Relating the Approximability of the Fixed Cost and Space Constrained Assortment Problems

May 27, 2018

Abstract

We study the classic assortment optimization problem in which a retailer seeks the revenue maximizing set of products to offer to each arriving customer. This paper relates two variants of this assortment problem: the space constrained assortment problem, in which the retailer has a limit on the total space of the offered assortment, and the fixed cost assortment problem, in which the retailer incurs a fixed cost for each offered product. In particular, we develop an approximation scheme for the space constrained problem for any random utility choice model that only relies on the ability to solve the corresponding fixed cost assortment problem. We then apply this technique to give a constant factor approximation scheme for the space constrained assortment problem under a general choice model for vertically differentiated products. Last, we present computational results to show the efficacy of this approach.

1 Introduction

Developing and analyzing models for how consumers make purchasing decisions is critical for any retailer. Customer choice models provide a means for the retailer to do exactly this. Specifically, these models allow the retailer to map any assortment of products to the probability that each product in the assortment is purchased. A key characteristic of these models is that they provide a framework for representing customer substitution behavior, the phenomenon when a customer settles for a suitable alternative when she does not find her target product available for purchase. Further, customer choice models are able to capture how various product features affect these substitution patterns and hence affect purchase probabilities. This latter point is a modeling nuance; there is no single way to model how consumers are influenced by the various features of each product. As a result, there are a variety of customer choice models, each of which models how the features of a product shape substitution patterns differently.

Once the retailer has selected a suitable customer choice model, she can then use this model to help make optimal stocking and inventory decisions. One of the most fundamental optimization problems retailers face has come to be known as the assortment optimization problem. In this problem, the retailer must decide which products to offer with the objective of maximizing the expected revenue from each arriving customer. The unconstrained version of this problem assumes that there are no restrictions on the set of products that the retailer can offer. This classic problem can be extended in two ways in an effort to model a more realistic retailing setting. First, a space constraint is often added, capturing the setting in which there is a physical space restriction (e.g.

shelf space) or limited online real estate (e.g. page limit). The second extension associates a fixed cost for offering each product, which can represent an ordering or stocking cost.

These additions to the assortment problem are usually imposed separately. For example, Rusmevichientong et al. (2010), Désir and Goyal (2014), and Davis et al. (2013) study variations of the space constrained assortment problem under the multinomial logit choice (MNL) model without a fixed cost for offering a product. On the other hand, Honhon et al. (2012) and Kunnumkal and de Albeniz (2017) study the fixed cost assortment problem under various choice models without any consideration of a space constraint. Further, the algorithmic ideas and techniques employed to tackle these two optimization problems are generally quite different. Nonetheless, the underlying structural similarity between the two problems is that both problems associate a "penalty" for each product that is included in the assortment. For the space constrained problem, this penalty is a decrease in the available shelf space for other products. For the fixed cost problem, this penalty is more straightforward, as the retailer incurs a cost for adding the given product to the assortment. Further, when the space constraint is dualized using Lagrangian multipliers, the resulting optimization problem for a fixed Lagrangian multiplier is exactly the fixed cost assortment problem. Building on this notion that the two assortment problems are seemingly related, we seek to develop a more technical connection between the two problems.

In this paper, we provide the first link between the approximability of the space constrained assortment problem and the fixed cost assortment problem. For any random utility maximization (RUM) choice model, we show that if one can solve the fixed cost assortment problem optimally, then this immediately leads to a polynomial time $\frac{1-\epsilon}{3}$-approximation for any $\epsilon > 0$ for the space constrained problem. We go on to show how these results and techniques lead to new results for the space constrained assortment problem under a classic model for vertically differentiated products.

Contributions. We begin by relating the approximability of the fixed cost and space constrained assortment problems. We show that if the fixed cost assortment can be solved optimally, then this solution approach can be used to derive a $\frac{1-\epsilon}{3}$-approximation for any $\epsilon > 0$ for the space constrained problem. Our approach has two parts and can be applied to any RUM choice model. Further, the approximation scheme we present is simple in nature and efficient to implement. In the first part, we assume that we have a solution to a relaxed version of the space constrained assortment problem in which the retailer is allowed to choose two assortments and randomize her offer decision between these two assortments. In this relaxed version, we only require that the space constraint holds in expectation. We show that given a solution to this relaxed problem it possible to find an assortment of products that is feasible to the original space constrained problem and garners an expected revenue of at least one-third of the objective value of the relaxed solution. The second part of our approach details how we find a solution to the aforementioned relaxed problem with objective value at least $1-\epsilon$ times the optimal expected revenue for the space constrained problem, for any $\epsilon > 0$. To find such an assortment, we rely on a black box algorithm for the fixed cost assortment problem. In the setting in which each product consumes a single unit of space, and hence the space constraint merely implies a limit on the total number of offered products, we are able to tighten the analysis of our approach.

We use the link that we develop to give a new result for the space constrained assortment problem under a general model for vertically differentiated products. The choice model that we introduce is a generalization of the classic linear utility model first seen in Mussa and Rosen (1978), where the random utility that an arriving customer associates with product i is given by $U_i(\theta) = \theta q_i - p_i$. In this expression, q_i and p_i are the quality and price of product i, respectively, and θ is a random variable with distribution $F(\theta)$ that represents how an arriving customer values quality over price when making a purchasing decision. We defer the details of this model to a later section but note that we refer to this more general choice model for vertically differentiated products as the sequential flips nonparametric choice model. We first show that the space constrained assortment problem under the sequential flips nonparametric choice model is NP-Hard. We then show there exists a simple dynamic program for the fixed cost assortment problem, immediately yielding a $\frac{1-\epsilon}{3}$-approximation for any $\epsilon > 0$ for the space constrained problem.

In Appendix A.1, we extend our general approximation scheme to settings where the fixed cost assortment problem can only be approximated. We show that if the fixed cost assortment can be approximated within a factor of $\delta < 1$, then this solution approach can be used to derive a $\frac{(1-\epsilon)\delta}{3}$-approximation for any $\epsilon > 0$ for the space constrained problem. In fact, our approach is valid even when a heuristic approach with no performance guarantee is applied to the fixed cost problem. In this case, we are still guaranteed to produce a feasible assortment for the space constrained problem, however the performance of this assortment will be dependent on how well the heuristic performs on the specific instances that are considered and hence does not come with a theoretical guarantee.

We follow our theoretical results with an extensive set of computational experiments, which study the efficacy of our two step approximation scheme for solving space constrained assortment problems. We apply our scheme to solve space constrained assortment problems under the MNL, Markov chain, and sequential flips choice models. We add the former two models to our study due to their ubiquity in the revenue management literature and their reputation as robust choice models. Further, in studying these three models, we are able to apply our approximation scheme in scenarios in which the corresponding fixed cost assortment problem can be approximated and solved optimally. In particular, for the MNL choice model, we use the $\frac{1}{2}$-approximation for the fixed cost problem that is developed in Kunnumkal and de Albeniz (2017). For the fixed cost problem under the sequential flips model, we use the optimal dynamic program that we develop in Section 2. Finally, to the best of our knowledge, there is no known approximation algorithm for the fixed cost problem under the Markov chain choice model, and hence we apply a local search heuristic. In each of these settings, we find that our two step approximation scheme performs quite well, often producing assortments that are near optimal.

The implications of our results extend beyond a theoretical link between the two assortment problems and a new means to solved constrained assortment problems. First, we show that our results immediately provide a constant factor bound on the optimal objective value of the dual problem that results from relaxing the knapsack constraint in the space constrained assortment through a Lagrangian multiplier. In this way, we give a worst case gap of the upper bound provided by this relaxation, which can be used to assess the performance of various heuristics for the space constrained problem. Dualizing complicated constraints in an effort to develop tractable upper bounds is a common approach applied to various revenue management problems. For example, Topaloglu (2009) and Feldman and Topaloglu (2015*a*) apply this approach in the context of the network revenue management problem and the assortment optimization problem under a mixture of multinomial logit models, respectively. Neither, however, is able to bound the ensuing duality gap as we do. In this way, we believe that our results represent an important step in better understanding the effectiveness of this approach.

Second, the algorithmic ideas presented in Section 2.1 prove useful for solving other assortment optimization problems. A critical component of the algorithm presented in Feldman (2017) for the space constrained assortment problem under the paired combinatorial logit model is the ability to derive a single feasible assortment from the relaxed space constrained problem described above, in which the retailer is allowed to choose two assortments and randomize her offer decision between these two assortments. Hence, while the techniques supplied in this paper are applied to general RUM choice models, there is the potential to focus these techniques towards specific choice models and obtain new results.

Literature Review. Assortment optimization has been thoroughly studied under many choice models. In a seminal paper, Talluri and van Ryzin (2004) solve the unconstrained assortment optimization problem when customers choose according to the MNL choice model. Extending this result, Rusmevichientong et al. (2010), Wang (2012), Davis et al. (2013), and Wang (2013) study various versions of the constrained assortment problems under the MNL model. Further, Kunnumkal and de Albeniz (2017) consider the fixed cost version of the assortment problem under the MNL choice model.

Many other parametric models have been introduced and studied in order to generalize the MNL model and better capture more complex substitution behavior. In particular, Méndez-Díaz et al. (2014), Désir and Goyal (2014), and Rusmevichientong et al. (2014) focus on assortment problems when customer choices are governed by a mixture of multinomial logit models. Li and Rusmevichientong (2014) and Davis et al. (2014) develop efficient solution methods for the unconstrained assortment problem when customers choose under the nested logit model. Gallego and Topaloglu (2014) and Feldman and Topaloglu (2015*b*) consider constrained versions of the assortment problem when customers choose according to the nested logit model. With regards to the recently introduced Markov chain choice model, Blanchet et al. (2016) and Désir et al. (2015) solve the unconstrained and constrained versions of the assortment problem, respectively. Finally, Jagabathula (2014) considers the efficacy of greedy algorithms under general RUM choice behavior when the retailer is given access to an oracle that can evaluate the revenue of any given assortment.

There is also fairly extensive literature on the nonparametric choice model. To our knowledge, the first to consider assortment optimization under the nonparametric choice model was Honhon et al. (2012), which considers a restricted set of potential preferences lists related to a tree structure of the products. The authors add fixed costs to the assortment problem, but they do not consider the addition of a space constraint. This result is extended in Paul et al. (2017), in which the authors work in a more general tree setting. Two other papers that study the assortment problem under the nonparametric choice model are Aouad, Farias, Levi and Segev (2015) and Aouad, Farias and Levi (2015). The former proves various hardness results related to the unconstrained assortment problem as well as a $\frac{1}{e \cdot d}$-approximation for the special case where the lengths of the preference lists are at most length d. The latter considers the unconstrained assortment optimization problem under the nonparametric choice model when customer preference lists are associated with structured set systems defined over a single overarching ordering of the products. Finally, van Ryzin and Vulcano (2017) and Farias et al. (2013) study the problem of estimating the underlying arrival probabilities of each customer class from historical sales data.

The work we present regarding the sequential flips nonparametric choice model is most closely related to the work in Pan and Honhon (2012). The authors of this paper study both pricing and assortment problems under the linear utilities choice model but do not consider the addition of a space constraint. The authors of this paper show that the pricing and assortment problems can be reduced to a tractable shortest path problem whose size scales nicely with the number of products. Our dynamic programming approach for the fixed cost assortment optimization problem under the sequential flips nonparametric builds on this approach by viewing this choice process through the lens of the nonparametric choice model. To the best of our knowledge, we are the first paper that considers the space constrained assortment problem under this model for vertically differentiated products.

1.1 Organization

The remainder of this paper is organized as follows. In Section 2, we formally define the space constrained and fixed cost assortment optimization problems and present our two part approximation scheme. In Section 3, we introduce the sequential flips model in relation to the linear utilities model and develop a simple dynamic program for the fixed cost assortment problem, allowing us to apply the approximation scheme from the preceding section to solve the space constrained assortment problem. We then test the efficacy of our algorithm through a series of computational experiments in Section 4. Last, we conclude and propose directions for future work in Section 5.

2 The Fixed Cost and Space Constrained Assortment Problems

In the assortment optimization problem, a firm chooses a subset of products $S \subseteq N$ to offer to customers with the intention of maximizing the expected revenue from an arriving customer when purchasing decisions are governed by a pre-specified customer choice model. For assortment $S \subseteq N$ and product $j \in S$, we let $\mathrm{Pr}_j(S)$ be the probability that product j is purchased when assortment S is offered under this choice model. We work under the assumption that the pre-specified choice model is a random utility maximizing (RUM) choice model, which includes the majority of popular choice models such as the MNL, nested logit, and Markov chain choice models among many others. We will make use of the following well known property of any RUM choice model, which states that when a product is added to an assortment, the purchase probabilities of other products in the assortment cannot increase. This property is often referred to as regularity or substitutability and extends beyond the family of RUM choice models.

Fact 2.1. *For any assortments* S_1, S_2 *such that* $S_1 \subseteq S_2$ *and* $j \in S_1 \cap S_2$ *we must have that* $\Pr_j(S_1) \geq \Pr_j(S_2)$.

Given these purchase probabilities, we denote the expected revenue of a given assortment $S \subseteq N$ as $\text{Rev}(S) = \sum_{j \in S} r_j \Pr_j(S)$. In the **space constrained assortment problem**, every product $i \in N$ also has a given space consumption $c_i \geq 0$, and there is a limit C on the total space consumption of an offered assortment. In this case, the goal of the retailer is to find the revenue maximizing assortment subject to this knapsack constraint on the space consumption of the offered assortment. This problem can be expressed as follows

$$\text{OPT}_c = \max_{S \subseteq N : C(S) \leq C} \text{Rev}(S), \tag{1}$$

where $C(S) = \sum_{i \in S} c_i$ is the space consumption of assortment $S \subseteq N$. In the case that $c_i = 1$ for all $i \in N$, the space constraint enforces that at most C products are included in the offered assortment. This version of the problem is often referred to as the **cardinality constrained assortment problem**.

In order to present our general approximation scheme for the space constrained assortment problem we also consider another variant of the assortment problem in which there is a fixed cost f_i for offering product $i \in N$, representing an ordering or stocking cost, and the retailer would like to maximize expected revenue minus these offering costs. In this case, the **fixed cost assortment** problem can be expressed as

$$\text{OPT}_f = \max_{S \subseteq N} \text{Rev}(S) - \sum_{i \in S} f_i. \tag{2}$$

In what follows, we provide the first link between the approximability of problems (1) and (2). In particular, we assume that we can solve the fixed cost assortment problem and then use this algorithm as a black box to develop an efficient constant factor approximation algorithm to the space constrained problem.

2.1 Solving the Space Constrained Assortment Problem

Our approach unfolds in two steps. We begin by assuming that we can find a solution to a modified instance of the space constrained assortment problem that has objective function value $\mathcal{R}$. In particular, we assume that we can find two assortments S_1 and S_2 along with a weight $0 \leq \alpha \leq 1$ such that

$$\alpha \text{Rev}(S_1) + (1-\alpha)\text{Rev}(S_2) \geq \mathcal{R}$$

and

$$\alpha C(S_1) + (1-\alpha)C(S_2) \leq C.$$

In other words, if the retailer offers S_1 with probability α and S_2 with probability $1-\alpha$, then she obtains an expected revenue at least $\mathcal{R}$ in expectation and also satisfies the space constraint in expectation. In the first step, we show how to derive an assortment $\bar{S}$ from S_1 and S_2 with expected revenue at least $\mathcal{R}/3$ and space consumption at most C. We do so by considering the linear programming relaxation of a knapsack problem. Next, we show how to find S_1, S_2, and α satisfying $\alpha\text{Rev}(S_1)+(1-\alpha)\text{Rev}(S_2) \geq (1-\epsilon)\text{OPT}_c$ and $\alpha C(S_1)+(1-\alpha)C(S_2) \leq C$. Our approach for this latter step considers the Lagrangian relaxation of the space constrained problem, which turns out to be exactly the fixed cost version of the assortment problem. The critical step in our approach is choosing the appropriate Lagrangian multiplier in order to produce two assortments of the nature described above. Combining the two steps yields an assortment $\bar{S}$ that satisfies $\text{Rev}(\bar{S}) \geq \frac{(1-\epsilon)}{3}\text{OPT}_c$. In the setting in which $c_i = 1$ for all $i \in N$, we show that our analysis of this two step approach can be tightened to show that $\text{Rev}(\bar{S}) \geq \frac{(1-\epsilon)}{2}\text{OPT}_c$.

We proceed by detailing our approach for step one. Suppose that we have found S_1, S_2, and α as described above. We assume without loss of generality that $C(S_1) \leq C(S_2)$ and hence the assortment S_1 is feasible for the space constrained problem. Thus, we have found one initial feasible solution with expected revenue $\text{Rev}(S_1) \geq \alpha\text{Rev}(S_1)$. Next, we show how to find an assortment $\hat{S}$ such that $\text{Rev}(\hat{S}) \geq (1-\alpha)\text{Rev}(S_2)/2$. Consider the following linear programming relaxation of a constructed knapsack problem in which the items are the products of S_2.

$$\begin{aligned} Z^*_{knap} = \max \ & \sum_{j \in S_2} r_j \text{Pr}_j(S_2) x_j \\ \text{s.t.} \ & \sum_{j \in S_2} c_j x_j \leq C \\ & 0 \leq x_i \leq 1. \end{aligned} \tag{KNAP}$$

In this knapsack problem, the value of item $j \in S_2$ is $r_j\text{Pr}_j(S_2)$ and its space consumption is c_j. The following proposition shows that we can construct a feasible solution to Problem (KNAP) that has objective function value $(1-\alpha)\text{Rev}(S_2)$.

Lemma 2.2. *The solution $\tilde{x}_j = 1 - \alpha$ for all $j \in S_2$ is a feasible solution to the knapsack linear program and has objective function value equal to $(1-\alpha)\text{Rev}(S_2)$.*

Proof. First we show that the proposed solution is feasible by considering its space consumption.

$$\sum_{j \in S_2} c_j \tilde{x}_j = (1-\alpha) \sum_{j \in S_2} c_j = (1-\alpha) C(S_2) \leq C.$$

The inequality follows from the properties of S_1 and S_2. Thus, $\tilde{x}_j$ is a feasible solution to the knapsack linear program. Further, its objective function value is

$$\sum_{j \in S_2} r_j \text{Pr}_j(S_2) \tilde{x}_j = (1-\alpha) \sum_{j \in S_2} r_j \text{Pr}_j(S_2) = (1-\alpha)\text{Rev}(S_2).$$

□

Lemma 2.2 shows that the optimal objective function value of the knapsack linear program is at least $(1-\alpha)\text{Rev}(S_2)$. Further, it is well known that the optimal solution to the linear programming relaxation of any knapsack problem has at most one fractional variable value. Let x^* be an optimal solution to the linear programming relaxation, and let $\hat{S}_1 = \{j \in S_2 : x_j^* = 1\}$ and $\hat{S}_2 = \{j \in S_2 : 0 < x_j^* < 1\}$. Note that $|\hat{S}_2| \leq 1$. The following lemma bounds the revenue of the best of these two solutions.

Lemma 2.3. *Let $\hat{S} = \text{argmax}_{S \in \{\hat{S}_1, \hat{S}_2\}} \text{Rev}(S)$. Then, $C(\hat{S}) \leq C$ and*

$$\text{Rev}(\hat{S}) \geq \frac{1}{2}(1-\alpha)\text{Rev}(S_2).$$

Further, if $c_i = 1$ for all $i \in N$, then $\text{Rev}(\hat{S}) \geq (1-\alpha)\text{Rev}(S_2)$.

Proof. By the feasibility of x^*, $C(\hat{S}_1) \leq C$. Further, without loss of generality, every product has a space consumption of at most C. Therefore, $C(\hat{S}_2) \leq C$ as well, and so $C(\hat{S}) \leq C$. Next, we consider the revenue of assortment $\hat{S}$.

$$\begin{aligned}
\text{Rev}(\hat{S}) &= \max\{\text{Rev}(\hat{S}_1), \text{Rev}(\hat{S}_2)\} \\
&\geq \max\left[\sum_{j \in \hat{S}_1} r_j \text{Pr}_j(S_2), \sum_{j \in \hat{S}_2} r_j \text{Pr}_j(S_2)\right] \\
&\geq \frac{1}{2}\left[\sum_{j \in S_2} r_j \text{Pr}_j(S_2) x_j^*\right] \\
&\geq \frac{1}{2}(1-\alpha)\text{Rev}(S_2).
\end{aligned}$$

The first inequality comes from the fact that $\hat{S}_1, \hat{S}_2 \subseteq S_2$ and hence we can apply Fact 2.1. The second inequality results from the fact that

$$\sum_{j \in \hat{S}_1} r_j \Pr_j(S_2) + \sum_{j \in \hat{S}_2} r_j \Pr_j(S_2) \geq \sum_{j \in S_2} r_j \Pr_j(S_2) x_j^*.$$

Finally, the last inequality follows from Lemma 2.2.

In the case that $c_i = 1$ for all $i \in N$, we may assume that C is an integer value. In this case, the optimal solution to the knapsack problem is integer valued and so $\hat{S}_2 = \emptyset$. This immediately implies that $\text{Rev}(\hat{S}_1) = Z^*_{knap} \geq (1-\alpha)\text{Rev}(S_2)$. □

At this point, we consider either offering the assortment S_1 or $\hat{S}$. Let $\bar{S} = \text{argmax}_{S \in \{S_1, \hat{S}\}} \text{Rev}(S)$. The following proposition shows that returning the best of these two assortments produces an assortment $\bar{S}$ with revenue $\text{Rev}(\bar{S}) \geq \mathcal{R}/3$.

Proposition 2.4.

$$\text{Rev}(\bar{S}) \geq \mathcal{R}/3.$$

Further, if $c_i = 1$ for all $i \in N$, then $\text{Rev}(\bar{S}) \geq \mathcal{R}/2$.

Proof. Applying Lemma 2.3, $\text{Rev}(\bar{S}) = \max\{\text{Rev}(S_1), \text{Rev}(\hat{S})\} \geq \max\{\alpha\text{Rev}(S_1), \frac{1}{2}(1-\alpha)\text{Rev}(S_2)\}$. Recall that $\alpha\text{Rev}(S_1) + (1-\alpha)\text{Rev}(S_2) \geq \mathcal{R}$. Therefore, if $\alpha\text{Rev}(S_1) \geq \frac{1}{3}\mathcal{R}$, the result holds. Otherwise, $(1-\alpha)\text{Rev}(S_2) \geq \frac{2}{3}\mathcal{R}$, and again, the result holds. In the case that $c_i = 1$ for all $i \in N$, Lemma 2.3 implies that $\text{Rev}(\bar{S}) \geq \max\{\alpha\text{Rev}(S_1), (1-\alpha)\text{Rev}(S_2)\} \geq \frac{1}{2}\mathcal{R}$. □

2.2 Finding a Convex Combination of Assortments

Finally, we show how to find S_1, S_2, and α such that $\alpha C(S_1) + (1-\alpha)C(S_2) \leq C$ and $\alpha\text{Rev}(S_1) + (1-\alpha)\text{Rev}(S_2) \geq (1-\epsilon)\text{OPT}_c$ for any $\epsilon > 0$, yielding a $\frac{1-\epsilon}{3}$-approximation algorithm for the space constrained assortment problem. First, we assume that for a given customer choice model we have access to a polynomial-time algorithm for the fixed cost assortment optimization problem. We extend this result to the case when we have access to a δ-approximation algorithm for the fixed cost assortment optimization problem, where $0 \leq \delta \leq 1$, in Appendix A.1. We consider the Lagrangian relaxation of the space constrained assortment optimization problem in which we associate a Lagrangian multiplier $\lambda \geq 0$ with the space constraint. This problem is formulated as

$$\text{OPT}_\lambda = \max_{S \subseteq N} \text{Rev}(S) + \lambda(C - C(S)) = \max_{S \subseteq N} \text{Rev}(S) - \lambda C(S) + \lambda C. \quad (3)$$

First, note that for fixed $\lambda \geq 0$, this problem is equivalent to the fixed cost assortment optimization problem with $f_i = \lambda c_i$, and hence, we can find an assortment S_λ in polynomial time such that

$$\text{Rev}(S_\lambda) - \lambda C(S_\lambda) + \lambda C = \text{OPT}_\lambda.$$

Note that for any $\lambda \geq 0$, $\text{OPT}_\lambda \geq \text{OPT}_c$ since the optimal solution to the space constrained assortment optimization problem is also a feasible solution to problem (3). Suppose that we can find $\lambda^* \geq 0$ such that $C(S_{\lambda^*}) = C$. Then,

$$\text{Rev}(S_{\lambda^*}) = \text{Rev}(S_{\lambda^*}) - \lambda^* C(S_{\lambda^*}) + \lambda^* C = \text{OPT}_{\lambda^*} \geq \text{OPT}_c.$$

This also implies that $\text{OPT}_{\lambda^*} = \text{OPT}_c$, since we also know that $\text{Rev}(S_{\lambda^*}) \leq \text{OPT}_c$. We show that we either we can find λ^* or we can find assortments S_{λ_1} and S_{λ_2} along with a carefully chosen α such that $\alpha \text{Rev}(S_{\lambda_1}) + (1-\alpha) R(S_{\lambda_2}) \geq (1-\epsilon)\text{OPT}_c$ for any $\epsilon > 0$.

We find S_{λ_1} and S_{λ_2} through bisection search on the interval $[0, r_{\max}/c_{\min}]$, where $r_{\max} = \max_{i \in N} r_i$ and $c_{\min} = \min_{i \in N} c_i$. Initially, we set $\lambda_1 = 0$ and $\lambda_2 = r_{\max}/c_{\min}$. Note that for $\lambda = \lambda_1$ the problem reduces to the unconstrained assortment problem and $C(S_{\lambda_1}) > C$. On the other hand, for $\lambda = \lambda_2$ the penalty for offering any subset is always at least the revenue generated by that subset and so $S_{\lambda_2} = \emptyset$. Consequently, $C(S_{\lambda_2}) < C < C(S_{\lambda_1})$. Given λ_1 and λ_2, in each iteration the algorithm tests the midpoint $\lambda' = \frac{1}{2}(\lambda_1 + \lambda_2)$. If $C(S_{\lambda'}) = C$, then we return $S_{\lambda'}$. Otherwise, if $C(S_{\lambda'}) > C$, then we set $\lambda_1 = \lambda'$, and if $C(S_{\lambda'}) < C$, then we set $\lambda_2 = \lambda'$. We repeat this process until either we find an assortment with a space consumption of exactly C or until $\lambda_2 - \lambda_1 \leq \epsilon \cdot \text{Rev}_{\min}/\mathcal{C}$, where $\text{Rev}_{\min} = \max_{i \in N} \text{Rev}(\{i\})$ and $\mathcal{C} = \sum_{i \in N} c_i$. If $\text{Rev}_{min} = 0$, then we must have $\text{OPT}_c = 0$ and hence it is optimal to offer nothing. Otherwise, we know that $\text{Rev}_{min} \leq \text{OPT}_c$ since every product is itself feasible. We note that this implies that there are at most $O(\log \frac{\mathcal{C} \cdot r_{\max}}{\epsilon \text{Rev}_{\min} \cdot c_{\min}})$ calls to the fixed cost assortment problem, which is polynomial in the input size.

At the end of this bisection procedure, let $S_1 = S_{\lambda_1}$ and $S_2 = S_{\lambda_2}$ and set

$$\alpha = \frac{C - C(S_2)}{C(S_1) - C(S_2)}$$

which implies

$$1 - \alpha = \frac{C(S_1) - C}{C(S_1) - C(S_2)}.$$

By construction of S_1 and S_2, we know that $0 \leq \alpha \leq 1$. We now prove that S_1, S_2, and α satisfy the properties we are looking for in order to find $\bar{S}$.

Proposition 2.5. *Given assortments S_1 and S_2 and weight α, we have that $\alpha\mathrm{Rev}(S_1) + (1 - \alpha)\mathrm{Rev}(S_2) \geq (1-\epsilon)\mathrm{OPT}_c$ and $\alpha C(S_1) + (1-\alpha)C(S_2) \leq C$.*

Proof. First, we note that

$$\alpha C(S_1) + (1-\alpha)C(S_2) = \frac{C - C(S_2)}{C(S_1) - C(S_2)}C(S_1) + \frac{C(S_1) - C}{C(S_1) - C(S_2)}C(S_2) = C.$$

Second, we analyze the revenue of the convex combination.

$$\begin{aligned}
\mathrm{OPT}_c &\leq \alpha\mathrm{OPT}_{\lambda_1} + (1-\alpha)\mathrm{OPT}_{\lambda_2} \\
&= \alpha\mathrm{Rev}(S_1) + (1-\alpha)\mathrm{Rev}(S_2) + \alpha\lambda_1[C - C(S_1)] + (1-\alpha)\lambda_2[C - C(S_2)] \\
&= \alpha\mathrm{Rev}(S_1) + (1-\alpha)\mathrm{Rev}(S_2) + \lambda_2\left(\alpha[C - C(S_1)] + (1-\alpha)[C - C(S_2)]\right) \\
&- (\lambda_2 - \lambda_1)\alpha(C - C(S_1)) \\
&= \alpha\mathrm{Rev}(S_1) + (1-\alpha)\mathrm{Rev}(S_2) - (\lambda_2 - \lambda_1)\alpha(C - C(S_1)) \\
&\leq \alpha\mathrm{Rev}(S_1) + (1-\alpha)\mathrm{Rev}(S_2) + (\lambda_2 - \lambda_1)\mathcal{C} \\
&\leq \alpha\mathrm{Rev}(S_1) + (1-\alpha)\mathrm{Rev}(S_2) + \epsilon\mathrm{Rev}_{\min}
\end{aligned}$$

Subtracting the $\epsilon\mathrm{Rev}_{\min}$ term over and recalling that $\mathrm{Rev}_{min} \leq \mathrm{OPT}_c$ gives the desired result. The second to last inequality follows from our stopping criterion on the difference between λ_2 and λ_1. □

Combining the results from both steps yields the following theorem.

Theorem 2.6. *If there exists a polynomial-time algorithm for the fixed cost assortment optimization problem under any RUM choice model, then there exists a $\frac{1-\epsilon}{3}$-approximation algorithm for the corresponding space constrained assortment optimization problem and a $\frac{(1-\epsilon)}{2}$-approximation algorithm for the cardinality constrained assortment optimization problem.*

Proof. Combining Proposition 2.4 and 2.5 gives the result. □

An interesting consequence of the above approach is that it allows us to bound the duality gap when we associate a Lagrangian multiplier with the knapsack constraint in the space constrained assortment problem. This relaxed problem is given in equation (3). Let $\hat{\lambda} = \mathrm{argmin}_{\lambda \geq 0}\,\mathrm{OPT}_\lambda$. The following theorem relates the optimal dual objective $\mathrm{OPT}_{\hat{\lambda}}$ to OPT_c.

Theorem 2.7. *For any RUM choice model, we have that $\mathrm{OPT}_{\hat{\lambda}} \leq (3+\epsilon)\mathrm{OPT}_c$, for any $\epsilon > 0$.*

Proof. We have that $\text{OPT}_{\hat{\lambda}} \leq \alpha\text{OPT}_{\lambda_1} + (1-\alpha)\text{OPT}_{\lambda_2} \leq \alpha\text{Rev}(S_1) + (1-\alpha)\text{Rev}(S_2) + \epsilon\text{Rev}_{\min} \leq (3+\epsilon)\text{OPT}_c$, as desired. The first inequality follows because $\text{OPT}_{\hat{\lambda}} \leq \text{OPT}_{\lambda_1}, \text{OPT}_{\lambda_2}$ by definition of $\hat{\lambda}$. The last inequality follows because we have shown above that $\alpha\text{Rev}(S_1) + (1-\alpha)\text{Rev}(S_2) \leq 3\text{OPT}_c$. To see this, note that S_1 is feasible to the space constrained assortment problem and hence $\alpha\text{Rev}(S_1) \leq \text{OPT}_c$. Further, in Lemma 2.3, we show that $\text{Rev}(\hat{S}) \geq \frac{1}{2}(1-\alpha)\text{Rev}(S_2)$, and since $\hat{S}$ is a feasible assortment, we have that $(1-\alpha)\text{Rev}(S_2) \leq 2\text{OPT}_c$. $\square$

In the next section, we use the results and techniques developed in this section to establish new results for the space constrained assortment problem under a model for vertically differentiated products.

3 Sequential Flips Nonparametric Choice Model

In this section, we introduce and study a choice model which generalizes a classical model for vertically differentiated products. For this model, solving the space constrained problem directly is difficult. Instead, we show that the fixed cost assortment problem can be solved optimally via a relatively straightforward dynamic programming approach. Using the two step approach that is detailed in Section 2.1, this result immediately yields a constant factor approximation for the space constrained version of the problem.

We now formally introduce the well-studied model of Mussa and Rosen (1978), which captures the behavior of customers purchasing from a set of vertically differentiated products. In this model, the random utility that an arriving customer associates with each product is assumed to be a linear function of the product's quality and price. Consequently, we refer to this model as the linear utility choice model. More formally, given a set $N = \{1, \ldots, n\}$ of vertically differentiated products we let the quality and price of each product $i \in N$ be given by q_i and p_i, respectively. An arriving customer associates utility $U_i(\theta) = \theta q_i - p_i$ with each product $i \in N$, where θ is a random variable with arbitrary distribution $F(\theta)$. This random variable reflects the extent to which arriving customers value the quality of the product. The no-purchase option, also referred to as product 0, is assumed to have utility $U_0(\theta) = 0$. We assume customers are utility maximizing and thus will purchase the offered product with the highest positive utility. In particular, if the retailer offers assortment $S \subseteq N$, then we can express the probability that an arriving customer purchases product $j \in S$ as $\text{Pr}_j(S) = \text{Pr}(U_j(\theta) = \max_{i \in S\cup\{0\}} U_i(\theta))$.

The sequential flips nonparametric model is inspired by considering the linear utility model from a nonparametric perspective. We refer to our new choice model as the sequential flips nonparametric choice model since it is a special case of the general nonparametric choice model. In the general nonparametric choice model, we are given a set of customer classes $\mathcal{G} = \{1, \ldots, m\}$, where each customer class $g \in \mathcal{G}$ arrives with probability λ_g and is associated with an arbitrary ranking σ_g on $N+ = N \cup \{0\}$, the set of all products and the no-purchase option. For $i \in N+$, let $\sigma_g(i)$ give the index of product i in customer class g's preference list. We use the convention that lower indexed products have a higher ranking. So a product that is first in a customer's preference list has the highest ranking. An arriving customer will purchase her highest ranked offered item, which could potentially be the ever-present no-purchase option. In particular, if the retailer offers an assortment $S \subseteq N$, then a customer of class g will purchase product

$$\pi_g(S) := \operatorname*{argmin}_{i \in S \cup \{0\}} \sigma_g(i).$$

For an offered assortment $S \subseteq N$, we can express the probability that product j is purchased as $\mathrm{Pr}_j(S) = \sum_{g \in \mathcal{G}: \pi_g(S) = j} \lambda_g$.

Under the sequential flips nonparametric choice model, the set of customer classes can be indexed such that the preference list for customer class $g+1$ can be derived from the preference list for customer class g by merely swapping the ordering of two adjacent products in customer class g's preference list. More formally, for customer class $g \in \mathcal{G}$, let products l_g and k_g satisfy $\sigma_g(l_g) = \sigma_g(k_g) - 1$ and assume that these are the two products that swap places to produce the preference list of customer class $g+1$. Then, the preference list of customer class $g+1$ can be written as

$$\sigma_{g+1}(i) = \begin{cases} \sigma_g(i) & \text{if } i \notin \{l_g, k_g\} \\ \sigma_g(l_g) & \text{if } i = k_g \\ \sigma_g(k_g) & \text{if } i = l_g \end{cases}.$$

For the remainder of this paper, we assume that the customer classes are indexed as described above.

The second restriction we impose on the set of preference lists is that once one product swaps ahead of another product in a preference list, it can never swap back. In other words, for any pair of products $j, k \in N$, let $G(j,k) = \{g \in \mathcal{G} : \sigma_g(j) < \sigma_g(k)\}$ be the set of customer classes for which product j is preferred to product k. This restriction implies that $G(j,k)$ is a consecutive set of customer classes. Note that with this restriction in place, it is not hard to see that $m = O(n^2)$. Lastly, for ease of presentation, we assume that the no-purchase option never swaps above another product. This implies that if a customer of class g makes a purchase, then all customers of class

$g' > g$ will also make a purchase. All results continue to hold without this last restriction.

3.1 Relationship to Linear Utility Model

Before considering the fixed cost assortment problem, we first prove that the sequential flips nonparametric choice model generalizes the linear utility choice model. Given any arbitrary linear utility choice model on n products with prices $p = \{p_1, \dots, p_n\}$ and qualities $q = \{q_1, \dots, q_n\}$, we define $\Theta(p, q) = \{\theta_1, \theta_2, \dots, \theta_m, \theta_{m+1}\}$ to be the ordered values of $\theta > 0$ for which at least two products have the same utility, meaning there exists $i, j \in N^+$ such that $\theta q_i - p_i = \theta q_j - p_j$. We also set $\theta_1 = F^{-1}(0)$ and $\theta_{m+1} = F^{-1}(1)$ to represent the endpoints of the distribution of θ, possibly equal to $-\infty$ or ∞. Since the utilities are linear functions of θ, $m = O(n^2)$ since n lines have at most $O(n^2)$ intersection points. Consider the following set of intervals $\mathcal{I} = \{I_1, \dots, I_m\}$, where $I_l = [\theta_l, \theta_{l+1}]$ for all $1 \leq l \leq m$. Note that, by construction, any customer that arrives with $\theta \in I_l$ will have the same ordering of product utilities. Let ρ_l give this ordering of the product utilities when $\theta \in I_l$, and let $\rho_l(i)$ give the index of product i in this ordering. Again, we use the convention that a lower index in this ordering implies a higher utility. See Figure 1 for a full visualization of this procedure. For a given linear utility choice model, the following proposition shows how to construct an equivalent sequential flips nonparametric choice model.

Proposition 3.1. *For an arbitrary linear utility choice model with utility intersection points given by $\Theta(p, q) = \{\theta_1, \theta_2, \dots, \theta_m, \theta_{m+1}\}$, we can construct an equivalent sequential flips nonparametric choice model with $\mathcal{G} = \{1, 2, \dots, m\}$ by setting the arrival probability of each customer class $l \in \mathcal{G}$ is $\lambda_l = P(\theta \in I_l)$ and the preference list for customer class l is created by setting $\sigma_l(i) = \rho_l(i)$ for all $i \in N$.*

Proof. Under this sequential flips nonparametric choice model, let $\hat{\Pr}_j(S)$ be the probability that product j is purchased when assortment $S \subseteq N$ is offered. Then,

$$
\begin{aligned}
\hat{\Pr}_j(S) &= \sum_{l \in \mathcal{G}: \pi_l(S) = j} \lambda_l \\
&= \sum_{l=0}^{m} \mathbb{1}_{j = \text{argmin}_{i \in S \cup \{0\}} \sigma_l(i)} \Pr(\theta \in I_l) \\
&= \sum_{l=0}^{m} \mathbb{1}_{j = \text{argmax}_{i \in S \cup \{0\}: \theta \in I_l} U_i(\theta)} \Pr(\theta \in I_l) \\
&= \Pr(U_j = \max_{i \in S \cup \{0\}} U_i(\theta))
\end{aligned}
$$

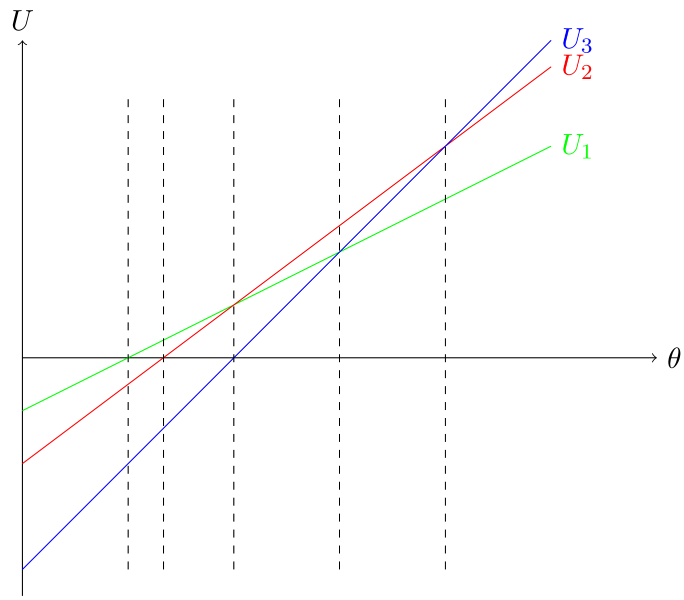

Figure 1: Visualization of the transformation from the linear utility model to the sequential flips nonparametric choice model for a 3 product example with 5 customer classes. Note that we have $\sigma_1 = [0,1,2,3], \sigma_2 = [1,0,2,3], \sigma_3 = [1,2,0,3], \sigma_4 = [2,1,0,3], \sigma_5 = [2,1,3,0], \sigma_6 = [2,3,1,0], \sigma_7 = [3,2,1,0]$.

as desired. The second equality follows from the definition of λ_l, and the third equality follows since the ordering of the products in customer class l's preference list is the same as the ordering of the utilities in I_l. $\square$

Advantages of the sequential flips model. We contend that the sequential flips nonparametric choice model should be viewed as an improved way to represent the linear utility model. First, consider the process of estimating the underlying parameters of each model using historical data. In the case of the linear utility model, this task is near impossible without the transformation to the sequential flips model that we propose. To see this, consider the maximum likelihood problem that results when one attempts to estimate the parameters of the linear utility model from historical sales data. Even when the distribution $F(\theta)$ with support $[0, \bar{\theta}]$ is given, it is still difficult to write the likelihood for a single purchase event as a function of the qualities of each product. Namely, consider the event that assortment S is offered and product $i \in S$ is purchased. In this case, the likelihood is simply the purchase probability $\Pr_i(S) = \int_0^{\bar{\theta}} \mathbb{1}\{q_i\theta - p_i \geq \max_j q_j\theta - p_j\} d\theta$. The lack of a closed form expression for $\Pr_i(S)$ shows that there is little hope for a tractable MLE approach to estimate the qualities of each product. Pan and Honhon (2012) fit linear utility models to iPhone sales data, but because of the inherent difficulty just discussed, they revert to what essentially amounts to guessing the distribution $F(\theta)$ and the qualities of each product.

On the other hand, in estimating the sequential flips nonparametric choice model, there are many settings that involve vertically differentiated products where we can assume the qualities of each product are given, and then use Proposition 3.1 to derive the set of preference lists. In the case of hotels or restaurants, one could simply use the published star rating as the qualities. In the second case study presented in Pan and Honhon (2012), where the authors fit a linear utility model to USB stick sales data, the authors simply take the memory of each different stick to be the quality. Given the set of preference lists, we can then use MLE to find the corresponding arrival probabilities of each customer type, which is akin to estimating the distribution $F(\theta)$. In this case, it is well known (see van Ryzin and Vulcano (2017)) that the MLE problem is concave in the arrival probabilities.

In Appendix B, we provide a simple set of computational experiments that illustrates this advantage. In this study, we find significant improvements in the accuracy of our fitted models when we view the linear utility model through the lens of the sequential flips nonparametric choice model. Since the focus of this paper is on developing tractable algorithms for assortment problems, we leave a more extensive set of experiments for future work.

A second advantage of casting the linear utility model as a sequential flips nonparametric choice model is that this perspective allows us to develop a simple dynamic program for the fixed cost assortment problem. Our dynamic programming approach revolves around a simple recursive way to compute the expected revenue of any assortment by exploiting the relationship between the preference lists of adjacent customer types in the sequential flips model. We detail this approach in the next section.

3.2 The Fixed Cost Assortment Problem

In this section, we study the fixed cost assortment problem given in (2) under the sequential flips nonparametric choice model. The authors of Pan and Honhon (2012) show that the fixed cost assortment problem under the linear utility choice model can be recast as a shortest path problem. We boil down the insights used to create the shortest path problem into a simple dynamic programming framework, which allows us to solve the fixed cost assortment problem for the more general sequential flips nonparametric choice model with a runtime that improves upon the approach of Pan and Honhon (2012) by a factor of n. This result immediately implies a a $\frac{1-\epsilon}{3}$-approximation for the space constrained version of the assortment problem, which we show to be NP-Hard in Appendix A.2.

Before we present the dynamic program, we derive an efficient way to compute the product that a particular customer class will purchase. For any assortment $S \subseteq N$, we first note that the product that customer class g purchases can be derived only from two pieces of information: the product purchased by customer class $g-1$ and whether or not product k_g is offered. Recall that k_g is the product that flipped above product l_g to form customer class g. Therefore, if customer class $g-1$ purchases product $j \neq l_g$, then we know that customer class g must also purchase product j. On the other hand, if customer class $g-1$ purchases product $j = l_g$, then customer class g will purchase product k_g if $k_g \in S$ and product j otherwise. In other words, for assortment $S \subseteq N$ we have that

$$\pi_g(S) = \begin{cases} \pi_{g-1}(S), & \text{if } k_g \notin S \text{ or } \pi_{g-1}(S) \neq l_g \\ k_g & \text{if } k_g \in S \text{ and } \pi_{g-1}(S) = l_g. \end{cases} . \tag{4}$$

Building on this observation, it becomes fairly straightforward to develop a simple recursion for computing the revenue of any assortment. Let $R(S,g)$ be the revenue accrued from customer classes $g, g+1, \ldots, m$ from assortment $S \subseteq N$. For any assortment $S \subseteq N$ and customer class $g \in \mathcal{G}$, we can compute $R(S,g)$ through the following recursion:

$$R(S,g) = \begin{cases} \lambda_1 r_{\pi_1(S)} - f_{\pi_1(S)} + R(S,2) & g = 1 \\ \lambda_g r_{\pi_{g-1}(S)} + R(S,g+1) & \text{if } k_g \notin S \text{ or } \pi_{g-1}(S) \neq l_g \\ \lambda_g r_{k_g} - f_{k_g} + R(S,g+1) & \text{if } k_g \in S \text{ and } \pi_{g-1}(S) = l_g \end{cases}$$

with base case

$$R(S, m+1) = 0.$$

We can rewrite this recursion more concisely by aggregating consecutive customer classes that purchase the same product. To do so, we introduce the following two definitions for $g > 1$. Let

$$O(g) := \min\{g' > g : l_{g'} = k_g\}$$
$$N(g) := \min\{g' > g : l_{g'} = l_g\}.$$

For the case where there does not exists $g' > g$ such that $l_{g'} = k_g$, we set $O(g) = m+1$. In the analogous setting for $N(g)$, we also set $N(g) = m+1$. There are a two important insights about $O(g)$ and $N(g)$. First, we have that $k_g = l_{O(g)}$ and $l_g = l_{N(g)}$, by construction. Further, note that if customer class g purchases product k_g, so will all customer classes indexed $g+1, \ldots, O(g)-1$. On the other hand, if customer class g purchases product l_g, we know that customer classes indexed

$g+1,\ldots,N(g)-1$ must also purchase l_g. For $g=1$, we develop a slightly modified version of $O(g)$ that is a function of the product purchased by the first customer class. Let N_1 be all products j such that $\sigma_1(j) \le \sigma_j(0)$. This is the set of all products, including the no-purchase option, that the first customer class could have purchased. Then, for $j \in N_1$, we define

$$O(1,j) := \min\{g' > 1 : l_{g'} = j\}.$$

Again, if there does not exist customer class $g' > 1$ such that $l_{g'} = j$, then we set $O(1,j) = m+1$.

Building on this notation, we also find associated probabilities for these intervals in which customers purchase the same product. For each customer class g, we define

$$Q^o(g) := \sum_{q=g}^{O(g)-1} \lambda_q \quad \text{and} \quad Q^n(g) := \sum_{q=g}^{N(g)-1} \lambda_q,$$

and for each product $j \in N_1$, we define

$$Q^1(j) := \sum_{q=1}^{O(1,j)-1} \lambda_q.$$

These terms are useful for developing concise notation when computing expected revenues of assortments. Specifically, for any assortment $S \subseteq N$ and any customer class $g \in \mathcal{G}$ such that $\pi_{g-1}(S) = l_g$ we have that

$$R(S,g) = \begin{cases} r_{\pi_1(S)} Q^1(\pi_1(S)) - f_{\pi_1(S)} + R(S, O(1, \pi_1(S))) & g=1 \\ r_{l_g} Q^n(g) + R(S, N(g)), & \text{if } k_g \notin S \\ r_{k_g} Q^o(g) - f_{k_g} + R(S, O(g)) & \text{if } k_g \in S \end{cases} \tag{5}$$

with base case

$$R(S, m+1) = 0.$$

This more concise recursion is critical in proving the correctness of the dynamic program that we develop next. The value function $V(g)$ in our dynamic program represents the maximum expected revenue accrued from customer classes $g, g+1, \ldots, m$ when customer class $g-1$ purchases product l_g. The dynamic program is given below.

$$V(g) = \begin{cases} \max\{r_{k_g} Q^o(g) - f_{k_g} + V(O(g)), r_{l_g} Q^n(g) + V(N(g))\} & g > 1 \\ \max_{j \in N_1} \{r_j Q^1(j) - f_j + V(O(1,j))\} & g = 1 \end{cases} \tag{6}$$

with base case

$$V(m+1) = 0.$$

Note that $V(1)$ will be the maximum expected revenue from all customer classes and the maximization over $j \in N_1$ represents which product a customer of class $g = 1$ purchases (possibly equal to the no-purchase option). For $g > 1$, the first case of the maximum given in (6) represents choosing to offer product k_g, which costs f_{k_g} and implies that customer classes $g, g+1, \ldots, O(g)-1$ also purchase this product. The second case represents not offering product k_g and hence customer classes $g, g+1, \ldots, N(g)$ purchase product l_g. The following theorem shows that our value functions have the desired interpretation.

Theorem 3.2.

$$V(g) = \begin{cases} \max\limits_{\substack{S \subseteq N: \\ \pi_{g-1}(S) = l_g}} R(S, g) & \text{if } g > 1 \\ \max\limits_{S \subseteq N} R(S, 1) & \text{if } g = 1 \end{cases}. \tag{7}$$

Proof. We show both cases in the theorem statement by induction. First, we consider the case in which $g > 1$. The result holds trivially for the base case for customer class $m + 1$. Therefore, we assume inductively that the result holds for customer classes $g' > g$ and prove the result for g.

Let S^* be the assortment that maximizes the right hand side of (7) for customer class g. Suppose that $k_g \in S^*$. Then, all customers from g to $O(g) - 1$ purchase product k_g. Further, by our assumption that two products never flip twice, no customer class $g' > g$ will purchase product l_g. In other words, for any subset $S \subseteq N$ such that $\pi_{O(g)-1}(S) = k_g$, we know that $R(S, O(g)) = R(S \cup \{l_g\}, O(g))$. Therefore, given our assumption that $k_g \in S^*$, S^* also maximizes $R(S, O(g))$, and

$$\begin{aligned} \max_{\substack{S \subseteq N: \\ \pi_{g-1}(S) = l_g}} R(S, g) &= r_{k_g} Q^o(g) - f_{k_g} + \max_{\substack{S \subseteq N: \\ \pi_{O(g)-1}(S) = k_g}} R(S, O(g)) \\ &= r_{k_g} Q^o(g) + V(O(g)), \end{aligned}$$

where the first equality holds by our assumption of $k_g \in S^*$ and our observation above, and the second equality follows by the inductive hypothesis. Similarly, if $k_g \notin S^*$ then we have

$$\begin{aligned} \max_{\substack{S \subseteq N: \\ \pi_{g-1}(S) = l_g}} R(S, g) &= r_{l_g} Q^n(g) + \max_{\substack{S \subseteq N: \\ \pi_{N(g)-1}(S) = l_g}} R(S, N(g)) \\ &= r_{l_g} Q^n(g) + V(N(g)). \end{aligned}$$

By taking the maximum of the two cases above,

$$\max_{\substack{S \subseteq N: \\ \pi_{g-1}(S) = l_g}} R(S, g) = \max\{r_{k_g} Q^o(g) - f_{k_g} + V(O(g)), r_{k_g} Q^n(g) + V(N(g))\} = V(g),$$

where the second equality follows by definition of $V(g)$. For $g = 1$, a similar argument conditioning on the product purchased by the first customer class yields

$$\max_{S \subseteq N} R(S, 1) = \max_{j \in N_1} \{r_j Q^1(j) - f_j + V(O(1, j))\} = V(1).$$

$\square$

Theorem 3.2 immediately gives that $V(1) = \text{OPT}_f$, which proves the correctness of the dynamic program given in (6). In the proposition below, we show that the value functions can be computed in a runtime of $O(m)$.

Proposition 3.3. *The value functions of the dynamic program given in (6) can be computed in $O(m) = O(n^2)$ operations.*

Proof. We delay the proof to Appendix A. $\square$

The argument that we use to prove Proposition 3.3 relies on a method to efficiently compute the two sums in (6) for each customer class $g \in \mathcal{G}$. Our final runtime of $O(m) = O(n^2)$ improves upon the best previous runtime of $O(n^3)$ for the unconstrained assortment problem under the linear utilities model achieved by Pan and Honhon (2012). Considering that it takes $O(m)$ operations to compute the revenue of any assortment, it is likely that our approach achieves the best possible runtime.

4 Computational Experiments

In this section, we present computational experiments on a testbed of randomly generated choice models in order to better understand the performance of our algorithm in practice. Specifically, we test the performance of our algorithm on sequential flips (SF) choice models, multinomial logit (MNL) choice models, and Markov chain (MC) choice models. For these three choice models, the underlying fixed cost problem is solved exactly, approximately, or heuristically, respectively. All experiments were implemented in C++11 and conducted on a MacBook Air with a 1.7 GHz Intel Core i5 processor and 4 GB of RAM. We detail the testbed of instances for each choice model below. For all instances, we set $n = 100$.

Sequential Flips Instances. Consistent with the motivation for the sequential flips model, we use an underlying linear utilities model to generate each test choice model. In particular, we first independently generate the quality q_i and space consumption c_i for each product i uniformly at random on the interval $[0, 1]$. Next, we set the price of each product i to be $p_i = f(q_i)$, an increasing function of the quality. We first note that setting f to be a linear function of the qualities implies that all customers prefer higher quality products to lower quality products leading to a trivial optimization problem. Therefore, to test various sensitivities to quality, we either set $f(q) = \alpha \exp(q) + \beta$ or $f(q) = \alpha q^2 + \beta$. To determine α and β, we set the lowest price of a product to be $p_l = 100$ and the highest price to be $p_h \in \{500, 1000\}$. By varying p_h, we capture settings with lower or higher price differentiation. We let exp denote the function f when $f(q) = \alpha \exp(q) + \beta$, and we let quad denote f when $f(q) = \alpha q^2 + \beta$.

As shown in Proposition 3.1, the prices and qualities of products determine the set of preference lists for the corresponding sequential flips choice model. Therefore, all that is left to do is to set the arrival probabilities for each preference list. To do so, we first discard all preference lists where 0 is the first choice product since these lists represent customers for which θ is small enough so that the utility of every product is negative. The remaining lists we index $1, 2, \dots, m$ such that list g is generated from list $g - 1$ by flipping two products. By the nature of the linear utilities model, these flips represent higher quality (higher priced) products flipping above lower quality (lower priced) products. Given this intuition, we set the probability of list $g \in \{1, 2, \dots, m\}$ to either be $\lambda_g = h(g) = U[0, 1]$ or $\lambda_g = h(g) = (m - g + 1) \cdot U[0, 1]$, where $U[0, 1]$ represents an independent uniform random variable on the interval $[0, 1]$. We then normalize these probabilities. In the first case, arriving customers are equally likely to have preference lists with low or high values of θ and so there is an even balance between more price-conscious or more quality-conscious customers. On the other hand, in the second case, arriving customers are more likely to have low values of θ and so are more likely to be price-conscious. This implies that higher revenue products capture a smaller piece of the overall population so there is a more delicate trade-off with regards to which products to offer. We let rand denote the function h when $h(g) = U[0, 1]$ and let dec denote the function h when $h(g) = (m - g + 1) \cdot U[0, 1]$.

For every parameter setting $\text{SF}(p_h, f, h)$ with $p_h \in \{500, 1000\}$, $f \in \{\exp, \text{quad}\}$, and $h \in \{\text{rand}, \text{dec}\}$, we generate 100 random instances. We solve the corresponding fixed cost problem using the algorithm described in Section 3.2.

Multinomial Logit Instances. We use a similar approach to generate our MNL instances. For each product i we uniformly generate value v_i on the interval $[0, 50]$ and set $v_0 = 10$. Recall that under the MNL model the probability that product i is purchased in assortment S is

$$\Pr{}_i(S) = \frac{v_i}{v_0 + \sum_{j\in S} v_j}.$$

Then, to test various sensitivities to value, we either set $r_i = f(v) = \alpha \exp(v) + \beta$ or $r_i = f(v) = \alpha v^2 + \beta$. To determine α and β, we set the lowest price of a product to be $p_l = 100$ and the highest price to be $p_h \in \{500, 1000\}$. We let exp denote the function f when $f(q) = \alpha \exp(q) + \beta$, and we let quad denote f when $f(q) = \alpha q^2 + \beta$.

Last, we generate n space consumptions uniformly on the interval $[0, 1]$. We denote this vector of capacities as cap and assume that products are indexed by revenue. It is well-known that optimal assortments for the uncapacitated assortment problem under the MNL choice model are revenue-ordered. Therefore, we either set the capacity of product i to be $h(i) = \text{cap}[i]$ or $h(i) = \text{sort}(\text{cap})[i]$. The former associates random capacities with each product while the latter case sorts capacities by revenue and is more likely to break the structure of the uncapacitated optimal solution. We let rand denote the function h when $h(i) = \text{cap}[i]$ and let inc denote the function h when $h(i) = \text{sort}(\text{cap})$.

For every parameter setting $\text{ML}(p_h, f, h)$ with $p_h \in \{500, 1000\}$, $f \in \{\exp, \text{quad}\}$, and $h \in \{\text{rand}, \text{inc}\}$, we generate 100 random instances. The fixed cost problem under the MNL model is NP-hard. Therefore, we solve the corresponding fixed cost problem using the approximation algorithm described in Kunnumkal and de Albeniz (2017).

Markov Chain Instances. Last, we generate the Markov chain instances. In this setting, the corresponding choice process is represented by a Markov chain on the set of products (plus the no-purchase option) in which offered products are absorbing states and the corresponding purchase probability is the probability of absorption. For each product i, we uniformly generate each entry of the ith row of the transition matrix T on the interval $[0, 1]$ (setting $T_{ii} = 0$) and then scale so that the transition probabilities sum to 1. Next, to test various sensitivities to value, we either set $r_i = f(x) = \alpha \exp(x) + \beta$ or $r_i f(x) = \alpha x^2 + \beta$, where x is randomly generated on the interval $[0, 1]$. To determine α and β, we set the lowest price of a product to be $p_l = 100$ and the highest price to be $p_h \in \{500, 1000\}$. We let exp denote the function f when $f(q) = \alpha \exp(q) + \beta$, and we let quad denote f when $f(q) = \alpha q^2 + \beta$. We then generate space consumption c_i uniformly on the interval $[0, 1]$.

Last, we generate n initial arrival probabilities uniformly on the interval $[0, 1]$. We denote this vector of probabilities as probs and assume that products are indexed by revenue. We then either set the arrival probability of product i to be $h(i) = \text{probs}[i]$ or $h(i) = \text{sort}(\text{probs})[n-i]$. The former case associates random arrival probabilities with each product while the latter case sorts the arrival probabilities by decreasing revenue and is likely to yield a more complicated trade-off between revenue and purchase probability. We let rand denote the function h when $h(i) = \text{cap}[i]$ and let dec denote the function h when $h(i) = \text{sort}(\text{cap})$.

The fixed cost problem under the MC choice model is NP-hard, and to the best of our knowledge, there is no known approximation algorithm. Therefore, we solve the corresponding fixed cost problem using a local search algorithm that on each iteration considers adding a product, removing a product, or swapping one product in and one product out of the assortment to increase revenue. This is a heuristic and is not guaranteed to find an optimal solution. However, we find that in practice it performs well.

Upper Bounds For each instance, we first solve the uncapacitated problem to find the optimal assortment S^u. We then solved the space constrained assortment problem for both $C = 0.5 \cdot C(S^u)$ and $C = 0.75 \cdot C(S^u)$ with $\epsilon = 0.01$. To analyze the returned solution S^* for the sequential flips instances, we note that Proposition 2.5 implies that $\text{UB} = [\alpha \text{Rev}(S_1) + (1-\alpha)\text{Rev}(S_2)]/(1-\epsilon)$ is an upper bound on the optimal revenue for the space constrained problem. This is a useful metric for the retailer to analyze the returned solution. However, for the MNL and MC instances this upper bound does not apply since we do not optimally solve the fixed cost problem. Instead, we find the optimal expected revenue by solving integer programs for the corresponding space constrained problem (see Kunnumkal (2015) for the MNL model and Désir et al. (2015) for the MC model). Although inefficient relative to the speed of our two step approximation scheme, this approach allows us to more accurately analyze the performance of our returned assortments.

4.1 Results

The results of the experiments are reported in Table 2. Column 1 gives the instance type. For the case when $C = 0.5 \cdot C(S^u)$, Column 2 reports the average runtime in seconds, Column 3 reports the average number of products offered, Column 4 reports the relative cardinality of the outputted assortment compared to $|S^u|$, Column 5 reports the percentage of products offered that overlap with S^u, Column 6 reports the average optimality gap, and Column 7 reports the maximum optimality gap. Columns 8-13 report the same for $C = 0.75 \cdot C(S^u)$.

We first note the surprising efficiency of the algorithm despite the binary search on λ to find

Instance	$C = 0.5 \cdot C(S^u)$						$C = 0.75 \cdot C(S^u)$					
	Time (s)	Size	% Size	% Over	Avg. Gap	Max. Gap	Time (s)	Size	% Size	% Over	Avg. Gap	Max. Gap
SF(500,exp,dec)	2.57e-4	9.55	64.3	80.3	0.03	0.16	2.93e-4	11.4	76.0	92.0	0.01	0.11
SF(500,exp,rand)	2.46e-4	7.21	68.2	76.1	0.09	0.81	3.18e-4	8.23	77.5	88.0	0.02	0.14
SF(500,quad,dec)	5.00e-4	12.33	59.6	75.3	0.02	0.08	5.65e-4	15.48	74.2	92.2	0.01	0.05
SF(500,quad,rand)	4.35e-4	5.27	68.3	76.6	0.09	1.09	5.77e-4	6.06	76.4	90.9	0.03	0.80
SF(1000,exp,dec)	4.96e-4	13.37	57.8	78.2	0.02	0.14	5.77e-4	17.21	73.8	95.1	0.01	0.05
SF(1000,exp,rand)	5.37e-4	9.21	68.2	69.2	0.05	0.22	6.18e-4	10.35	76.9	82.8	0.02	0.17
SF(1000,quad,dec)	6.61e-4	8.84	55.9	80.7	0.01	0.03	7.38e-4	11.97	75.1	98.2	0.00	0.02
SF(1000,quad,rand)	5.82e-4	3.3	68.0	79.9	0.18	1.30	6.63e-4	3.95	79.4	86.3	0.12	1.06
ML(500,exp,inc)	4.71e+0	3.56	60.0	95.9	0.24	2.17	4.67e+0	4.47	75.4	99.3	0.08	0.70
ML(500,exp,rand)	3.98e+0	3.73	61.6	95.2	0.21	2.16	4.18e+0	4.59	75.3	99.1	0.08	1.14
ML(500,quad,inc)	2.55e+0	3.15	60.1	94.1	0.23	3.06	2.55e+0	3.9	73.9	98.1	0.18	2.03
ML(500,quad,rand)	3.23e+0	3.19	57.6	94.7	0.40	3.02	3.27e+0	4.09	73.8	99.0	0.11	1.21
ML(1000,exp,inc)	4.73e+0	3.44	61.8	92.3	0.42	5.62	4.67e+0	4.28	75.7	98.7	0.06	0.77
ML(1000,exp,rand)	3.03e+0	3.41	59.3	97.0	0.20	2.12	3.13e+0	4.19	73.2	98.9	0.10	1.01
ML(1000,quad,inc)	2.60e+0	3.01	60.8	93.5	0.43	4.86	2.66e+0	3.73	74.3	98.7	0.16	1.64
ML(1000,quad,rand)	3.64e+0	2.84	58.1	92.8	0.41	5.54	3.64e+0	3.56	72.6	97.9	0.10	1.36
MC(500,exp,dec)	3.00e+1	6.45	44.2	100.0	0.15	2.42	2.89e+1	9.56	69.6	100.0	0.01	0.68
MC(500,exp,rand)	3.40e+1	6.53	44.2	100.0	0.16	2.62	3.10e+1	9.72	69.9	100.0	0.02	0.78
MC(500,quad,dec)	2.99e+1	5.9	43.6	100.0	0.23	5.34	2.85e+1	8.8	69.5	100.0	0.03	1.35
MC(500,quad,rand)	3.32e+1	5.75	43.4	100.0	0.40	7.66	3.15e+1	8.55	69.1	100.0	0.05	1.44
MC(1000,exp,dec)	3.00e+1	6.21	44.3	99.7	0.27	3.12	2.90e+1	9.19	69.6	100.0	0.02	0.73
MC(1000,exp,rand)	3.21e+1	6.12	43.7	100.0	0.21	2.80	3.04e+1	9.15	69.7	100.0	0.01	0.43
MC(1000,quad,dec)	2.86e+1	5.48	42.9	99.6	0.35	6.43	2.72e+1	8.21	69.3	99.9	0.02	0.96
MC(1000,quad,rand)	3.23e+1	5.4	42.8	100.0	0.38	6.89	3.02e+1	8.03	68.5	100.0	0.03	1.49

Table 1: Performance of the space-constrained approximation scheme on a set of choice models. Reported numbers are averages unless otherwise stated.

S_1 and S_2. The largest average runtime was 0.0006 seconds for SF model instances, 4.7 seconds for MNL model instances, and 34 seconds for MC model instances, where this increase is due to the time spent solving the underlying fixed cost problem. Further, the algorithm performs significantly better than the approximation guarantee. The largest optimality gap seen was 1.3% for the SF model instances, 5.62% for the MNL model instances, and 7.66% for the MC model instances with the average optimality gap never exceeding 0.4% over all test cases. Interestingly, the fact that we only heuristically solve the fixed cost assortment problem for the MC models does not seem to have a drastic effect on the profitability of the assortments that our approximation scheme produces. This result suggests that a provably good approximation scheme for the fixed cost assortment problem is not a barrier for using our approach. We also find that setting $C = 0.5 \cdot C(S^u)$ yields larger optimality gaps than $C = 0.75 \cdot C(S^u)$, which matches our intuition that as we make the space constraint tighter, the assortment problem becomes harder. For the MC model instances, we observe that the returned assortments have the smallest cardinalities relative to $|S^u|$. This indicates that there may be a small subsets of products that garner a large portion of the revenue.

With regards to the overlap in composition between the returned assortment and S^u, we find under the MNL and Markov chain choice models that the two assortments very much align, indicating that a heuristic which solves unconstrained assortment problems and then greedily removes products from this assortment might perform well for some instances of the space constrained problem. On the other hand, for the sequential flips model, we find that the two assortments are less aligned; suggesting that when products are vertically differentiated, it is important to account for a space constraint by specifically solving space constrained assortment problems. When $C = 0.5 \cdot C(S^u)$, we find that this is especially the case. In this tighter capacity setting, the overlap can be as low as 69%; we observe that the returned assortment seems to retain a key subset of the optimal unconstrained assortment and then adds in products with high revenue and low capacity.

5 Conclusion

In this paper, we present a link between the space constrained assortment problem and the fixed cost assortment problem for any random utility choice model by developing an approximation scheme for the former that relies on a black box solver for the latter. Most notably, this approximation scheme allows us to solve the space constrained assortment problem for the sequential flips model, a generalization of the classical linear utilities model. Further, we show that this approach performs near optimal in practical experiments. One interesting direction for future work could be to investigate whether the fixed cost problem can be shown to be strictly easier than the space constrained problem. In other words, can an optimal algorithm for the space constrained problem be used as a black box to provide an optimal algorithm for the fixed cost problem? This is certainly true when the fixed costs are the same for each product; one can simply enumerate over the n guesses for the sum of the fixed costs in the optimal solution and write a constraint akin to a space constraint enforcing that the sum of the fixed cost of the chosen assortment does not exceed this guess. However, this technique cannot be extended easily for arbitrary fixed costs.

References

Aouad, A., Farias, V. F. and Levi, R. (2015), Assortment optimization under consider-then-choose choice models. Available at SSRN: `https://ssrn.com/abstract=2618823`.

Aouad, A., Farias, V. F., Levi, R. and Segev, D. (2015), The approximability of assortment optimization under ranking preferences. Available at SSRN: `https://ssrn.com/abstract=2612947`.

Blanchet, J., Gallego, G. and Goyal, V. (2016), 'A Markov chain approximation to choice modeling', *Operations Research* **64**(4), 886–905.

Davis, J. M., Gallego, G. and Topaloglu, H. (2013), Assortment planning under the multinomial logit model with totally unimodular constraint structures. Working Paper, Department of IEOR, Columbia University. Available at `http://www.columbia.edu/gmg2/logit_const.pdf` (accessed on May 25, 2017).

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

Désir, A. and Goyal, V. (2014), Near-optimal for capacity constrained assortment optimization. Available at SSRN: `https://ssrn.com/abstract=2543309`.

Désir, A., Goyal, V., Segev, D. and Ye, C. (2015), Capacity constrained assortment optimization under the Markov chain based choice model. Available at SSRN: `https://ssrn.com/abstract=2626484`.

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

Feldman, J. (2017), Space constrained assortment optimization under the paired combinatorial logit model. Unpublished Manuscript.

Feldman, J. and Topaloglu, H. (2015*a*), 'Bounding optimal expected revenues for assortment optimization under mixtures of multinomial logits', *Production and Operations Management* **24**(10), 1598–1620.

Feldman, J. and Topaloglu, H. (2015*b*), 'Capacity constraints across nests in assortment optimization under the nested logit model', *Operations Research* **63**(4), 812–822.

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

Honhon, D., Jonnalagedda, S. and Pan, X. A. (2012), 'Optimal algorithms for assortment selection under ranking-based consumer choice models', *Manufacturing & Service Operations Management* **14**(2), 279–289.

Jagabathula, S. (2014), Assortment optimization under general choice. Available at SSRN: `https://ssrn.com/abstract=2512831`.

Kunnumkal, S. (2015), 'On upper bounds for assortment optimization under the mixture of multinomial logit models', *Operations Research Letters* **43**(2), 189 – 194.
URL: *http://www.sciencedirect.com/science/article/pii/S0167637715000176*

Kunnumkal, S. and de Albeniz, V. M. (2017), Tractable models and algorithms for assortment planning with product costs. Unpublished Manuscript.

Li, G. and Rusmevichientong, P. (2014), 'A greedy algorithm for the two-level nested logit model', *Operations Research Letters* **42**(5), 319–324.

Mahajan, S. and van Ryzin, G. (2001*a*), 'Inventory competition under dynamic consumer choice', *Operations Research* **49**(5), 646–657.

Mahajan, S. and van Ryzin, G. (2001*b*), 'Stocking retail assortment under dynamic consumer substitution', *Operations Research* **49**, 334–351.

Méndez-Díaz, I., Miranda-Bront, J. J., Vulcano, G. and Zabala, P. (2014), 'A branch-and-cut algorithm for the latent-class logit assortment problem', *Discrete Applied Mathematics* **164**, 246–263.

Mussa, M. and Rosen, S. (1978), 'Monopoly and product quality', *Journal of Economic Theory* **18**(2), 301–317.

Pan, X. A. and Honhon, D. (2012), 'Assortment planning for vertically differentiated products', *Production and Operations Management* **21**(2), 253–275.

Paul, A., Feldman, J. and Davis, J. (2017), 'Assortment optimization and pricing under a nonparametric tree choice model', *Manufacturing & Service Operations Management* . To appear.

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

Rusmevichientong, P., Shmoys, D. B., Tong, C. and Topaloglu, H. (2014), 'Assortment optimization under the multinomial logit model with random choice parameters', *Production and Operations Management* **23**(11), 2023–2039.

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

Topaloglu, H. (2009), 'Using lagrangian relaxation to compute capacity-dependent bid prices in network revenue management', *Operations Research* .

van Ryzin, G. and Vulcano, G. (2017), 'Technical note—an expectation-maximization method to estimate a rank-based choice model of demand', *Operations Research* **65**(2), 396–407.

Wang, R. (2012), 'Capacitated assortment and price optimization under the multinomial logit model', *Operations Research Letters* **40**(6), 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.

A Proofs

A.1 Extension of Approximation Guarantee

Theorem A.1. *If there exists a δ-approximation algorithm, where $0 \leq \delta \leq 1$, for the fixed cost assortment optimization problem under any RUM, then there exists a $\frac{(1-\epsilon)\delta}{3}$-approximation algorithm for the corresponding space constrained assortment optimization problem and a $\frac{(1-\epsilon)\delta}{2}$-approximation algorithm for the cardinality constrained assortment optimization problem.*

Proof. We construct α, S_1, and S_2 as before using bisection search with calls to the approximation algorithm for the fixed cost assortment problem. As before, suppose that we can find $\lambda^* \geq 0$ such that $C(S_{\lambda^*}) = C$. Then,

$$\text{Rev}(S_{\lambda^*}) = \text{Rev}(S_{\lambda^*}) - \lambda^* C(S_{\lambda^*}) + \lambda^* C \geq \delta \cdot \text{OPT}_{\lambda^*} \geq \delta \cdot \text{OPT}_c.$$

This implies that we are within a factor of δ of the optimal revenue for the space constrained assortment problem. We show that we either we can find λ^* or we can find S_{λ_1} and S_{λ_2} along with a carefully chosen α such that $\alpha\text{Rev}(S_1) + (1-\alpha)\text{Rev}(S_2) \geq \delta(1-\epsilon) \cdot \text{OPT}_c$ for any $\epsilon \geq 0$. Hence, in the worst case, we produce an assortment with an approximation guarantee of $\frac{(1-\epsilon)\delta}{3}$, using the results of Section 2.1.

We find S_{λ_1} and S_{λ_2} through bisection search on the same interval $[0, r_{\text{max}}/c_{\text{min}}]$. Again, initially, we set $\lambda_1 = 0$ and $\lambda_2 = r_{\text{max}}/c_{\text{min}}$. Note that for $\lambda = \lambda_1$ the problem reduces to the unconstrained assortment problem. If $C(S_{\lambda_1}) \leq C$, then $\text{Rev}(S_1) \geq \delta\text{OPT}_c$. Otherwise, assume $C(S_{\lambda_1}) > C$. On the other hand, for $\lambda = \lambda_2$ the penalty for offering any subset is always at least the revenue generated by that subset and $S_{\lambda_2} = \emptyset$. Consequently, $C(S_{\lambda_2}) < C < C(S_{\lambda_1})$. Given λ_1 and λ_2, in each iteration the algorithm tests the midpoint $\lambda' = \frac{1}{2}(\lambda_1 + \lambda_2)$. If $C(S_{\lambda'}) = C$, then we return $S_{\lambda'}$. Otherwise, if $C(S_{\lambda'}) > C$, then we set $\lambda_1 = \lambda'$, and if $C(S_{\lambda'}) < C$, then we set $\lambda_2 = \lambda'$. We repeat this process until either we find an assortment with a space consumption of exactly C or until $\lambda_2 - \lambda_1 \leq (\epsilon\delta\text{Rev}_{\text{min}})/\mathcal{C}$.

Given $S_1 = S_{\lambda_1}$ and $S_2 = S_{\lambda_2}$ as constructed above, we set

$$\alpha = \frac{C - C(S_2)}{C(S_1) - C(S_2)}.$$

By construction of S_1 and S_2, $0 \leq \alpha \leq 1$. First, we note that

$$\alpha C(S_1) + (1-\alpha)C(S_2) = \frac{C - C(S_2)}{C(S_1) - C(S_2)} C(S_1) + \frac{C(S_1) - C}{C(S_1) - C(S_2)} C(S_2) = C.$$

Second, we analyze the revenue of the convex combination. For any $\lambda \geq 0$, we note that $\mathrm{OPT}_\lambda \geq \mathrm{OPT}_c$ since any feasible assortment only has higher revenue in this fixed cost problem. Therefore,

$$\begin{aligned}
\delta \mathrm{OPT}_c &\leq \delta \left[\alpha \mathrm{OPT}_{\lambda_1} + (1-\alpha)\mathrm{OPT}_{\lambda_2}\right] \\
&\leq \alpha \mathrm{Rev}(S_1) + (1-\alpha)\mathrm{Rev}(S_2) + \alpha\lambda_1[C - C(S_1)] + (1-\alpha)\lambda_2[C - C(S_2)] \\
&= \alpha \mathrm{Rev}(S_1) + (1-\alpha)\mathrm{Rev}(S_2) + \lambda_2\left(\alpha[C - C(S_1)] + (1-\alpha)[C - C(S_2)]\right) \\
&- (\lambda_2 - \lambda_1)\alpha(C - C(S_1)) \\
&= \alpha \mathrm{Rev}(S_1) + (1-\alpha)\mathrm{Rev}(S_2) - (\lambda_2 - \lambda_1)\alpha(C - C(S_1)) \\
&\leq \alpha \mathrm{Rev}(S_1) + (1-\alpha)\mathrm{Rev}(S_2) + (\lambda_2 - \lambda_1)\mathcal{C} \\
&\leq \alpha \mathrm{Rev}(S_1) + (1-\alpha)\mathrm{Rev}(S_2) + \epsilon\delta \mathrm{Rev}_{\min},
\end{aligned}$$

where the second to last inequality follows from our stopping criterion on the difference between λ_2 and λ_1. □

A.2 Computational Complexity of Space Constrained Problem

Theorem A.2. *The space constrained assortment problem under the SF nonparametric choice model is NP-Hard.*

Proof. We use a reduction from the knapsack problem. Given an instance of the knapsack problem with n items, let $v_1, \ldots, v_n$ be the values of the items and let $w_1, \ldots, w_n$ be the sizes of the items. We assume that the items are indexed such that if $i < j$ then $v_i \leq v_j$.

We create an equivalent instance of the space constrained assortment problem under the SF nonparametric choice model. First, we create $2n-1$ products indexed by the set $N = \{1, 2, \ldots, 2n-1\}$ and set $M > \max_i v_i$. For odd $i \in N$, we set $r_i = v_{(i+1)/2} + M \cdot (i-1)/2$ and set $c_i = w_{(i+1)/2}$. For even $i \in N$, we set $r_i = M \cdot i/2$ and set $c_i = 0$. We start with the preference list $[2n-1, 2n-2, \ldots, 1, 0]$. Then, we flip product $2n-1$ down until it flips with product 0. This creates the preference list $[2n-2, 2n-3, \ldots, 0, 2n-1]$. We continue to iteratively flip the current first choice product down until it flips with product 0 to create $2n-1$ customer classes with preference lists of the form $[i, i-1, \ldots, 1, 0, \ldots]$ for $i = 1, \ldots, 2n-1$ each arriving with unnormalized probability $\lambda_g = 1$. All other generated preference lists have arrival probability $\lambda_g = 0$. Last, we set $C = W$.

We first show that it is always optimal to offer all even indexed products. To see this, note that an even indexed product i only blocks products $1, \ldots, i-1$, which all have lower revenues than product i, by construction. Further, product i does not consume any space. Conditioned on the fact that even indexed products are always offered, we show that adding an odd indexed product i generates an additional revenue of $v_{(i+1)/2}$. Specifically, for any assortment S containing all even indexed products and $i \notin S$, we have

$$\text{Rev}(S \cup \{i\}) - \text{Rev}(S) = v_{(i+1)/2} + M \cdot (i-1)/2 - M \cdot (i-1)/2 = v_{(i+1)/2}.$$

This follows from the fact that the only customer class to switch products will be the customer class with product i as their first choice product. Further, adding product i to the assortment takes up additional space $c_i = w_{(i+1)/2}$. Thus, the problem of determining which additional products to offer (beyond all even indexed products) is exactly the original knapsack problem. □

A.3 Proof of Proposition 3.3

The running time of the dynamic program depends on calculating the value functions $V(g)$ that represent the maximum expected revenue accrued from customer classes $g, g+1, \ldots, m$ when customer class $g-1$ purchases product l_g. The dynamic program is given below.

$$V(g) = \begin{cases} \max\{r_{k_g} Q^o(g) - f_{k_g} + V(O(g)), r_{l_g} Q^n(g) + V(N(g))\} & g > 1 \\ \max_{j \in N_1}\{r_j Q^1(j) - f_j + V(O(1,j))\} & g = 1 \end{cases} \tag{8}$$

with base case

$$V(m+1) = 0.$$

To solve the dynamic program, we will first calculate all $N(g)$, $O(g)$, $Q^o(g)$, $Q^n(g)$, and $Q^1(j)$ values in pre-processing. To do so, for each product i, we will keep a stack of tuples (g, F_g, I) where g will be a customer class, $F_g = \sum_{i=1}^{g-1} \lambda_g$, and I will either be O or N. Initially, all stacks will be empty and we will set $F = 0$. For $g = 1, 2, \ldots m$, we first empty the stack for l_g.

- If the stack for l_g is already empty, we set $Q^1(l_g) = F$ and $O(1, j) = g$.
- Otherwise, for each $(g', F_{g'}, O)$ on the stack for l_g, we set

  $$Q^o(g') = F - F_{g'}$$

  and for each $(g', F_{g'}, N)$ on the stack for l_g, we set

  $$Q^n(g') = F - F_{g'}.$$

We then add (g, F, O) to the stack for product k_g and add (g, F, N) to the stack for product l_g. Lastly, we set $F = F + \lambda_g$ before moving to the next customer class.

For each customer class g, (g, F_g, O) is added to the stack when processing g and it is removed the first time $l_{g'} = k_g$, and (g, F_g, N) is added to the stack when processing g and removed the first time $l_{g'} = l_g$. Therefore, all $O(g)$, $Q^o(g)$, $N(g)$, and $Q^n(g)$ values are calculated correctly. Similarly, $Q^1(j)$ is set the first time $l_g = j$ so $O(1, j)$ and $Q^1(j)$ are set correctly. Further, the pre-processing iterates through all customer classes and adds/removes each element to the stack exactly once. Therefore, the overall running time is $O(m)$.

After pre-processing, calculating all $V(g)$ values involves iterating through all $O(m)$ states and taking the maximum over two possible values. This gives an overall running time of $O(m)$ for the assortment optimization algorithm.

B Estimating the Sequential Flips Model

We fit a sequential flips nonparametric choice model and a linear utility model to synthetically generated sales data sets to see which model can more accurately capture customer purchasing behavior on a set of ten vertically differentiated products indexed by the set $N = \{1, \dots 10\}$. We assume that each model has access to the past purchasing history of τ customers given by $PH_\tau = \{(S_t z_t)\}$, where S_t and Z_t respectively give the set of products offered and the product purchased for the t^{th} customer. We vary $\tau \in \{1000, 5000\}$ to study the benefit of additional sales data on the accuracy of the fits and for each value of τ, we generate five different streams of sales data. We form the subsets S_t for each arrival by randomly including each product in N with probability 0.5. Each arriving customer is assumed to make purchases according to a linear utility model, which we refer to as the ground choice model GC. To capture a setting in which the products could represent hotels, we assume that $q_i = i * 0.5$ so that a product's quality could be viewed as its star rating. We then generate the prices of each product so as to ensure that higher quality products have higher prices. Specifically, we set $p_i = p_{i-1} + U[0, 100]$, where $U[0, 100]$ is a random variables that is uniformly distributed on the interval $[0, 100]$ and $p_1 = U[0, 100]$. Finally, we assume that $F(\theta)$ is uniformly distributed on the interval $[0, 200]$.

For the case of our linear utility model fits LU, we assume access to $F(\theta)$ and then we use MLE to estimate the qualities of each product. Since the log-likelihood in this case is not concave, we solve the MLE problem with ten different random starting points and ultimately choose the set of parameter estimates that has the highest training log-likelihood. On the other hand, for our sequential flips model fits SF, we assume access to the qualities q_i of each product and we use MLE to estimate the distribution $F(\theta)$. For choice model $M \in \{GC, LU, SF\}$, let $\Pr_i^M(S)$ be the probability that product i is purchased under choice model M when assortment $S \subseteq N$ is offered. We measure the efficacy of the fitted model using the Mean Absolute Error (MAE) of the predicted purchase probabilities for the two fitted models in relation to the true purchase probabilities dictated by the ground choice model. More specifically, for fitted choice model $M \in \{LU, SF\}$, we let $\text{MAE}(M) = \frac{1}{2^{10}} \sum_{S \subseteq N} \sum_{i \in S} |\Pr_i^M(S) - \Pr_i^{GC}(S)|/|S|$ be the average error in predicted purchase probability over every product in every assortment.

Table 2 shows the MAE of the two fitted choice models averaged over five different streams of purchase histories. We observe that the average MAE of the fitted sequential flips models are on average at least two orders of magnitude smaller than the MAE of the fitted linear utility

	τ	
Model	1000	5000
SF	0.010	0.006
LU	1.46	1.27

Table 2: Average MAE of the two fitted models.

models. Hence, not only is the sequential flips model easier to estimate since the corresponding MLE problem is concave, but the model's estimates of purchase probabilities produced are far more accurate than those produced by the fitted linear utility models.