Revenue Management Under the Markov Chain Choice Model

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

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

July 31, 2014

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 so as to maximize the expected revenue 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 and establish that the optimal policy can be implemented through protection levels. 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. This type of customer choice behavior is even more prevalent today with the common use of online sales channels that conveniently bring a variety of options to customers. When customers choose among the available products, the demand for a particular product naturally depends on what other products are made available to the customers, creating interactions between the demands for the different products. When such interactions exist between the demands for the different products, finding the right set of products to offer to customers can be a challenging task.

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

Main Contributions. First, we consider assortment optimization problems. In the assortment optimization setting, 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 set of products to offer so as to maximize the expected revenue obtained from each customer. We relate the probability of purchasing different products under the Markov chain choice model to the extreme points of a polyhedron (Lemma 1). Although the assortment optimization problem is inherently a combinatorial optimization problem, we use the relationship between the purchase probabilities and the extreme points of a polyhedron to give a linear program that can be used to obtain the optimal set of products to offer (Theorem 2). We show a useful structural property of the optimal assortment, which demonstrates that as the revenues of the products increase by the same amount, the optimal assortment to offer becomes larger (Lemma 3). This property becomes critical when we study the optimal policy for the single resource revenue management problem.

Second, we consider revenue management problems with a single resource. In this setting, we need to decide which set of products to make available to customers dynamically over a selling horizon. At each time period, an arriving customer chooses among the set of available products. There is a limited inventory of the resource and the sale of a product consumes one unit of the resource. The goal is to find a policy to decide with set of products to make available at each time period in the selling horizon so as to maximize the total expected revenue. Assuming that the customer choices are governed by the Markov chain choice model, we formulate the problem as a dynamic program and show that the optimal policy offers a larger set of products as we have more capacity at a certain time period or as we get closer to the end of the selling horizon, all else being equal (Theorem 4). In other words, as we have more capacity or we get closer to the end of the selling horizon, the urgency to liquidate the resource inventory takes over and we offer a larger set of products. Using these properties, we show that the optimal policy can be implemented by associating a protection level with each product so that a product is made available to the customers whenever the remaining inventory of the resource exceeds the protection level of the product. Thus, although the problem is formulated in terms of which set of products to offer to customers, we can individually decide whether to offer each product by comparing the remaining inventory of the resource with the protection level of a product. These intuitive properties of the optimal policy are consequences of the Markov chain choice model and they do not necessarily hold when customers choose according to other choice models.

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

Focusing on the deterministic linear program described above, we show that if the customers choose according to the Markov chain choice model, then the deterministic linear program can immediately be reduced to an equivalent linear program whose numbers of decision variables and constraints increase only linearly with the numbers of products and resources (Theorem 5). We develop an algorithm to recover the optimal solution to the original deterministic linear program by using the optimal solution to the reduced linear program. This algorithm allows us to recover the frequency with which we should offer each subset of products to customers by using the optimal solution to the reduced linear program. Finally, by using the reduced linear program, we show that the optimal solution to the deterministic linear program offers only nested subsets of products (Theorem 8). In other words, the subsets of products offered by the optimal solution to the deterministic linear program can be ordered such that one subset is included in another one. This result implies that the optimal solution to the deterministic linear program offers at most $n + 1$ different subsets, where n is the number of products. Therefore, the optimal solution to the deterministic linear program does not offer too many different subsets and these subsets are related to each other in the sense that they are nested. Similar to our results for the single resource revenue management problem, these properties of the optimal solution are consequences of the Markov chain choice model and they do not necessarily hold under other choice models. Our computational experiments show that using the reduced linear program can provide remarkable computational savings over solving the original deterministic linear program through column generation. For smaller problem instances with 50 resources and 500 products, we can obtain the optimal subset offer frequencies up to 10,000 times faster by using the reduced linear program, instead of solving the original deterministic linear program through column generation. For larger problem instances with 100 resources and 2,000 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.

Related Literature. The Markov chain choice model has recently been proposed by Blanchet et al. (2013). The authors demonstrate that the multinomial logit model, which is often used to model customer choices in practice, is a special case of the Markov chain choice model. They also show that the Markov chain choice model can approximate a rich class of choice models quite accurately. They study assortment optimization problems under the Markov chain choice model without inventory considerations and show that the optimal assortment can be computed through a dynamic program. We give an alternative solution approach for the assortment problem that is based on a linear program. This linear program becomes useful to derive structural properties of the optimal assortment. Also, 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.

Another paper closely related to our work is Talluri and van Ryzin (2004), where the authors formulate the single resource revenue management problem as a dynamic program and derive structural properties of the optimality policy under the assumption that the choices of the customers are governed by the multinomial logit model. Since the Markov chain choice model encapsulates the multinomial logit model as a special case, our results for the single resource revenue management problem are more general. A final paper that is of interest to us is Gallego et al. (2011). This paper shows 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 when customers choose according to 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. Overall, effectively solving single resource and network revenue management problems under the Markov chain choice model has important implications since the work of Blanchet et al. (2013) shows that the Markov chain choice model can approximate a rich class of choice models quite accurately.

Our work is also related to the growing literature on assortment optimization 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) and Desir and Goyal (2013) and Rusmevichientong et al. (2013) focus on assortment problems when the customer choices are governed by a mixture of multinomial logit models. Li and Rusmevichientong (2012), Li et al. (2013), Davis et al. (2013) and Gallego and Topaloglu (2014) develop tractable solution methods for the assortment problem when customers choose according to 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 (2011) study related pricing problems under the nested logit model.

Network revenue management literature under customer choice behavior is related to our work as well. As mentioned above, dynamic programming formulations of network revenue management problems tend to be intractable. Thus, it is customary to formulate deterministic linear programming approximations under the assumption that customer choices take on their expected values. Such deterministic linear programs go back to the work of Gallego et al. (2004) and Liu and van Ryzin (2008). Similar deterministic linear programs are studied by Bront et al. (2009), Kunnumkal and Topaloglu (2008), Talluri (2011), Meissner et al. (2012) and Vossen and Zhang (2013). 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 under customer choice.

Organization. In Section 1, we formulate the Markov chain choice model. In Section 2, we show how to solve the assortment optimization problem under the Markov chain choice model. In Section 3, we give structural properties of the optimal policy for the single resource revenue management problem and characterize the optimal policy through protection levels. In Section 4, 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 5, 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 6, we provide computational experiments. In Section 7, we conclude.

1 Markov Chain Choice Model

We describe the Markov chain choice model that we use throughout the paper. There are n products, indexed by $N = \{1, \ldots, n\}$. With probability λ_j, a customer arriving into the system considers purchasing product j. If this product is offered for purchase, then the customer purchases this product. Otherwise, the customer transitions into product i with probability $\rho_{j,i}$ and considers purchasing product i. With probability $1 - \sum_{i \in N} \rho_{j,i}$ the customer decides to leave the system without making a purchase. In this way, the customer transitions between the products according to a Markov chain until she considers purchasing a product that is offered for purchase or she decides to leave the system without making a purchase. Given that we offer the subset $S \subset N$ of products, we use $P_{j,S}$ to denote the probability that a customer considers purchasing product j during the course of her choice process and purchases this product, whereas we use $R_{j,S}$ to denote the probability that a customer considers purchasing product j during the course of her choice process and does not purchase this product due to the unavailability of this product. Using the vectors $P_S = (P_{1,S}, \ldots, P_{n,S})$ and $R_S = (R_{1,S}, \ldots, R_{n,S})$, we observe that (P_S, R_S) satisfies

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

We interpret the Balance equations as follows. Noting the definitions of $P_{j,S}$ and $R_{j,S}$, a customer considers purchasing product j during the course of her choice process with probability $P_{j,S} + R_{j,S}$. To consider purchasing product j, the customer should either consider purchasing product j when she arrives into the system or consider purchasing some product i during the course of her choice process, not purchase product i and transition from product i to product j. If product j is not offered, then the probability that the customer considers purchasing this product during the course of her choice process and purchases it is zero, implying that $P_{j,S} = 0$ for all $j \notin S$. Similarly, if product j is offered, then the probability that the customer considers purchasing this product during the course of her choice process and does not purchase it is zero, implying that $R_{j,S} = 0$ for all $j \in S$. So, if we offer the subset S of products, then a customer purchases product j with probability $P_{j,S}$. In this setup, a customer may visit an unavailable product multiple times. Substituting $P_{j,S} = 0$ for all $j \notin S$ and $R_{j,S} = 0$ for all $j \in S$ in the Balance equations, we can drop these probabilities altogether, but the current form of the Balance equations will be more convenient for us.

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

2 Assortment Optimization

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

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

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

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

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

Lemma 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, it follows that $(\hat{x}, \hat{z})$ has $|S_{\hat{x}}| + n - |S_{\hat{x}}| + 1 = n + 1$ nonzero components. Since an extreme point of a polytope defined by n equalities cannot have more than n nonzero components, we get a contradiction and the claim follows. By the claim and the definition of $S_{\hat{x}}$, we have $\hat{x}_j = 0$ for all $j \notin S_{\hat{x}}$ and $\hat{z}_j = 0$ for all $j \in S_{\hat{x}}$. Furthermore, noting that $(\hat{x}, \hat{z}) \in \mathcal{H}$, we have $\hat{x}_j + \hat{z}_j = \lambda_j + \sum_{i \in N} \rho_{i,j}\, \hat{z}_j$ for all $j \in N$. The last two statements imply that $(\hat{x}, \hat{z})$ satisfies the Balance equations with $S = S_{\hat{x}}$. Thus, it must be the case that $(\hat{x}, \hat{z}) = (P_{S_{\hat{x}}}, R_{S_{\hat{x}}})$. □

An important implication of Lemma 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

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

where we use the vector $v = (v_1, \ldots, v_n)$. In this 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 $\hat{S} = \{j \in N : \hat{v}_j = r_j\}$. 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. 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 $j \not\in \hat{S}$ 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. □

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. Blanchet et al. (2013) demonstrate the tractability of the Assortment problem, but they do not use a linear program. One useful implication of Theorem 2 is that it is possible to build on this theorem to 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. This result becomes useful when we show the optimality of protection level policies for the single resource revenue management problem. In the next lemma, we show that the optimal solution to the Assortment problem becomes a smaller subset as we decrease the product revenues.

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

Proof. By the same argument in the proof of Theorem 2, we have $\hat{v}^0_j = r_j$ or $\hat{v}^0_j = \sum_{i \in N} \rho_{j,i}\, \hat{v}^0_i$ for all $j \in N$, in which case, noting the constraints in the Dual problem, we obtain $\hat{v}^0_j = \max\{r_j, \sum_{i \in N} \rho_{j,i}\, \hat{v}^0_i\}$ for all $j \in N$. A similar argument yields $\hat{v}^\eta_j = \max\{r_j - \eta, \sum_{i \in N} \rho_{j,i}\, \hat{v}^\eta_i\}$ for all $j \in N$. We define $\tilde{v} = (\tilde{v}_1, \ldots, \tilde{v}_n)$ as $\tilde{v}_j = \min\{\hat{v}^0_j, \hat{v}^\eta_j + \eta\}$ for all $j \in N$. We claim that $\tilde{v}$ is a feasible solution to the Dual problem. To see the claim, by the definition of $\tilde{v}$, we have $\tilde{v}_j \leq \hat{v}^0_j$ and $\tilde{v}_j \leq \hat{v}^\eta_j + \eta$ for all $j \in N$. In this case, using the fact that $\hat{v}^0_j = \max\{r_j, \sum_{i \in N} \rho_{j,i}\, \hat{v}^0_i\}$, we obtain $\max\{r_j, \sum_{i \in N} \rho_{j,i}\, \tilde{v}_i\} \leq \max\{r_j, \sum_{i \in N} \rho_{j,i}\, \hat{v}^0_i\} = \hat{v}^0_j$ for all $j \in N$. Similarly, using the fact that $\hat{v}^\eta_j = \max\{r_j - \eta, \sum_{i \in N} \rho_{j,i}\, \hat{v}^\eta_i\}$, we have $\max\{r_j, \sum_{i \in N} \rho_{j,i}\, \tilde{v}_i\} \leq \max\{r_j, \sum_{i \in N} \rho_{j,i}\, (\hat{v}^\eta_i + \eta)\} \leq \max\{r_j - \eta, \sum_{i \in N} \rho_{j,i}\, \hat{v}^\eta_i\} + \eta = \hat{v}^\eta_j + \eta$ for all $j \in N$. Therefore, the discussion so far shows that $\max\{r_j, \sum_{i \in N} \rho_{j,i}\, \tilde{v}_i\} \leq \hat{v}^0_j$ and $\max\{r_j, \sum_{i \in N} \rho_{j,i}\, \tilde{v}_i\} \leq \hat{v}^\eta_j + \eta$, indicating that $\max\{r_j, \sum_{i \in N} \rho_{j,i}\, \tilde{v}_i\} \leq \min\{\hat{v}^0_j, \hat{v}^\eta_j + \eta\} = \tilde{v}_j$ for all $j \in N$, where the last equality is by the definition of $\tilde{v}$. Having $\max\{r_j, \sum_{i \in N} \rho_{j,i}\, \tilde{v}_i\} \leq \tilde{v}_j$ for all $j \in N$ implies that $\tilde{v}$ is a feasible solution to the Dual problem and the claim follows. To get a contradiction to the result that we want to show, we assume that there exists some $j \in N$ such that $j \in \{i \in N : \hat{v}^\eta_i = r_i - \eta\}$, but $j \not\in \{i \in N : \hat{v}^0_i = r_i\}$. So, we have $\hat{v}^\eta_j + \eta = r_j < \hat{v}^0_j$, in which case, noting that $\tilde{v}_j = \min\{\hat{v}^0_j, \hat{v}^\eta_j + \eta\}$, we get $\tilde{v}_j < \hat{v}^0_j$. Since $\tilde{v}_i \leq \hat{v}^0_i$ for all $i \in N$ by the definition of $\tilde{v}$, it follows that the components of $\tilde{v}$ are no larger than the corresponding components of $\hat{v}^0$ and there exists some $j \in N$ such that $\tilde{v}_j$ is strictly smaller than $\hat{v}^0_j$. 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. □

Noting Theorem 2, $\{j \in N : \hat{v}^\eta_j = r_j - \eta\}$ is the optimal solution to the ASSORTMENT problem when we decrease the revenues associated with all of the products by η. Therefore, Lemma 3 implies that if we decrease the revenues associated with all of the products by the same positive amount η, then the optimal solution to the ASSORTMENT problem becomes a smaller subset.

3 Single Resource Revenue Management

In the single resource revenue management setting, we manage one resource with a limited amount of capacity. At each time period in the selling horizon, we need to decide which subset of products to offer to customers. Customers arrive into the system one by one and choose among the offered products according to the Markov chain choice model. When we sell a product, we generate a revenue and consume one unit of the resource. The goal is to find a policy to dynamically decide which subsets of products to offer over the selling horizon so as to maximize the total expected revenue. Such single resource revenue management problems arise when airlines control the availability of different fare classes on a single flight leg. Different fare classes correspond to different products and the seats on the flight leg correspond to the resource. Talluri and van Ryzin (2004) consider this 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 $P_{j,S}$. There are T time periods in the selling horizon. For simplicity of notation, we assume that there is exactly one customer arrival 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

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

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 $\hat{S}_t(x) \subset \hat{S}_{t-1}(x)$. The first property implies that if we have fewer units of remaining capacity at a particular time period, then the optimal subset of products to offer becomes smaller. The second property implies that if we have more time periods left in the selling horizon with a particular number of units of remaining capacity, then the optimal subset of products to offer becomes smaller. The first property has an important implication when implementing the policy obtained from the Single Resource dynamic program. Since the optimal subset of products to offer becomes smaller when we have fewer units of remaining capacity at a time period, we let $\bar{x}_{jt}$ be the smallest value of the remaining capacity such that it is still optimal to offer product j at time period t. In this case, if we have x units of remaining capacity at the beginning of time period t and $x \geq \bar{x}_{jt}$, then it is optimal to offer product j. Otherwise, it is optimal not to offer product j. Therefore, we can associate a threshold value $\bar{x}_{jt}$ for each product j and time period t such that we can decide whether it is optimal to offer product j at time period t by comparing the remaining resource capacity with the threshold value. The threshold value $\bar{x}_{jt}$ is referred to as the protection level for product j at time period t and the resulting policy is referred to as a protection level policy.

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

Theorem 4 *There exists an optimal policy for the* Single Resource *dynamic program such that* $\hat{S}_t(x-1) \subset \hat{S}_t(x)$ *and* $\hat{S}_t(x) \subset \hat{S}_{t-1}(x)$.

Proof. It is a standard result that the first differences of the value functions computed through the Single Resource dynamic program increases as we have fewer units of remaining capacity or as we have more time periods left in the selling horizon. In particular, letting $\Delta V_t(x) = V_t(x) - V_t(x-1)$, 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 $\hat{S}_{t-1}(x) \subset \hat{S}_t(x)$. □

By Theorem 4, implementing the optimal policy obtained from the SINGLE RESOURCE dynamic program requires only keeping track of the optimal protection level $\bar{x}_{jt}$ for all $j \in N$ and $t = 1, \ldots, T$, instead of keeping track of the optimal subset of products $\hat{S}_t(x)$ to offer for all $x = 0, \ldots, c$ and $t = 1, \ldots, T$. In the next section, we consider network revenue management problems when customers choose according to the Markov chain choice model.

4 Network Revenue Management

In the network revenue management setting, we manage a network of resources, each of which has a limited amount of capacity. At each time period in the selling horizon, we need to decide which subset of products to offer to customers. Customers arrive into the system one by one and choose among the offered products according to the Markov chain choice model. Each product uses a certain combination of resources. When we sell a product, we generate a revenue and consume the capacities of the resources used by the product. The goal is to find a policy to dynamically decide which subsets of products to make available over the selling horizon so as to maximize the total expected revenue. Such network revenue management problems model the situation faced by an airline making itinerary availability decisions over a network of flight legs. Different itineraries correspond to different products and seats on different flight legs correspond to different resources. Gallego et al. (2004) and Liu and van Ryzin (2008) study network revenue management problems when customers choose according to a general choice model. The dynamic programming formulation of the problem requires keeping track of the remaining capacities on all of the flight legs, resulting in a high dimensional state variable. To deal with this difficulty, the authors formulate a linear program under the assumption that the customer choices take on their expected values. In practice, this linear program may have a large number of decision variables and it is commonly solved through column generation. In this section, we show that the size of the linear program can be drastically reduced when customers choose under the Markov chain choice model.

There are m resources indexed by $M = \{1, \ldots, m\}$. Similar to the previous two sections, we index the products by $N = \{1, \ldots, n\}$. Each customer chooses among the offered products according to the Markov chain choice model. Therefore, if we offer the subset S of products, then a customer purchases product j with probability $P_{j,S}$, where (P_S, R_S) satisfies the BALANCE equations. There are T time periods in the selling horizon. For simplicity of notation, we assume that exactly 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. Under the assumption that the customer choices take on their expected values, we can formulate the network revenue management problem as linear program. In particular, using the decision variable u_S to capture the probability of offering subset S of products at a time period, we can find good subset offer probabilities $u = \{u_S : S \subset N\}$ by solving the linear program

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

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

In our formulation of the network revenue management problem, the probabilities (P_S, R_S) do not depend on the time period, which allows us to use a single decision variable u_S in the CHOICE BASED linear program to capture the probability of offering subset S at any time period. In practical applications, customers arriving at different time periods can choose according to choice models with different parameters. If the choice models governing the customer choices at different time periods have different parameters, then we can address this situation by using the decision variable $u_{S,t}$, which corresponds to the probability of offering subset S of products at time period t. Our results in this section continue to hold with simple modifications when we work with the decision variables $\{u_{S,t} : S \subset N,\ t = 1, \ldots, T\}$. We can interpret the CHOICE BASED linear program as a crude approximation to the network revenue management problem formulated under the assumption that the customer choices take on their expected values. Liu and van Ryzin (2008) propose various approximate policies that use the solution to the CHOICE BASED linear program to decide which subset of products to offer at each time period.

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

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

By the definition of the decision variable x_j, the expression Tx_j in the objective function and the first constraint of the problem above corresponds to the expected number of sales for product j over the selling horizon. In this case, the objective function of the problem above computes the total expected revenue over the selling horizon. The first constraint ensures that the total expected capacity of resource q consumed over the selling horizon does not exceed the capacity of this resource. The second constraint is similar to the Balance equations. Noting the definitions of the decision variables x_j and z_j, $x_j + z_j$ is the fraction of customers that consider product j during the course of their choice process. For a customer to consider product j during the course of her choice process, she should either consider purchasing product j when she arrives into the system or consider purchasing some product i, not purchase this product and transition from product i to product j. There are $2\,n$ decision variables and $m+n$ constraints in the Reduced linear program and it can be possible to solve the Reduced linear program for practical networks directly by using linear programming software. In the next theorem, we show that the Reduced linear program is equivalent to the Choice Based linear program in the sense that we can use the optimal solution to the Reduced linear program to obtain the optimal solution to the Choice Based one.

Theorem 5 *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 $\hat{z} = \sum_{k=1}^K \gamma^k\, R_{S^k}$. 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 $S \not\in \{S^1,\ldots,S^K\}$ 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 $(\hat{x},\hat{z})\in\mathcal{H}$. 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 $S^k = \{j\in N : x^k_j > 0\}$. By Lemma 1, we have $P_{j,S^k} = x^k_j$ and $R_{j,S^k} = z^k_j$ for all $j\in N$, which implies that $\hat{x} = \sum_{k=1}^K \gamma^k\, x^k = \sum_{k=1}^K \gamma^k\, P_{S^k}$ and $\hat{z} = \sum_{k=1}^K \gamma^k\, z^k = \sum_{k=1}^K \gamma^k\, R_{S^k}$, showing the first part of the theorem. To show the second part of the theorem, by the definition of $\hat{u}$, we have $\sum_{S\subset N} P_{j,S}\,\hat{u}_S = \sum_{k=1}^K P_{j,S^k}\,\gamma^k = \hat{x}_j$. Thus, we obtain $\sum_{j\in N}\sum_{S\subset N} T\, a_{q,j}\, P_{j,S}\,\hat{u}_S = \sum_{j\in N} T\, a_{q,j}\,\hat{x}_j \le c_q$, where the inequality uses the fact that

$(\hat{x}, \hat{z})$ is a feasible solution to the Reduced linear program. Furthermore, we have $\sum_{S \subset N} \hat{u}_S = \sum_{k=1}^K \gamma^k = 1$. Therefore, $\hat{u}$ is a feasible solution to the Choice Based linear program. Since $\sum_{S \subset N} P_{j,S} \, \hat{u}_S = \hat{x}_j$, we obtain $\sum_{S \subset N} \sum_{j \in N} T \, r_j \, P_{j,S} \, \hat{u}_S = \sum_{j \in N} T \, r_j \, \hat{x}_j$, which implies that the solution $\hat{u}$ provides an objective value for the Choice Based linear program that is equal to the optimal objective value of the Reduced linear program. Thus, the optimal objective value of the Choice Based linear program is at least as large as that of the Reduced linear program.

On the other hand, let $\tilde{u}$ be the optimal solution to the Choice Based linear program. Define $(\tilde{x}, \tilde{z})$ as $\tilde{x}_j = \sum_{S \subset N} P_{j,S} \, \tilde{u}_S$ and $\tilde{z}_j = \sum_{S \subset N} R_{j,S} \, \tilde{u}_S$ for all $j \in N$. We have $\sum_{j \in N} T \, a_{q,j} \, \tilde{x}_k = \sum_{S \subset N} \sum_{j \in N} T \, a_{q,j} \, P_{j,S} \, \tilde{u}_S \leq c_q$, 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 $\tilde{x}_j + \tilde{z}_j = \sum_{S \subset N} P_{j,S} \, \tilde{u}_S + \sum_{S \subset N} R_{j,S} \, \tilde{u}_S = \lambda_j + \sum_{S \subset N} \sum_{i \in N} \rho_{i,j} \, R_{i,S} \, \tilde{u}_S = \lambda_j + \sum_{i \in N} \rho_{i,j} \, \tilde{z}_j$, where the first and last equalities use the definitions of $\tilde{x}$ and $\tilde{z}$. Thus, $(\tilde{x}, \tilde{z})$ is a feasible solution to the Reduced linear program. Noting that $\sum_{j \in N} T \, r_j \, \tilde{x}_j = \sum_{S \subset N} \sum_{j \in N} T \, r_j \, P_{j,S} \, \tilde{u}_S$, the solution $(\tilde{x}, \tilde{z})$ provides an objective value for the Reduced linear program that is equal to the optimal objective value of the Choice Based linear program. Therefore, the optimal objective value of the Reduced linear program is at least as large as that of the Choice Based linear program. Noting the discussion at the end of the previous paragraph, the Choice Based and Reduced linear programs have the same optimal objective value and the solution $\hat{u}$ defined in the theorem is optimal to the Choice Based linear program. $\square$

By Theorem 5, we have a general approach for recovering the optimal solution to the Choice Based linear program by using the optimal solution to the Reduced linear program. Letting $(\hat{x}, \hat{z})$ be the optimal solution to the Reduced linear program, the key is to find subsets $S^1, \ldots, S^K$ and positive scalars $\gamma^1, \ldots, \gamma^K$ summing to one such that $(\hat{x}, \hat{z})$ can be expressed as $\hat{x} = \sum_{k=1}^K \gamma^k \, P_{S^k}$ and $\hat{z} = \sum_{k=1}^K \gamma^k \, R_{S^k}$. Finding such subsets and scalars is not a straightforward task. In the next section, we give a tractable algorithm for this purpose.

5 Recovering the Optimal Solution

In this section, we consider the question of recovering the optimal solution to the Choice Based linear program by using the optimal solution to the Reduced linear program. By the discussion at the end of the previous section, letting $(\hat{x}, \hat{z})$ be the optimal solution to the Reduced linear program, the key is to find subsets $S^1, \ldots, S^K$ and positive scalars $\gamma^1, \ldots, \gamma^K$ summing to one such that $(\hat{x}, \hat{z})$ can be expressed as $\hat{x} = \sum_{k=1}^K \gamma^k \, P_{S^k}$ and $\hat{z} = \sum_{k=1}^K \gamma^k \, R_{S^k}$. In this case, the solution $\hat{u}$ obtained by letting $\hat{u}_{S^k} = \gamma^k$ for all $k = 1, \ldots, K$ and $\hat{u}_S = 0$ for all $S \notin \{S^1, \ldots, S^K\}$ is optimal to the Choice Based linear program. One possible approach to find the subsets $S^1, \ldots, S^K$ and scalars $\gamma^1, \ldots, \gamma^K$ is to express $(\hat{x}, \hat{z})$ as $\hat{x} = \sum_{S \subset N} \gamma^S \, P_S$ and $\hat{z} = \sum_{S \subset N} \gamma^S \, R_S$ for unknown positive scalars $\{\gamma_S : S \subset N\}$ satisfying $\sum_{S \subset N} \gamma^S = 1$ and solve these equations for the unknown scalars. Since there are $2\,n+1$ equations, there exists a solution where at most $2\,n+1$ of the unknown scalars take strictly positive values, but solving these equations directly is not tractable since the number of unknown scalars grows exponentially with the number of products. We can use an idea similar to column generation, where we focus on a small subset of the unknown scalars $\{\alpha_S : S \subset N\}$ and iteratively extend this subset, but this approach can be as computationally intensive as solving the Choice Based linear program directly by using column generation, which ultimately becomes problematic when dealing with large problem instances.

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

In the second case, we assume that $\alpha_{\hat{x}} \geq 1$. Having $\alpha_{\hat{x}} \geq 1$ implies that $\hat{x}_j \geq P_{j,S_{\hat{x}}}$ for all $j \in S_{\hat{x}}$. By the definition of $S_{\hat{x}}$ and the Balance equations, we also have $\hat{x}_j = 0 = P_{j,S_{\hat{x}}}$ for all $j \notin S_{\hat{x}}$. Thus, we have $\hat{x}_j \geq P_{j,S_{\hat{x}}}$ for all $j \in N$. Since $(\hat{x}, \hat{z}) \in \mathcal{H}$, Lemma 10 in the appendix shows that we also have $\hat{z}_j \geq R_{j,S_{\hat{x}}}$ for all $j \in N$. In this case, using the fact that $(\hat{x}, \hat{z}) \in \mathcal{H}$, if we add the equalities in the definition of $\mathcal{H}$ for all $j \in N$ and arrange the terms, then we get $\sum_{j\in N} \hat{x}_j + \sum_{j \in N}(1 - \sum_{i\in N} \rho_{j,i})\, \hat{z}_j = \sum_{j\in N} \lambda_j$. Similarly, adding the Balance equations for all $j \in N$, we get $\sum_{j\in N} \lambda_j = \sum_{j\in N} P_{j,S_{\hat{x}}} + \sum_{j\in N}(1 - \sum_{i\in N} \rho_{j,i})\, R_{j,S_{\hat{x}}}$. Thus, we obtain $\sum_{j\in N} \hat{x}_j + \sum_{j\in N}(1 - \sum_{i\in N} \rho_{j,i})\, \hat{z}_j = \sum_{j\in N} P_{j,S_{\hat{x}}} + \sum_{j\in N}(1 - \sum_{i\in N} \rho_{j,i})\, R_{j,S_{\hat{x}}}$. Since we have $\hat{x}_j \geq P_{j,S_{\hat{x}}}$ and $\hat{z}_j \geq R_{j,S_{\hat{x}}}$ for all $j \in N$, the last equality implies that $\hat{x}_j = P_{j,S_{\hat{x}}}$ and $\hat{z}_j = R_{j,S_{\hat{x}}}$ for all $j \in N$. Thus, we can express $(\hat{x}, \hat{z})$ simply as $\hat{x} = P_{S_{\hat{x}}}$ and $\hat{z} = R_{S_{\hat{x}}}$ and we are done. As a side note, if $\alpha_{\hat{x}} > 1$, then we have $\hat{x}_j > P_{j,S_{\hat{x}}}$ for all $j \in N$. Using the fact that $\hat{z}_j \geq R_{j,S_{\hat{x}}}$ for all $j \in N$, we get a contradiction to the last equality. So, the largest possible value of $\alpha_{\hat{x}}$ is one.

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

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 6 *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 10 in the appendix, 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 $\hat{z} = \sum_{k=1}^K \gamma^k \, R_{S^k}$. We refer to this algorithm as the Dimension Reduction algorithm.

Dimension Reduction

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

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

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

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

Steps 1, 2 and 3 in the Dimension Reduction algorithm respectively correspond to the first, second and third cases considered at the beginning of this section. We can use induction over the iterations of the algorithm to show that $(x^k, z^k) \in \mathcal{H}$ at each iteration k. In particular, since $(\hat{x}, \hat{z})$ is a feasible solution to the Reduced linear program, we have $(x^1, z^1) \in \mathcal{H}$. Assuming that $(x^k, z^k) \in \mathcal{H}$, if we identify (x^k, z^k) and (x^{k+1}, z^{k+1}) in the algorithm respectively with $(\hat{x}, \hat{z})$ and $(\hat{u}, \hat{v})$ in Lemma 6, then this lemma implies that $(x^{k+1}, z^{k+1}) \in \mathcal{H}$, completing the induction. On the other hand, letting $j^k = \arg\min\{x_j^k / P_{j,S^k} : j \in S^k\}$ and noting that $(x^k, z^k) \in \mathcal{H}$, if we identify S^k and S^{k+1} in the algorithm with $S_{\hat{x}}$ and $S_{\hat{u}}$ in Lemma 6, then this lemma implies that $S^{k+1} \subset S^k \setminus \{j^k\}$. Therefore, x^{k+1} has strictly fewer nonzero components than x^k. This observation implies that the algorithm stops after at most $n+1$ iterations with $S^{n+1} = \varnothing$. In the next theorem, we show that we can use the Dimension Reduction algorithm to find subsets $S^1, \ldots, S^K$ and positive scalars $\gamma^1, \ldots, \gamma^K$ summing to one such that the optimal solution $(\hat{x}, \hat{z})$ to the Reduced linear program can be expressed as $\hat{x} = \sum_{k=1}^K \gamma^k P_{S^k}$ and $\hat{z} = \sum_{k=1}^K \gamma^k R_{S^k}$.

Theorem 7 *Assume that the* Dimension Reduction *algorithm stops at iteration K and generates the subsets $S^1, \ldots, S^K$ and the scalars $\alpha^1, \ldots, \alpha^K$. Then, letting $\gamma^k = (1-\alpha^1) \ldots (1-\alpha^{k-1})\, \alpha^k$ for all $k = 1, \ldots, K$, we have $\hat{x} = \sum_{k=1}^K \gamma^k P_{S^k}$, $\hat{z} = \sum_{k=1}^K \gamma^k R_{S^k}$ and $\sum_{k=1}^K \gamma^k = 1$.*

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

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

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 $\alpha^K \geq 1$ and we can follow the same argument in the second case at the beginning of this section to conclude that $(x^K, z^K) = (P_{S^K}, R_{S^K})$. Furthermore, if the algorithm stops in Step 2, then the discussion in the second case at the beginning of this section shows that $\alpha^K \leq 1$. Thus, we must have $\alpha^K = 1$. Therefore, we always have $(x^K, z^K) = (P_{S^K}, R_{S^K})$ and $\alpha^K = 1$ when the DIMENSION REDUCTION algorithm stops. In this case, using the equalities $\hat{x} = \gamma^1 P_{S^1} + \ldots + \gamma^{k-1} P_{S^{k-1}} + (\gamma^k/\alpha^k)\, x^k$ and $\hat{z} = \gamma^1 R_{S^1} + \ldots + \gamma^{k-1} R_{S^{k-1}} + (\gamma^k/\alpha^k)\, z^k$ with $k = K$, we obtain $\hat{x} = \sum_{k=1}^K \gamma^k P_{S^k}$ and $\hat{z} = \sum_{k=1}^K \gamma^k R_{S^k}$. Finally, noting that $\alpha^K = 1$ and using the definition of γ^k in the theorem, it follows that $\gamma^{K-1} + \gamma^K = (1-\alpha^1) \ldots (1-\alpha^{K-2})$. Repeating the same argument recursively, we obtain $\gamma^2 + \ldots + \gamma^K = (1-\alpha^1)$, in which case, since $\gamma^1 = \alpha^1$ by the definition of γ^k, we get $\gamma^1 + \gamma^2 + \ldots + \gamma^K = 1$. □

By Theorem 7, 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 5 implies that the solution $\hat{u}$ obtained by letting $\hat{u}_{S^k} = \gamma^k$ for all $k = 1, \ldots, K$ and $\hat{u}_S = 0$ for all $S \notin \{S^1, \ldots, S^K\}$ is optimal to the CHOICE BASED linear program. These observations provide a tractable approach for recovering the optimal solution to the CHOICE BASED linear program by using the optimal solution to the REDUCED linear program. In the next theorem, we use the DIMENSION REDUCTION algorithm to show structural properties of the subsets of products offered by the optimal solution to the CHOICE BASED linear problem. In particular, we show that the optimal solution to the CHOICE BASED linear program offers at most $n+1$ different subsets and these subsets are nested in the sense that one of the subsets is included in another one, naturally with the exception of the largest subset.

Theorem 8 *There exists an optimal solution $\hat{u}$ to the* CHOICE BASED *linear program, where at most $n+1$ of the decision variables can take on nonzero values. Furthermore, letting $\hat{u}_{S^1}, \ldots, \hat{u}_{S^{n+1}}$ be these decision variables, we have $S^1 \supset \ldots \supset S^{n+1}$.*

Proof. Letting the subsets $S^1, \ldots, S^K$ be generated by the DIMENSION REDUCTION algorithm, Theorems 5 and 7 imply that there exists an optimal solution $\hat{u}$ to the CHOICE BASED linear program, where we have $\hat{u}_{S^k} \geq 0$ for all $k = 1, \ldots, K$ and $\hat{u}(S) = 0$ for all $S \notin \{S^1, \ldots, S^K\}$. By discussion that follows the description of the DIMENSION REDUCTION algorithm, the algorithm stops after at most $n+1$ iterations, which implies that $K \leq n+1$ and the first part of the theorem follows. By the discussion that follows the description of the DIMENSION REDUCTION algorithm, we also have $S^k \supset S^{k+1}$ for all $k = 1, \ldots, K-1$ and the second part of the theorem follows. □

In the next section, we provide computational experiments that test the performance of the DIMENSION REDUCTION algorithm.

6 Computational Experiments

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.

6.1 Experimental Setup

In our computational experiments, we generate a number of network revenue management problem instances. For each problem instance, we use two strategies to obtain the optimal solution to the Choice Based linear program. The first strategy solves the Choice Based linear program directly by using column generation. We refer to this strategy as CG, standing for column generation. The second strategy solves the Reduced linear program and carries out the Dimension Reduction algorithm to recover the optimal solution to the Choice Based linear program by using the optimal solution to the Reduced linear program. We refer to this strategy as DR, standing for dimension reduction. Our objective is to compare the performances of CG and DR. We use the following approach to generate our test problems. We sample β_j from the uniform distribution over $[0, 1]$ and set $\lambda_j = \beta_j / \sum_{i \in N} \beta_i$. Similarly, we sample $\sigma_{j,i}$ from the uniform distribution over $[0, 1]$ and set $\rho_{j,i} = (1 - P_0)\, \sigma_{j,i} / \sum_{k \in N} \sigma_{j,k}$, where P_0 is a parameter that we vary. In this case, if a customer considers purchasing product j and product j is not available, then she leaves without making a purchase with probability $1 - \sum_{i \in N} \rho_{j,i} = 1 - (1 - P_0) \sum_{i \in N} \sigma_{j,i} / \sum_{k \in N} \sigma_{j,k} = P_0$. 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 solve the problem $\max_{S \subset N} \big\{ \sum_{j \in N} r_j \, P_{j,S} \big\}$ to obtain the optimal subset of products to offer under the assumption that we have unlimited resource capacities. Letting S^* be the optimal solution to this problem, we set the capacity of resource q as $c_q = \kappa \sum_{j \in N} T \, a_{q,j} \, P_{j,S^*}$, where κ is another parameter that we vary. We note that $\sum_{j \in N} T \, a_{q,j} \, P_{j,S^*}$ is the total expected capacity consumption of resource q when we offer the subset S^* of products that is computed under the assumption that we have unlimited resource capacities. Therefore, the capacity of resource q is obtained by multiplying this total expected capacity consumption by κ and the parameter κ controls the tightness of the resource capacities.

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

6.2 Computational Results

Table 1 summarizes our main computational results. The first column in this table shows the problem instances by using the tuple (m, n, P_0, ξ, κ). The second column shows the CPU seconds for CG. The third column shows the CPU seconds for DR. The fourth column shows the ratio of the CPU seconds for CG and DR, corresponding to the relative improvement in the CPU seconds obtained by recovering the optimal solution to the Choice Based linear program by using the Reduced linear program, instead of solving the Choice Based linear program directly by using column generation. The fifth column shows what portion of the CPU seconds for DR is spent on solving the Reduced linear program, whereas the sixth column shows what portion of the CPU second for DR is spent on carrying out the Dimension Reduction algorithm.

The results in Table 1 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 1,836 and there is a test problem where DR can improve the CPU seconds for CG by a factor exceeding 10,000. We observe a number of test problems where the CPU seconds for CG exceed 7,200, corresponding to more than two hours of run time. For these test problems, DR can obtain the optimal solution to the Choice Based linear program within only 2.5 seconds. About 80% of the CPU seconds for DR is spent on solving the Reduced linear program and the rest is spent on carrying out the Dimension Reduction algorithm. In Table 1, we observe two trends in the performance improvements provided by DR. First, as the number of resources and 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 4,229. Second, as κ decreases and the capacities of the resources become tighter, the performance improvement provided by CG also tends to increase. For the test problems with $\kappa = 0.8$, DR improves the CPU seconds for CG by an average factor of 1,173, whereas for the test problems with $\kappa = 0.6$, DR improves the CPU seconds for CG by an average factor of 2,498. Overall, our computational results indicate that DR can be faster than CG by orders of magnitude in obtaining the optimal solution to the Choice Based linear program.

Column generation approaches can be fast in obtaining a near optimal solution, but they can also be slow in ultimately reaching the optimal solution, which can be a factor in the relatively poor performance of CG in Table 1. To ensure that this concern is not a significant factor in our computational experiments, Table 2 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 1, the first column in this table shows the problem instances. The second column shows the CPU seconds for CG to obtain a solution within 1% of the optimal objective value. For comparison purposes, the third column shows the CPU seconds for DR to obtain the optimal solution to the Choice Based linear program. Thus, the entries of the third column are

			CPU		
Test Problem	CPU Secs. for		Secs.	Red.	Dim.
$(m,\ n,\ P_0,\ \xi,\ \kappa)$	CG	DR	Ratio	Lin.	Red.
(25, 250, 0.1, 0.02, 0.8)	162	0.23	722	0.18	0.05
(25, 250, 0.1, 0.02, 0.6)	200	0.35	571	0.27	0.08
(25, 250, 0.1, 0.2, 0.8)	112	0.25	447	0.20	0.05
(25, 250, 0.1, 0.2, 0.6)	170	0.22	772	0.18	0.04
(25, 250, 0.3, 0.02, 0.8)	59	0.22	270	0.17	0.05
(25, 250, 0.3, 0.02, 0.6)	137	0.19	722	0.15	0.04
(25, 250, 0.3, 0.2, 0.8)	36	0.26	139	0.22	0.04
(25, 250, 0.3, 0.2, 0.6)	104	0.27	384	0.23	0.04
(25, 500, 0.1, 0.02, 0.8)	1,575	1.58	997	1.22	0.36
(25, 500, 0.1, 0.02, 0.6)	1,957	1.36	1,439	0.98	0.38
(25, 500, 0.1, 0.2, 0.8)	1,772	2.00	886	1.63	0.37
(25, 500, 0.1, 0.2, 0.6)	2,487	1.63	1,526	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)	1,174	1.52	772	1.18	0.34
(25, 500, 0.3, 0.2, 0.8)	559	1.99	281	1.67	0.32
(25, 500, 0.3, 0.2, 0.6)	1,383	2.01	688	1.66	0.35
(50, 250, 0.1, 0.02, 0.8)	410	0.22	1,831	0.18	0.05
(50, 250, 0.1, 0.02, 0.6)	926	0.22	4,208	0.16	0.06
(50, 250, 0.1, 0.2, 0.8)	398	0.29	1,374	0.24	0.05
(50, 250, 0.1, 0.2, 0.6)	468	0.25	1,901	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	1,758	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	1,495	0.21	0.05
(50, 500, 0.1, 0.02, 0.8)	6,628	1.75	3,788	1.40	0.35
(50, 500, 0.1, 0.02, 0.6)	15,500	1.50	10,333	1.11	0.39
(50, 500, 0.1, 0.2, 0.8)	7,739	2.12	3,651	1.74	0.38
(50, 500, 0.1, 0.2, 0.6)	10,659	1.90	5,610	1.50	0.40
(50, 500, 0.3, 0.02, 0.8)	3,008	1.84	1,635	1.54	0.30
(50, 500, 0.3, 0.02, 0.6)	6,780	1.64	4,134	1.29	0.35
(50, 500, 0.3, 0.2, 0.8)	2,392	2.32	1,031	1.99	0.33
(50, 500, 0.3, 0.2, 0.6)	7,600	2.08	3,654	1.71	0.37
Average	2,378	1.03	1,836	0.87	0.16

Table 1: CPU seconds for CG and DR.

identical to those of the third column of Table 1. The fourth column in Table 2 shows the ratios of the CPU seconds in the second and third columns. The results indicate that DR continues to provide substantial performance improvements over CG even when we terminate CG once it obtains a solution within 1% of the optimal objective value. There are problem instances, where DR improves the CPU seconds for CG by factors exceeding 1,000. Over all of our test problems, DR improves the performance of CG by an average factor of 286. Similar to the trends in Table 1, the results in Table 2 indicate that the performance improvements provided by DR become more pronounced as the size of the problem, measured by the number of resources and products, increases, or as the capacities on the flight legs become tighter.

Table 3 provides computational experiments for test problems with 100 resources and 2,000 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

	CPU Secs. for		CPU
Test Problem	CG	DR	Secs.
$(m, \quad n, \; P_0, \quad \xi, \quad \kappa)$	1%	Opt.	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
Average	115	0.98	114

	CPU Secs. for		CPU
Test Problem	CG	DR	Secs.
$(m, \quad n, \; P_0, \quad \xi, \quad \kappa)$	1%	Opt.	Ratio
(50, 250, 0.1, 0.02, 0.8)	60	0.22	269
(50, 250, 0.1, 0.02, 0.6)	231	0.22	1,050
(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	1,734
(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
Average	528	1.07	458

Table 2: CPU seconds for CG and DR when we terminate CG once it obtains a solution within 1% of the optimal objective value.

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 3 lists the test problems by using the tuple (m, n, P_0, ξ, κ). The second column shows the optimality gaps obtained by CG after two hours of run time. The third column shows the CPU seconds for DR to obtain the optimal solution to the Choice Based linear program. On average, DR obtains the optimal solution to the Choice Based linear program in less than three minutes. The average optimality gap of the solutions obtained by CG even after two hours of run time is 7.2%. There are test problems where CG terminates with more than 10% optimality gap after two hours of run time. In contrast, the largest CPU seconds for DR is about 225. Our results indicate that DR continues to provide dramatic improvements over CG for larger test problems.

7 Conclusions

In this paper, we studied three classes of revenue management problems when customers choose according to the Markov chain choice model. For the assortment optimization setting, we showed that the choice probabilities under the Markov chain model can be associated with the extreme points of a polyhedron. Using this result, we showed that the assortment optimization problem can be formulated as a linear program. Also, we derived a structural property that shows that the optimal assortment becomes smaller as we decrease the product revenues by the same positive amount. For the single resource revenue management setting, we showed that the optimal policy can be implemented by associating a protection level with each product so that it is optimal

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

Table 3: Optimality gaps for CG when we terminate it after two hours of run time and CPU seconds for DR to obtain the optimal solution.

to offer a product only when the remaining resource capacity exceeds the protection level for the product. For the network revenue management setting, we considered a deterministic linear program that is formulated under the assumption that the customer choices take on their expected values. The number of decision variables in this linear program grows exponentially with the number of products, but we showed how to reduce this linear program into an equivalent one with a much smaller size. Furthermore, we gave an algorithm to recover the optimal solution to the original linear program by using the optimal solution to the reduced linear program. Computational experiments indicates that the reduced linear program is remarkably effective, improving the solution times for the original linear program by factors exceeding 10,000.

There are a number of research directions for future work. First, it is interesting to incorporate constraints on the offered subsets of products. When there is limited space availability to display products, one may be interested in limiting the total number of offered products. Similarly, if each product occupies a different amount of space, then one may be interested in limiting the total amount of space consumed by the offered products. Second, the Markov chain choice model is rich in parameters and it can potentially capture a variety of customer choice patterns. A systematic study of when the Markov chain choice model is a better approach to capture customer choices is likely to enhance our understanding of this choice model.

References

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. (2013), 'Assortment optimization under variants of the nested logit model', *Operations Research* (to appear).

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

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

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

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

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

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

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

Li, G., Rusmevichientong, P. and Topaloglu, H. (2013), The *d*-level nested logit model: Assortment and price optimization problems, Technical report, Cornell University, School of Operations Research and Information Engineering. Available at
`http://legacy.orie.cornell.edu/~huseyin/publications/publications.html`.

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.

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

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

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.

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.

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.

A Appendix: Omitted Results

For $(\hat{x}, \hat{z}) \in \mathcal{H}$, we define $S_{\hat{x}} = \big\{j \in N : \hat{x}_j > 0\big\}$. In this section, our objective is to show that we have $\hat{z}_j \geq R_{j,S_{\hat{x}}}$ for all $j \in N$. This result is used in several places throughout Section 5. In the next lemma, we give a preliminary result that shows that if we offer a larger subset of products, then the probability that a customer considers purchasing a product during the course of her choice process and does not purchase this product becomes smaller. This result becomes useful when we show that $\hat{z}_j \geq R_{j,S_{\hat{x}}}$ for all $j \in N$ later in this section.

Lemma 9 *For $\hat{S} \subset S$, we have $R_{j,\hat{S}} \geq R_{j,S}$ for all $j \in N$.*

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

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

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

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

Lemma 10 *For $(\hat{x}, \hat{z}) \in \mathcal{H}$, define $S_{\hat{x}} = \big\{j \in N : \hat{x}_j > 0\big\}$. Then, we have $\hat{z}_j \geq R_{j,S_{\hat{x}}}$ for all $j \in N$.*

Proof. Since $(\hat{x}, \hat{z}) \in \mathcal{H}$, there exist extreme points $(x^1, z^1), \ldots, (x^K, z^K)$ of $\mathcal{H}$ and positive scalars $\gamma^1, \ldots, \gamma^K$ summing to one such that $\hat{x} = \sum_{k=1}^K \gamma^k\, x^k$ and $\hat{z} = \sum_{k=1}^K \gamma^k\, z^k$. Without loss of generality, we assume that the scalars $\gamma^1, \ldots, \gamma^K$ are strictly positive. Otherwise, we can drop the scalars that are equal to zero. In this case, if we define the subset $S^k = \{j \in N : x_j^k > 0\}$, then Lemma 1 implies that $P_{j,S^k} = x_j^k$ and $R_{j,S^k} = z_j^k$ for all $j \in N$. Thus, we have $\hat{x}_j = \sum_{k=1}^K \gamma^k \, x_j^k = \sum_{k=1}^K \gamma^k \, P_{j,S^k}$ and $\hat{z}_j = \sum_{k=1}^K \gamma^k \, z_j^k = \sum_{k=1}^K \gamma^k \, R_{j,S^k}$ for all $j \in N$. We consider one of the subsets $S^1, \ldots, S^K$. If $j \in S^k$, then we have $P_{j,S^k} > 0$, but since $\hat{x}_j = \sum_{k=1}^K \gamma^k \, P_{j,S^k}$, we obtain $\hat{x}_j > 0$. Therefore, having $j \in S^k$ implies that $\hat{x}_j > 0$, in which case, by the definition of $S_{\hat{x}}$, we obtain $j \in S_{\hat{x}}$. Thus, we have $S^k \subset S_{\hat{x}}$. In this case, since we have $S^k \subset S_{\hat{x}}$ for all $k = 1, \ldots, K$, by Lemma 9, it follows that $R_{j,S^k} \geq R_{j,S_{\hat{x}}}$ for all $j \in N$, $k = 1, \ldots, K$. Using the last inequality, we obtain $\hat{z}_j = \sum_{k=1}^K \gamma^k \, R_{j,S^k} \geq \sum_{k=1}^K \gamma^k \, R_{j,S_{\hat{x}}} = R_{j,S_{\hat{x}}}$ for all $j \in N$. □