $j \in N \setminus M$. Thus, we have $x^1_j + z^1_j = \lambda_j + \sum_{i \in N} \rho_{i,j}\, z^1_i$ for all $j \in M$. The discussion in this paragraph shows that $x^1_j + z^1_j = \lambda_j + \sum_{i \in N} \rho_{i,j}\, z^1_i$ for all $j \in N$, establishing that $(x^1, z^1) \in \mathcal{H}$. A similar argument shows that $(x^2, z^2) \in \mathcal{H}$. By the discussion in the previous paragraph, we have $\hat{x} = \alpha\, x^1 + (1-\alpha) x^2$ and $\hat{z} = \alpha\, z^1 + (1-\alpha)\, z^2$ for some $\alpha \in (0,1)$. Thus, the vector $(\hat{x}, \hat{z})$ can be written as a nontrivial convex combination of $(x^1, z^1) \in \mathcal{H}$ and $(x^2, z^2) \in \mathcal{H}$, which contradicts the fact that $(\hat{x}, \hat{z})$ is an extreme point of $\mathcal{H}$. Therefore, the claim follows and $(\hat{y}, \hat{w})$ is an extreme point of $\mathcal{F}$.

The vectors $(\hat{x}, \hat{z})$ and $(\hat{y}, \hat{w})$ have the same components corresponding to the products in M. So, noting that $\hat{x}_j + \hat{z}_j > 0$ for all $j \in M$, we have $\hat{y}_j + \hat{w}_j > 0$ for all $j \in M$, which implies that $\hat{y}_j > 0$ or $\hat{w}_j > 0$, possibly both, for all $j \in M$. Thus, $(\hat{y}, \hat{w})$ is an extreme point of $\mathcal{F}$ and we have $\hat{y}_j > 0$ or $\hat{w}_j > 0$ for all $j \in M$. In this case, if we let $S_{\hat{y}} = \big\{ j \in M : \hat{y}_j > 0 \big\}$, then we can follow the same reasoning in the proof of Lemma 1, but focus on the products in M and the polyhedron $\mathcal{F}$, instead of the products in N and the polyhedron $\mathcal{H}$, to show that $\hat{w}_j = 0$ for all $j \in S_{\hat{y}}$. In particular, since $\mathcal{F}$ is a polyhedron defined by m equalities, the extreme point $(\hat{y}, \hat{w})$ can have at most m nonzero components. By the definition of $S_{\hat{y}}$, the vector $\hat{y}$ has $|S_{\hat{y}}|$ nonzero components. Since $\hat{y}_j = 0$ for all $M \setminus S_{\hat{y}}$ and $\hat{y}_j + \hat{w}_j > 0$ for all $j \in M \setminus S_{\hat{y}}$, we have $\hat{w}_j > 0$ for all

$j \in M \setminus S_{\hat{y}}$. So, the vector $\hat{w}$ has $m - |S_{\hat{y}}|$ nonzero components corresponding to the products in $M \setminus S_{\hat{y}}$, yielding a total of $|S_{\hat{y}}| + m - |S_{\hat{y}}| = m$ nonzero components in the vector $(\hat{y}, \hat{w})$. Therefore, the vector $(\hat{y}, \hat{w})$ can have no other nonzero components, which shows that $\hat{w}_j = 0$ for all $j \in S_{\hat{y}}$, as desired. In this case, since $\hat{w}_j = 0$ for all $j \in S_{\hat{y}}$, noting that the vectors $\hat{z}$ and $\hat{w}$ have the same components corresponding to the products in M and $S_{\hat{y}} \subset M$, we obtain $\hat{z}_j = 0$ for all $j \in S_{\hat{y}}$. Furthermore, since $\hat{y}_j = \hat{x}_j$ for all $j \in M$ and $\hat{x}_j = 0$ for all $j \in N \setminus M$, we have $S_{\hat{y}} = \{j \in M : \hat{y}_j > 0\} = \{j \in M : \hat{x}_j > 0\} = \{j \in N : \hat{x}_j > 0\} = S_{\hat{x}}$. Thus, we obtain $\hat{z}_j = 0$ for all $j \in S_{\hat{x}}$. By the definition of $S_{\hat{x}}$, we have $\hat{x}_j = 0$ for all $j \in N \setminus S_{\hat{x}}$. Finally, since $(\hat{x}, \hat{z}) \in \mathcal{H}$, we have $\hat{x}_j + \hat{z}_j = \lambda_j + \sum_{i \in N} \rho_{i,j}\, \hat{z}_i$ for all $j \in N$. Having $\hat{x}_j + \hat{z}_j = \lambda_j + \sum_{i \in N} \rho_{i,j}\, \hat{z}_i$ for all $j \in N$, $\hat{x}_j = 0$ for all $j \in N \setminus S_{\hat{x}}$ and $\hat{z}_j = 0$ for all $j \in S_{\hat{x}}$ implies that $(\hat{x}, \hat{z})$ satisfies the Balance equations for the subset of products $S_{\hat{x}}$. So, we have $P_{j,S_{\hat{x}}} = \hat{x}_j$ and $R_{j,S_{\hat{x}}} = \hat{z}_j$ for all $j \in N$.

Extension of Theorem 2. Letting $\hat{v} = (\hat{v}_1, \ldots, \hat{v}_n)$ be the optimal solution to the Dual problem, the proof of Theorem 2 uses the fact that $\hat{v}_j = r_j$ or $\hat{v}_j = \sum_{i \in N} \rho_{j,i}\, \hat{v}_i$ for all $j \in N$, which is a consequence of the assumption that $\lambda_j > 0$ for all $j \in N$. Our goal is to give a proof for Theorem 2 when we have $\lambda_j = 0$ for some $j \in N$. We use $\hat{v}$ to denote the optimal solution to the Dual problem and define $\hat{S} = \{j \in N : \hat{v}_j = r_j\}$. We want to show that $\hat{S}$ is the optimal solution to the Assortment problem. Without loss of generality, we assume that the revenues of the products are positive, since we can drop the products with negative revenues without changing the optimal solution to the Assortment problem, in which case, we can assume that $v \in \Re^n_+$ in the Dual problem. We let $M \subset N$ be such that $\hat{v}_j > r_j$ and $\hat{v}_j > \sum_{i \in N} \rho_{j,i}\, \hat{v}_i$ for all $j \in N \setminus M$. If $M = N$, then we have $\hat{v}_j = r_j$ or $\hat{v}_j = \sum_{i \in N} \rho_{j,i}\, \hat{v}_i$ for all $j \in N$ and the proof of Theorem 2 holds. Otherwise, letting $|M| = m$ for notational brevity, we consider the problem

$$\min_{w \in \Re^m_+} \left\{ \sum_{j \in M} \lambda_j\, w_j \;:\; w_j \geq r_j \;\forall\, j \in M, \quad w_j \geq \sum_{i \in M} \rho_{j,i}\, w_i \;\forall\, j \in M \right\}, \qquad \text{(Restricted)}$$

where we use the vector $w = \{w_j : j \in M\}$. We define the solution $\hat{w} = \{\hat{w}_j : j \in M\}$ to the problem above as $\hat{w}_j = \hat{v}_j$ for all $j \in M$. We claim that $\hat{w}$ is the optimal solution to the Restricted problem and the optimal objective value of the Restricted problem is equal to the optimal objective value of the Dual problem. To see the claim, by the definition of M, we have $\hat{v}_j > r_j$ and $\hat{v}_j > \sum_{i \in N} \rho_{j,i}\, \hat{v}_i$ for all $j \in N \setminus M$, which implies that the constraints in the Dual problem corresponding to the products in $N \setminus M$ are not tight at the optimal solution and we can drop these constraints. So, since we assume that $v \in \Re^n_+$ in the Dual problem, the Dual problem is equivalent to $\min_{v \in \Re^n_+} \{\sum_{j \in N} \lambda_j\, v_j : v_j \geq r_j \;\forall\, j \in M,\; v_j \geq \sum_{i \in N} \rho_{j,i}\, v_i \;\forall\, j \in M\}$. In the last optimization problem, the decision variables $\{v_j : j \in N \setminus M\}$ appear only in the objective function and the right side of the second constraint. Therefore, we can drop these decision variables by setting their values to their smallest possible value of zero, which establishes the claim.

The Restricted problem has the same structure as the Dual problem, but this problem focuses on the subset of products M, which is smaller than the set of products N. If there exists a set of products M' such that the optimal solution to the Restricted problem satisfies $\hat{w}_j > r_j$

and $\hat{w}_j > \sum_{i\in M} \rho_{j,i}\, \hat{w}_i$ for all $j \in M \setminus M'$, then we can repeat the same argument in the previous paragraph to come up with another problem that has the same structure as the Dual problem, but focuses on the subset of products M', which is smaller than the subset of products M. Repeating the argument recursively, we can assume that the optimal solution to the Restricted problem satisfies $\hat{w}_j = r_j$ or $\hat{w}_j = \sum_{i\in M} \rho_{j,i}\, \hat{w}_i$ for all $j \in M$, where $M \subset N$. Noting that $\hat{v}_j > r_j$ for all $j \in N \setminus M$ and $\hat{w}_j = \hat{v}_j$ for all $j \in M$, we obtain $\hat{S} = \{j \in N : \hat{v}_j = r_j\} = \{j \in M : \hat{v}_j = r_j\} = \{j \in M : \hat{w}_j = r_j\}$. We note that the last chain of equalities also implies that $\hat{S} \subset M$. Using the vectors $\bar{P}_{\hat{S}} = \{\bar{P}_{j,\hat{S}} : j \in M\}$ and $\bar{R}_{\hat{S}} = \{\bar{R}_{j,\hat{S}} : j \in M\}$, we let $(\bar{P}_{\hat{S}}, \bar{R}_{\hat{S}})$ be the solution to a version of the Balance equations given by $\bar{P}_{j,\hat{S}} + \bar{R}_{j,\hat{S}} = \lambda_j + \sum_{i\in M} \rho_{i,j}\, \bar{R}_{i,\hat{S}}$ for all $j \in M$, $\bar{P}_{j,\hat{S}} = 0$ for all $j \in M \setminus \hat{S}$ and $\bar{R}_{j,\hat{S}} = 0$ for all $j \in \hat{S}$. In this case, using the fact that $\hat{S} = \{j \in M : \hat{w}_j = r_j\}$ and the optimal solution to the Restricted problem satisfies $\hat{w}_j = r_j$ or $\hat{w}_j = \sum_{i\in M} \rho_{j,i}\, \hat{w}_i$ for all $j \in M$, we can follow precisely the same reasoning in the proof of Theorem 2 to show that

$$\sum_{j\in M} \bar{P}_{j,\hat{S}}\, r_j = \sum_{j\in M} \left\{ \bar{P}_{j,\hat{S}} + \bar{R}_{j,\hat{S}} - \sum_{i\in N} \rho_{i,j}\, \bar{R}_{i,\hat{S}} \right\} \hat{w}_j = \sum_{j\in M} \lambda_j\, \hat{w}_j.$$

If we can show that $P_{j,\hat{S}} \geq \bar{P}_{j,\hat{S}}$ for all $j \in M$, then we obtain $\sum_{j\in N} P_{j,\hat{S}}\, r_j \geq \sum_{j\in M} P_{j,\hat{S}}\, r_j \geq \sum_{j\in M} \bar{P}_{j,\hat{S}}\, r_j = \sum_{j\in M} \lambda_j\, \hat{w}_j = \sum_{j\in N} \lambda_j\, \hat{v}_j$, where the last equality uses the fact that the optimal objective value of the Restricted problem is equal to the optimal objective value of the Dual problem. By the discussion right before Theorem 2, the optimal objective value of the Dual problem corresponds to the optimal objective value of the Assortment problem. Thus, the last chain of inequalities implies that $\hat{S}$ must be the optimal solution to the Assortment problem, which is the desired result.

We proceed to showing that $P_{j,\hat{S}} \geq \bar{P}_{j,\hat{S}}$ for all $j \in M$. By the Balance equations, since $P_{j,\hat{S}} = 0$ for all $j \in N \setminus \hat{S}$ and $R_{j,\hat{S}} = 0$ for all $j \in \hat{S}$, we have $R_{j,\hat{S}} = \lambda_j + \sum_{i\in N\setminus\hat{S}} \rho_{i,j}\, R_{i,\hat{S}}$ for all $j \in N \setminus \hat{S}$. Noting that $\hat{S} \subset M \subset N$, we obtain $R_{j,\hat{S}} = \lambda_j + \sum_{i\in N\setminus M} \rho_{i,j}\, R_{i,\hat{S}} + \sum_{i\in M\setminus\hat{S}} \rho_{i,j}\, R_{i,\hat{S}}$ for all $j \in M \setminus \hat{S}$. By the definition of $(\bar{P}_{\hat{S}}, \bar{R}_{\hat{S}})$, since $\bar{P}_{j,\hat{S}} = 0$ for all $j \in M \setminus \hat{S}$ and $\bar{R}_{j,\hat{S}} = 0$ for all $j \in \hat{S}$, we also have $\bar{R}_{j,\hat{S}} = \lambda_j + \sum_{i\in M\setminus\hat{S}} \rho_{i,j}\, \bar{R}_{i,\hat{S}}$. Subtracting the last two equalities, we have $(R_{j,\hat{S}} - \bar{R}_{j,\hat{S}}) = \sum_{i\in N\setminus M} \rho_{i,j}\, R_{i,\hat{S}} + \sum_{i\in M\setminus\hat{S}} \rho_{i,j}\, (R_{i,\hat{S}} - \bar{R}_{i,\hat{S}})$ for all $j \in M \setminus \hat{S}$, which implies that $(R_{j,\hat{S}} - \bar{R}_{j,\hat{S}}) \geq \sum_{i\in M\setminus\hat{S}} \rho_{i,j}\, (R_{i,\hat{S}} - \bar{R}_{i,\hat{S}})$ for all $j \in M \setminus \hat{S}$. In this case, we can follow precisely the same reasoning in the proof of Lemma 11 to show that $R_{j,\hat{S}} \geq \bar{R}_{j,\hat{S}}$ for all $j \in M \setminus \hat{S}$. By the definition of $(\bar{P}_{\hat{S}}, \bar{R}_{\hat{S}})$ and the Balance equations, we obtain $\bar{P}_{j,\hat{S}} = \lambda_j + \sum_{i\in M\setminus\hat{S}} \rho_{i,j}\, \bar{R}_{i,\hat{S}} \leq \lambda_j + \sum_{i\in M\setminus\hat{S}} \rho_{i,j}\, R_{i,\hat{S}} \leq \lambda_j + \sum_{i\in N\setminus\hat{S}} \rho_{i,j}\, R_{i,\hat{S}} = P_{j,\hat{S}}$ for all $j \in M$, where the first and second equalities use the fact that $\bar{R}_{j,\hat{S}} = 0$ and $R_{j,\hat{S}} = 0$ for all $j \in \hat{S}$.

Extension of Lemma 3. If we have $\lambda_j = 0$ for some $j \in N$, then we need to modify the wording of Lemma 3 as follows. We use $\hat{v}^0$ to denote an optimal solution to the Dual problem and $\hat{v}^\eta$ to denote an optimal solution to the Dual problem when we decrease the revenues of all products by $\eta \geq 0$. In this case, Lemma 3 should state that there exists such optimal solutions to the Dual problem that satisfy $\{j \in N : \hat{v}^\eta_j = r_j - \eta\} \subset \{j \in N : \hat{v}^0_j = r_j\}$. The proof of Lemma 3 uses the fact that we have $\hat{v}^0_j = r_j$ or $\hat{v}^0_j = \sum_{i\in N} \rho_{j,i}\, \hat{v}^0_i$ for all $j \in N$ and $\hat{v}^\eta_j = r_j - \eta$ or

$\hat{v}_j^\eta = \sum_{i \in N} \rho_{j,i}\, \hat{v}_i^\eta$ for all $j \in N$. Even when we have $\lambda_j = 0$ for some $j \in N$, there exist solutions to the Dual problem that satisfy these properties. As a result, the proof of Lemma 3 continues to hold without any modifications. We observe that we need to modify the wording of the lemma since there can be multiple optimal solutions to the Assortment problem in general. In one optimal solution, we may include a particular product, but not include the same product in another optimal solution. If we have $\lambda_j > 0$ for all $j \in N$, then this issue is not a concern and Lemma 3 holds for any optimal solution to the Dual problem. On the other hand, if we have $\lambda_j = 0$ for some $j \in N$, then we can only guarantee that there exist optimal solutions to the Dual problem that satisfy the desired nesting property stated in Lemma 3.