http://pubsonline.informs.org/journal/mnsc

MANAGEMENT SCIENCE
Vol. 66, No. 2, February 2020, pp. 698–721
ISSN 0025-1909 (print), ISSN 1526-5501 (online)

Constrained Assortment Optimization Under the Markov Chain–based Choice Model

Antoine Désir,[a] Vineet Goyal,[b] Danny Segev,[c] Chun Ye[d]

[a] Technology and Operations Management Area, INSEAD, 77300 Fontainebleau, France; [b] Department of Industrial Engineering and Operations Research, Columbia University, New York, New York 10027; [c] Department of Statistics, University of Haifa, Haifa 31905, Israel; [d] Amazon, Seattle, Washington 98109

Contact: antoine.desir@insead.edu, https://orcid.org/0000-0001-5654-3301 (AD); vg2277@columbia.edu (VG); segevd@stat.haifa.ac.il, https://orcid.org/0000-0003-4684-2185 (DS); ychun@amazon.com (CY)

Received: October 13, 2017
Revised: June 27, 2018; September 25, 2018
Accepted: October 2, 2018
Published Online in Articles in Advance: August 5, 2019

https://doi.org/10.1287/mnsc.2018.3230

Abstract. Assortment optimization is an important problem that arises in many practical applications such as retailing and online advertising. The fundamental goal is to select a subset of items to offer from a universe of substitutable items to maximize expected revenue when customers exhibit a random substitution behavior captured by a choice model. We study assortment optimization under the Markov chain choice model in the presence of capacity constraints that arise naturally in many applications. The Markov chain choice model considers item substitutions as transitions in a Markov chain and provides a good approximation for a large class of random utility models, thereby addressing the challenging problem of model selection in choice modeling. In this paper, we present constant factor approximation algorithms for the cardinality- and capacity-constrained assortment-optimization problem under the Markov chain model. We show that this problem is APX-hard even when all item prices are uniform, meaning that, unless P = NP, it is not possible to obtain an approximation better than a particular constant. Our algorithmic approach is based on a new externality adjustment paradigm that exactly captures the externality of adding an item to a given assortment on the remaining set of items, thereby allowing us to linearize a nonlinear, nonsubmodular, and nonmonotone revenue function and to design an iterative algorithm that iteratively builds up a provably good assortment.

History: Accepted by Yinyu Ye, optimization.
Funding: Funding is acknowledged from the National Science Foundation Division of Civil, Mechanical and Manufacturing Innovation [Grants 1351838 and 1636046], a Google Faculty Award, and the Israel Science Foundation [Grant 148/16].
Supplemental Material: The online appendix is available at https://doi.org/10.1287/mnsc.2018.3230.

Keywords: assortment optimization • choice models • approximation algorithms • Markov chain

1. Introduction

Assortment-optimization problems arise widely in many practical applications, such as retailing and online advertising, in which the goal is to select a subset from a universe of substitutable items to offer to customers to maximize the expected revenue. One of the key challenges in assortment optimization is to model the stochastic demands of different items that depend on the substitution behavior of the customers, that is, how customers choose when their most preferred item is not available. Therefore, the demand of any item is a complex function that depends on the entire set of offered items. The substitution behavior and its resulting demand are commonly captured by a *choice model* that specifies the probability of a random customer selecting a particular item from a given offer set.

In the most general setting, a choice model can be thought of as a distribution over permutations that arise from preferences. In the literature, preferences are commonly modeled using a random utility model. In the random utility model of preferences, each customer has a utility of $u_j + \epsilon_j$ for item j, where u_j depends on the attributes of item j and ϵ_j is a random idiosyncratic component of the utility according to some distribution. Here, the preference of the customer is given by the decreasing order of utilities of items. Therefore, the parameters u_j and the distributions of ϵ_j completely specify the distribution over permutations and, hence, the choice model. Since the introduction of this model by Thurstone (1927), many random utility models have extensively been studied in diverse areas, including marketing, transportation, economics, and operations management.

The multinomial logit (MNL) model (Luce 1959, McFadden 1973, Plackett 1975), in which the random component of the utility is distributed according to a standard Gumbel distribution, has, by far, been the most popular model in practice, primarily because of the tractability of both its estimation and assortment-optimization problems (Gallego et al. 2004, Talluri and van Ryzin 2004, Farias et al. 2011). However, some of the model's implications (for instance, *independence from irrelevant alternatives* property) are not reasonable for many applications. Consequently, more complex choice models have been considered to capture a richer class of substitution behavior. Such models include, for instance, the nested logit model (Williams 1977, McFadden et al. 1978, Davis et al. 2014, Gallego and Topaloglu 2014) and the mixture of multinomial logit model (McFadden and Train 2000); see Train (2009) for a detailed overview of these models. Although the increased modeling complexity makes these models more flexible, both the estimation and the assortment-optimization problems become computationally difficult to solve, and there is a fundamental trade-off between their tractability and predictive power. However, even if we ignore the tractability issues associated with estimation and optimization over a particular model, one of the key challenges in choice modeling is to find the "right" model to describe customers' preferences and substitution behavior. This is especially challenging as customer preferences are latent and unobservable in the sales data, and errors in model selection can lead to substantial errors in demand predictions and, subsequently, to highly suboptimal decisions.

In a recent paper, Blanchet et al. (2016) study the Markov chain–based choice model in which customer decisions are captured by a Markov chain. In particular, each item (including the no-purchase option) corresponds to a state, and substitutions are modeled using transitions in the Markov chain. To our knowledge, Zhang and Cooper (2005) were the first to consider the Markov chain model in the context of airline revenue management, and they present a simulation study using the Markov chain model. Blanchet et al. (2016) revisit this model and formally investigate its predictive power. The authors show that this model provides a good approximation in choice probabilities to a large class of existing choice models, allowing it to circumvent the model-selection problem. In particular, the Markov chain choice model is a generalization of several well-studied choice models, including MNL, the generalized attraction model (Gallego et al. 2015), and the exogenous demand model (Kök and Fisher 2007). Furthermore, Blanchet et al. (2016) show that the unconstrained assortment-optimization problem under the Markov chain model can be solved efficiently using linear programming (LP) as well as a value iteration algorithm. Feldman and Topaloglu (2017b) study the network revenue management problem under the Markov chain model, for which a linear programming–based algorithm is proposed. Simsek and Topaloglu (2017) present an expectation maximization–based algorithm to estimate the parameters of the Markov chain model from choice data. Subsequently, several variants of the Markov chain model have been considered in recent works (see Ragain and Ugander 2016, Nip et al. 2017, and Paul et al. 2017).

In this paper, we consider the capacity-constrained assortment problem under the Markov chain model. Here, every item i is associated with a weight w_i, and the decision maker is restricted to selecting an assortment whose total weight is at most a given bound, W. Therefore, the capacity-constrained assortment-optimization problem can be formulated as

$$\max_{S\subseteq\mathcal{N}}\left\{R(S):\sum_{i\in S}w_i\le W\right\},\qquad \text{(Capacity-Assort)}$$

where $\mathcal{N}$ denotes the universe of substitutable items and $R(S)$ denotes the expected revenue of the assortment $S\subseteq\mathcal{N}$ under the Markov chain model. The formal definition of this model, along with additional notation, are given in Section 1.3. For the special case of uniform item weights (i.e., $w_i = 1$ for all i), the capacity constraint reduces to an upper bound on the number of items in the assortment. We refer to this setting as the cardinality-constrained assortment-optimization problem:

$$\max_{S\subseteq\mathcal{N}}\{R(S):|S|\le k\}.\qquad \text{(Cardinality-Assort)}$$

Cardinality and capacity constraints on assortments arise naturally in many applications, allowing one to model practical scenarios, such as a shelf-space constraint or budget limitations. For instance, in the context of e-commerce, an e-tailer, such as Amazon or Walmart .com, needs to choose a small subset out of tens of thousands of SKUs to display on its front item page when a user types in a search query, say for women's boots. Because the customer's ultimate purchasing decision is heavily influenced by the displayed contents, the e-tailer must incorporate its prediction of the customer's substitution behavior to maximize the sales probability and/or the expected revenue. Moreover, this personalized decision has to be made within a split second to minimize latency and to maximize customer experience. Consequently, in such applications, the e-tailer must deploy fast and provably good algorithms. The task of deciding on what personalized assortment should be offered to a customer given a search query is not restricted to the online retail industry. For example, airlines and hotels also face similar challenges when a customer is booking a flight or a hotel room online. Similarly, recommendation systems for sponsored search results that redirect traffic to other sites, such as Google or Facebook, can be thought of as solving an assortment-optimization problem to maximize the expected revenue generated from its recommendations. It is worth noting that the so-called expected revenue considered in the assortment-optimization literature is by no means limited to be actual revenue that a firm generates and could instead capture other proxy metrics, such as traffic or retention rate that the optimizer cares about.

Assortment optimization under cardinality or capacity constraints is, therefore, an important problem that has extensively been studied for many parametric choice models, including MNL (Davis et al. 2013, Désir et al. 2014), nested logit (Désir et al. 2014, Gallego and Topaloglu 2014, Feldman and Topaloglu 2015), and mixture of MNL models (Rusmevichientong et al. 2010, Désir et al. 2014, Feldman and Topaloglu 2017a).

Finally, although the constrained assortment-optimization problem studied in this paper has a seemingly simple and compact description, it is a fundamental problem that can be employed as a building block in more complex models of dynamic assortment optimization. In such problems, a sequence of item-display decisions is made over time, subject to inventory constraints. A much-needed technical tool in most algorithms that address these settings emerges because of the problem's natural restriction to single time periods, which boils down to solving a static assortment-optimization problem as a subroutine. Numerous recent papers (Zhang and Cooper 2005, Golrezaei et al. 2014, Gallego et al. 2016, Feldman and Topaloglu 2017b) have studied the complexity of dynamic assortment optimization, using a black-box algorithm for the single-period assortment-optimization problem. As a result, it is imperative that algorithms for the static subproblem would scale gracefully for large instances, while still providing provably good solutions.

One of the key challenges in designing an efficient algorithm for the capacity-constrained assortment-optimization problem under the Markov chain model arises from the fact that, unlike most parametric models, the choice probability function here does not have a simple functional form. Instead, the choice probability for any item i in an assortment S is specified as the solution to a system of linear equations that depends on S. This characterization results in a nonlinear mixed-integer programming formulation with a binary decision variable for each item indicating whether it is included in the assortment or not. In particular, the dependency of the system of linear equations on the assortment S in defining the choice probabilities results in nonlinear constraints in the integer formulation, making the latter particularly challenging to solve. In fact, in later parts of this paper, we formally establish APX-hardness for the capacity-constrained assortment problem. Our hardness results motivate us to study approximation algorithms with provably good guarantees.

1.1. Our Contributions

1.1.1. Assortment Optimization Under Cardinality and Capacity Constraints. We present fast approximation algorithms for the cardinality- and capacity-constrained assortment-optimization problems under the Markov chain choice model with provable worst-case approximation guarantees. In particular, we present a $(1/2-\epsilon)$-approximation for the cardinality-constrained problem and a $(1/3-\epsilon)$-approximation for the capacity-constrained problem for any $\epsilon > 0$, running in time polynomial in the input size and $1/\epsilon$. In other words, our algorithm for the cardinality-constrained assortment problem efficiently computes a solution with expected revenue that is provably at least $1/2-\epsilon$ times the optimal expected revenue for any instance. Similarly, the performance bound for the capacity-constrained version guarantees that the resulting expected revenue is at least $1/3-\epsilon$ times the optimal expected revenue.

Our algorithm is iterative and builds the assortment in steps. It is based on a new *externality-adjustment paradigm* that allows us to exactly capture the externality of adding an item to the assortment in any iteration. In particular, in each iteration t of the algorithm, we select an appropriate item, j_t, to add to the current solution and then construct a smaller modified instance of the problem using our externality-adjustment paradigm that perfectly captures the effect of adding j_t to the assortment on the remaining items. This approach allows us to focus on a smaller subproblem with only the remaining items in the modified instance. Therefore, our externality-adjustment paradigm enables us to *linearize* a highly nonlinear expected revenue function. As we demonstrate later, the expected revenue as a function of the assortment (or offer set) is neither monotone nor submodular in general and only satisfies the subadditivity property. Therefore, it is quite interesting and surprising to be able to obtain a linearization of such a function. Moreover, the iterative nature of our algorithm makes it particularly efficient. The number of iterations is upper bounded by the underlying number of items, and the computational effort made within each iteration is on par with that of computing the choice probability of a given offer set. As a result, the simplicity and speed of this algorithm makes it suitable for online retail applications.

1.1.2. Special Case: Uniform Item Prices. For the special case, when all items have identical prices, we show that the expected revenue function is both submodular and monotone. Therefore, we can obtain a $(1-1/e)$-approximation for the cardinality-constrained problem using a greedy algorithm (Nemhauser and Wolsey 1978). In fact, for this special case, we obtain an approximation ratio of $1-1/e$ under more general constraints, such as a constant number of capacity constraints (Kulik et al. 2013) and matroid constraints

(Calinescu et al. 2011). It is worth mentioning that, from a practical point of view, the uniform-price setting turns the objective function into that of maximizing sales probability. This scenario is very common when items are horizontally differentiated, that is, differ by characteristics that do not affect quality or price, such as iPads coming in a variety of colors or yogurt with different amounts of fat content.

1.1.3. Hardness of Approximation. We show that the capacity-constrained assortment-optimization problem under the Markov chain model is APX-hard; that is, it is NP-hard to approximate within a factor better than some given constant even when all items have uniform prices and unit weights. In this case, the capacity constraint reduces to a bound on the number of items, that is, to a cardinality constraint. Therefore, a constant factor approximation is the best possible result for the cardinality- and capacity-constrained assortment-optimization problems. Interestingly, although the unconstrained assortment-optimization problem under the Markov chain choice model can be solved optimally in polynomial time (Blanchet et al. 2016), the cardinality- and capacity-constrained problems are proven here to be APX-hard. In contrast, in both the MNL and nested logit models, the unconstrained assortment optimization and the cardinality-constrained assortment problems have the same complexity.

In addition, we show that assortment optimization under the Markov chain model with more general totally unimodular (TU) constraints on the assortment (generalizing cardinality constraints and capturing a wide range of practical constraints, such as precedence, display locations, and quality-consistent pricing constraints; Davis et al. 2013), is hard to approximate within a factor of $O(n^{1/2-\epsilon})$ for any fixed $\epsilon > 0$, where n is the number of items. This result drastically contrasts that of Davis et al. (2013), who prove that the assortment-optimization problem with TU constraints for the MNL model can be solved in polynomial time.

1.1.4. Computational Results. We conduct an extensive computational study focused on two main directions: (1) comparing the practical performance of our algorithms with their theoretical worst-case guarantees and (2) evaluating how well the Markov chain model performs when compared against the MNL model. For the first direction, we compare the performance of our algorithms with respect to the optimal solution that is computed by solving a mixed-integer programming (MIP) formulation. In these numerical experiments, we observe that the practical performance of our algorithms is significantly better than their theoretical worst-case guarantee. Specifically, although the worst-case approximation bound is $1/2 - \epsilon$ for the cardinality-constrained problem, we observe that the approximation ratio is 0.97, on average, and at least 0.77 across all instances considered. With respect to computational efficiency, our algorithm is scalable and terminates, on average, in a few seconds and within one minute in the worst case over all large instances tested (with $n = 200$). On the other hand, the MIP approach does not terminate even within a time limit of two hours on most of these instances. We further investigate different parameters that can potentially drive down the practical performance of our algorithm. We find that when there is a strong correlation in the model parameters and, simultaneously, the size of the constraint is moderate, the performance of our algorithm slightly degrades.

For the second direction, we conduct a numerical study to examine how the Markov chain model compares with the MNL model. In particular, we generate instances using an underlying mixture of MNL model as ground truth and compare the performance guarantees attained by both models. Here, we make use of a consideration set–inspired construction similar to that of Feldman and Topaloglu (2017b). For the constrained assortment problem, this question is particularly relevant because, despite the fact that the Markov chain model generalizes the MNL model, we compute an approximate solution for the Markov chain model, whereas an optimal solution can be obtained for the MNL model. That said, we observe that the Markov chain model significantly outperforms the MNL model and increases revenue by more than 12%, on average, in all settings tested.

1.2. Related Work

Assortment optimization under cardinality and capacity constraints has been studied widely in the literature for many parametric models. As we mentioned earlier, these constraints arise naturally in numerous applications, allowing one to model practical scenarios, such as a shelf-space constraint or budget limitations. Rusmevichientong et al. (2010) consider the cardinality-constrained assortment-optimization problem under the MNL model and propose an exact algorithm. Davis et al. (2013) provide an exact algorithm for assortment optimization under the MNL model for more general totally unimodular constraints that capture a wide range of practical constraints, such as precedence, display locations, and quality-consistent pricing constraints. Désir et al. (2014) show that the capacity-constrained assortment-optimization problem under the MNL model is NP-hard and present a fully polynomial-time approximation scheme (FPTAS) for this problem.

Gallego and Topaloglu (2014) present an exact algorithm for the cardinality-constrained problem for a special case of the nested logit model. Feldman and Topaloglu (2015) present an exact algorithm for the latter model when the cardinality constraint is across different nests. Rusmevichientong et al. (2010) devise a polynomial-time approximation scheme for the cardinality-constrained assortment problem under a mixture of MNL choice model. Désir et al. (2014) propose an FPTAS for the capacity-constrained assortment problem under both the nested logit and the mixture of MNL models for a constant number of mixtures. Feldman and Topaloglu (2017a) present an FPTAS for the capacity-constrained assortment optimization under the MNL model with nested consideration sets.

Furthermore, constrained assortment optimization has also been studied under the distribution over permutations models. Aouad et al. (2018) show that even unconstrained assortment optimization for general distribution over permutations is NP-hard to approximate within a factor of $O(n^{1-\epsilon})$ for any fixed $\epsilon > 0$. However, both unconstrained and constrained assortment-optimization problems have been studied for several structured distribution-over-permutations models. Farias et al. (2013) consider a distribution over permutations with the sparsest support that is consistent with the data, and Farias et al. (2011) present a local search–based algorithm for assortment optimization under this model. Désir et al. (2016) consider a mixture of Mallows model for choice and present an approximation scheme for the capacity-constrained assortment-optimization problem under a reasonable technical assumption. Paul et al. (2017) consider constrained assortment optimization under a nonparametric distribution over permutations model in which the permutations arise from a tree structure. Aouad et al. (2015) study constrained assortment optimization under consider-then-choose choice models. In summary, constrained assortment optimization has extensively been studied in the literature for a large class of parametric and nonparametric models.

In this paper, we focus on the cardinality- and capacity-constrained assortment-optimization problems under the Markov chain model. As we mentioned earlier, one of the key challenges arises because, unlike most of the models discussed, the choice probability function in the Markov chain model does not have a simple functional form and is given as a solution to a system of linear equations that depend on the assortment. This results in a highly nonlinear revenue function that is neither monotone nor submodular and only satisfies subadditivity. Consequently, none of the algorithms developed for constrained assortment optimization under other choice models appears to be useful in this setting.

Finally, the static assortment optimization model that this and many of the aforementioned papers consider can serve as a building block for a more elaborate dynamic assortment-optimization problem. In this setting, a sequence of item-display decisions is made over time subject to inventory constraints. Zhang and Cooper (2005), Feldman and Topaloglu (2017b), Golrezaei et al. (2014), and Gallego et al. (2016) have studied the dynamic assortment-optimization problem for different models of customer-arrival sequence, both adversarial and stochastic. Interestingly, when the dynamic problem is restricted to single time periods, it boils down to solving a static assortment-optimization problem as a subroutine. Although the formulations and algorithmic approaches considered in the aforementioned papers work for a broad class of choice models, Feldman and Topaloglu (2017b) focus their analysis on the Markov chain choice model. They show that when all items share a single resource constraint, the dynamic *unconstrained* assortment-optimization problem can be efficiently solved to optimality based on dynamic programming (DP). To evaluate the DP recursion, the authors make use of an algorithm for the static assortment-optimization problem as a subroutine. For the general network revenue-management setting, in which mapping of items to resource consumption forms a bipartite network, Feldman and Topaloglu (2017b) exploit the structure of the unconstrained assortment-optimization problem to reduce the size of the large-scale LP that needs to be solved.

It is worth pointing out that, unlike the resource constraints considered by Feldman and Topaloglu (2017b), which model the inventory dynamics of raw materials over time, our paper considers capacity limitations on the size of the offer set. From a computational complexity view, the two sets of constraints are also very different. For instance, the single-resource revenue-management problem can be solved exactly in polynomial time provided that the underlying static unconstrained assortment-optimization problem can be solved exactly. In contrast, we show that the static assortment-optimization problem with cardinality constraints is already APX-hard. In Online Appendix B, we provide more details on how adding a cardinality constraint really affects the fundamental nature of the optimization problem and, in particular, why previous work (such as that of Feldman and Topaloglu 2017b) does not carry over to the network revenue-management problem with an additional cardinality constraint. We also show how our algorithmic approach can be used in a dynamic setting to attain a constant factor approximation algorithm for the latter problem. In this sense, our work complements the existing literature on dynamic assortment-optimization problems.

1.3. The Markov Chain Model and Additional Notation

We denote the universe of n items by the set $\mathcal{N} = \{1, 2, \ldots, n\}$ and the no-purchase option by zero with the convention that $\mathcal{N}_+ = \mathcal{N} \cup \{0\}$. We consider a Markov chain $\mathcal{M}$ with states $\mathcal{N}_+$ to model the substitution behavior of customers. This model is completely specified by initial arrival probabilities λ_i for all states $i \in \mathcal{N}_+$ and by the transition probabilities ρ_{ij} for all $(i, j) \in \mathcal{N}_+^2$.

When a retailer chooses to offer a subset of items S to consumers, the corresponding states of the Markov chain become absorbing states. A customer arrives in state i with probability λ_i; if this state is not absorbing, the customer transitions to a different state $j \neq i$, and the process continues until the customer reaches an absorbing state, which is then purchased. In other words, the probability of a random customer purchasing item i with S being the offer set of items is the probability that the customer reaches state i before any other absorbing states in the underlying Markov chain.

Following Blanchet et al. (2016), we assume that for each state $j \in \mathcal{N}$ there is a path to state zero with nonzero probability. For a given offer set $S \subseteq \mathcal{N}$, let $\pi(i, S)$ be the probability that item i is chosen when the assortment S is offered. Let p_i denote the price of item i. For any assortment S, its expected revenue can be written as

$$R(S) = \sum_{i \in S} \pi(i, S) \cdot p_i.$$

For any (possibly empty) pairwise-disjoint subsets $U, V, W \subseteq \mathcal{N}_+$, let $\mathbb{P}_j(U \prec V \prec W)$ denote the probability that, starting from j, we first visit some state in U before visiting any state in $V \cup W$ and subsequently visit some state in V before visiting any state in W with respect to the transition probabilities of $\mathcal{M}$. Let $\mathbb{P}(U \prec V \prec W) = \sum_{j=1}^{n} \lambda_j \mathbb{P}_j(U \prec V \prec W)$. With this notation, we can write $\pi(i, S) = \mathbb{P}(i \prec S_+ \backslash \{i\})$, where $S_+ = S \cup \{0\}$ for all $S \subseteq \mathcal{N}$ (in this case, $W = \emptyset$).

1.4. Outline

The remainder of this paper is organized as follows. For ease of exposition, we first consider the special case of uniform-price items in Section 2. We also illustrate why several greedy algorithms, including the one that is provably good for uniform prices, do not provide nontrivial approximations for arbitrary prices. In Sections 3 and 4, we present the externality-adjustment paradigm and our algorithm for the cardinality-constrained problem. We describe the extension to the capacity-constrained problem in Section 5. In Section 6, we establish our hardness results for the constrained assortment-optimization problem under the Markov chain model. Finally, our computational study is presented in Section 7.

2. Revenue Function Properties and Key Challenges

In this section, we examine several structural properties of the revenue function and discuss the key challenges in designing an algorithm for the cardinality- or capacity-constrained versions of the assortment-optimization problem under the Markov chain model. We also consider the special case when the item prices are uniform. This setting is quite common when items are horizontally differentiated, that is, differ by characteristics that do not affect quality or price, such as iPads coming in a variety of colors or yogurt with different amounts of fat content. We show that, for this special case, the revenue function is both monotone and submodular, implying that the cardinality- and capacity-constrained assortment-optimization problems in this setting can be efficiently approximated within a factor of $1 - 1/e$. With general prices, we show that the revenue function is neither monotone nor submodular and that several natural variants of the greedy algorithm can lead to solutions with arbitrarily bad performance in comparison with the optimal solution.

2.1. Uniform Prices: Constant Factor Approximation

We start by formally defining the notions of monotonicity and submodularity.

Definition 1. A revenue function $R : 2^{\mathcal{N}} \to \mathbb{R}_+$ is monotone when, for all $S \subseteq \mathcal{N}$ and $i \in \mathcal{N}$, we have $R(S \cup \{i\}) \geq R(S)$.

Definition 2. A revenue function $R : 2^{\mathcal{N}} \to \mathbb{R}_+$ is submodular when, for all $S \subseteq T \subseteq \mathcal{N}$ and $i \in \mathcal{N} \backslash T$, we have $R(S \cup \{i\}) - R(S) \geq R(T \cup \{i\}) - R(T)$.

When all prices are equal, we show that the revenue function is both submodular and monotone.

Theorem 1. *When all items have uniform prices, the revenue function $R(\cdot)$ is submodular and monotone.*

Proof. Let p be the price of every item in $\mathcal{N}$. Because item prices are identical, for every subset S and item $i \in \mathcal{N} \backslash S$, we have

$$R(S \cup \{i\}) = R(S) + p \cdot \mathbb{P}(i \prec 0 \prec S).$$

To understand this equation, recall that $\mathbb{P}(i \prec 0 \prec S)$ is the probability that the Markov chain visits state i and then visits state 0 without visiting any state in S. When all prices are equal, the marginal increase in revenue by adding item i is only a result of the additional demand that item i is able to capture. Consequently, $R(\cdot)$ is monotone as the quantity $p \cdot \mathbb{P}(i \prec 0 \prec S)$ is nonnegative. Moreover, the submodularity of R holds because, for all $S \subseteq T$,

$$\begin{aligned} R(S \cup \{i\}) - R(S) &= p \cdot \mathbb{P}(i \prec 0 \prec S) \geq p \cdot \mathbb{P}(i \prec 0 \prec T) \\ &= R(T \cup \{i\}) - R(T). \end{aligned} \qquad \square$$

Consequently, by the classical result of Nemhauser and Wolsey (1978), we know that the greedy algorithm, in which we iteratively add the item that increases the objective value the most, guarantees a $(1 - 1/e)$-approximation for (Cardinality-Assort) with uniform prices. We refer to this procedure as the *incremental greedy algorithm*.

2.1.1. More General Constraints. As the revenue function is monotone and submodular for uniform prices, we can exploit the existing machinery for approximately maximizing monotone submodular functions subject to a wide range of constraints (see, for instance, Lee et al. 2010, Calinescu et al. 2011, Kulik et al. 2013, and Buchbinder et al. 2014). This way, constant factor approximations can be obtained for assortment optimization under the Markov chain model with more general constraints on the set of items offered. For instance, Kulik et al. (2013) give a $(1-1/e)$-approximation for maximizing a monotone submodular function under a fixed number of capacity (knapsack) constraints, and Calinescu et al. (2011) obtain a similar performance guarantee for maximizing a monotone submodular function under a matroid constraint.

2.2. Challenges in Extensions to Arbitrary Prices

We begin by observing that, for arbitrary prices, the expected revenue as a function of the assortment is neither monotone nor submodular under the Markov chain choice model. The proof of the next claim appears in Appendix A.1.

Lemma 1. *When items have arbitrary prices, the revenue function $R(\cdot)$ is neither monotone nor submodular.*

The examples we use to prove Lemma 1, although very simple, illustrate the fact that, conditioned on having high-price items in an assortment, adding lower-priced items may cannibalize the demand going into high-price ones. As a result, adding items to an existing assortment does not necessarily increase its expected revenue (monotonicity). Similarly, it is not the case that the bigger the assortment gets, the lesser of the impact additional items would have on the revenue function (submodularity). In what follows, we identify the drawbacks of the incremental greedy algorithm in approximating (Cardinality-Assort) for arbitrary prices as well as that of a modified greedy heuristic.

2.2.1. Incremental Greedy Algorithm. The incremental greedy algorithm that attains a $(1-1/e)$-approximation for the cardinality-constrained assortment-optimization problem with uniform prices does not extend to the more general setting with arbitrary prices. We formalize this intuition through the following result.

Lemma 2. *The approximation ratio of the incremental greedy algorithm is $\Theta(1/k)$ for* (Cardinality-Assort).

Intuitively, the performance of the incremental greedy algorithm for general prices can be highly suboptimal because of potentially making a low-price item absorbing, thereby blocking all probabilistic transitions going into high-price items. This intuition is formalized in Appendix A.2, in which we present the proof of Lemma 2.

2.2.2. Modified Greedy Algorithm. The bad example for the incremental greedy algorithm illustrates that we may have been too focused on local improvements in each iteration without taking into account the information of the entire network induced by the probability transition matrix or the number of remaining iterations. Therefore, we consider a modified greedy algorithm that accounts for the Markov chain structure by using the optimal solution to the unconstrained assortment problem, in which there is no restriction on the number of items picked. This solution can be computed in polynomial time via an algorithm proposed by Blanchet et al. (2016) (as a side note, we give an alternative algorithm for the unconstrained problem in Section 3.4). Intuitively, the items picked by the unconstrained optimal assortment should not block each other's demand too much. Let U^* be the optimal unconstrained assortment whose associated revenue can be written as

$$R(U^*) = \sum_{i \in U^*} \mathbb{P}(i \prec U^*_+ \backslash \{i\}) \cdot p_i. \tag{1}$$

A natural candidate algorithm takes the k states with the largest $\mathbb{P}(i \prec U^*_+ \backslash \{i\}) \cdot p_i$ value within the unconstrained optimal solution U^* and sets these states to be absorbing. We show that even the modified greedy algorithm performs poorly in the worst case.

Lemma 3. *The approximation ratio of the modified greedy algorithm is $\Theta(k/n)$ for* (Cardinality-Assort).

The poor performance of the modified greedy algorithm illustrates that an optimal assortment for the constrained problem may be very different from that of its unconstrained counterpart. Hence, searching within an unconstrained optimal solution for a good approximate solution to the constrained problem can be unfruitful in general. The proof of Lemma 3 is presented in Appendix A.3.

2.2.3. Summary. As one can observe in the aforementioned examples, the incremental greedy algorithm places too much emphasis on including the item that would result in the highest incremental revenue while ignoring the fact that such an item could have a low price and would irreversibly cannibalize most of the customer demand. On the other hand, the modified greedy algorithm places too much emphasis on including high-price items even though each item captures very little demand. Clearly, a competitive algorithm needs to properly address this revenue/price trade-off. However, naively taking the best of the two greedy algorithms does not do well either. The analysis of the two greedy variants provides important insights that we utilize toward designing a provably good algorithm. The main insight behind our algorithm comes from a paradigm that enables us to quantify the revenue effect by which including a new item in our assortment cannibalizes the demand to other items in the current assortment.

3. Externality Adjustment–based Algorithm Design

In this section, we present the general framework of our approximation algorithms for the cardinality- and capacity-constrained assortment-optimization problems under the Markov chain model.

3.1. High-Level Ideas

As the example in Figure A.1 illustrates, the incremental greedy algorithm could end up with a highly suboptimal solution because of picking cannibalizing items, that is, those blocking the demand for higher-price items. Picking the highest-price item will eliminate such a concern. However, a high-price item might only capture very little demand and, therefore, generate very small revenue as illustrated by the example in Figure A.2. In the presence of a capacity constraint on the assortment, picking such items may not be an optimal use of the capacity. This motivates us to choose the highest-price item in an appropriate consideration set. Intuitively, the consideration set will consist of items that generate sufficiently high incremental revenue.

We first give a high-level description of our algorithm ALG that builds the solution iteratively. Let $\mathcal{M}_t$ denote the problem instance in any iteration t. The algorithm performs the following two steps in each iteration t:

1. *Item selection.* Define an appropriate consideration set C_t of items and pick the highest-price item from C_t.
2. *Instance update (externality adjustment).* Construct a new instance, $\mathcal{M}_{t+1}$, of the constrained assortment-optimization problem with appropriately modified item prices and transition probabilities such that

$$\mathsf{ALG}(\mathcal{M}_t) = \Delta_t + \mathsf{ALG}(\mathcal{M}_{t+1}),$$

where $\mathsf{ALG}(\cdot)$ is the revenue of the solution obtained by the algorithm on a given instance and Δ_t is the incremental revenue in the objective value because of the item selected in iteration t.

The instance update step linearizes the revenue function even though the original revenue function is nonlinear, which is crucial for our iterative-solution approach. The update rule is a framework to capture the externality of our item selection in each iteration on the remaining items. To completely specify the algorithm, we need to provide a precise definition for the consideration set in the item-selection step and for the instance-update step. For both cardinality- and capacity-constrained assortment-optimization problems, the instance update step is similar as explained in Section 3.2. The consideration set, however, depends on the particular optimization problem being considered and is defined later. The intuition is to include items whose incremental revenue is above an appropriately chosen threshold.

Our algorithm can also be viewed in a local-ratio framework (see, for instance, Bar-Yehuda and Even 1985, Bar-Yehuda et al. 2005, and Bar-Yehuda and Rawitz 2006). We would like to note that the local-ratio framework does not provide a general recipe for designing an update rule or for analyzing the performance bound. In most algorithms belonging to this framework, the update rule follows from a primal-dual algorithm. However, for the capacity-constrained assortment-optimization problem under the Markov chain model, we are not aware of any good LP formulation, and therefore, the instance-update rule requires new ideas.

3.2. Instance Update or Externality Adjustment

3.2.1. Notation. Given an instance $\mathcal{M}$ of the Markov chain model, we define an updated instance $\mathcal{M}(S)$, given that the set of states S is made absorbing by modifying the item prices as well as the probability-transition matrix. Because we index the updates by a set S, the instance $\mathcal{M}_t$ introduced in the preceding discussion should be thought of as $\mathcal{M}(S_{t-1})$, where S_{t-1} denotes the set of items picked up to (and including) step $t-1$. For an instance $\mathcal{M}(S)$, we denote by p_i^S the updated price of item i and by ρ_{ij}^S the updated transition probabilities for every $i \in \mathcal{N}$, $j \in \mathcal{N}_+$; the arrival rate to any state remains unchanged; that is, $\lambda_i^S = \lambda_i$ for all $i \in \mathcal{N}$. We also denote by $R^S : 2^{\mathcal{N}} \to \mathbb{R}$ the revenue function associated with the instance $\mathcal{M}(S)$ and by $\mathbb{P}^S(\cdot)$ the probability of any event with respect to $\mathcal{M}(S)$.

3.2.2. Price Update. First, we introduce the price updates such that, when S is made absorbing, we account for the revenue generated by every state $j \in S$. To this end, consider a unit demand at state $i \notin S$. This unit demand generates a revenue of p_i when i is made absorbing. On the other hand, when i is not absorbing, this unit demand at i generates a revenue of

$$\sum_{j \in S} \mathbb{P}_i(j \prec S_+ \backslash \{j\}) \cdot p_j.$$

This revenue, which was already accounted for by S, is lost when i is also made absorbing in addition to S. Hence, the net revenue per unit demand at i when we make it absorbing, given that S is already absorbing, is

$$p_i - \sum_{j \in S} \mathbb{P}_i(j \prec S_+ \backslash \{j\}) p_j,$$

which we denote as the adjusted price p_i^S. This update is explicitly described in Figure 1. Now, two important remarks are in place:

- The adjusted prices can be negative, corresponding to the situation in which adding an item decreases the overall revenue.

- The probabilities $\mathbb{P}_i(j \prec S_+ \backslash \{j\})$, needed for our price updates, can be interpreted as the choice probability $\pi(j, S)$ for a modified instance with $\lambda_i = 1$ and $\lambda_\ell = 0$ for $\ell \neq i$. Therefore, these quantities can be efficiently computed via traditional Markov chain tools (see, for instance, Blanchet et al. 2016).

3.2.2. Transition Probabilities Update. Because the subset of states S is set to be absorbing, we simply redirect the outgoing probabilities from all states in S to zero as described in Figure 1.

3.3. Structural Properties of the Updates

We first show that the externality-adjustment updates allow us to linearize the revenue function.

Lemma 4. *$R(S_1 \cup S_2) = R(S_1) + R^{S_1}(S_2)$ for every $S_1, S_2 \subseteq \mathcal{N}$.*

Proof. We assume without lost of generality that $S_1 \cap S_2 = \emptyset$ because all items in $S_1 \cap S_2$ have zero as their adjusted price, and we can then apply the proof to $S_2 \backslash S_1$. Using the definition of the externality-adjustment updates, we have

$$\begin{aligned} R^{S_1}(S_2) &= \sum_{i \in S_2} \mathbb{P}^{S_1}(i \prec S_{2+} \backslash \{i\}) p_i^{S_1} \\ &= \sum_{i \in S_2} \mathbb{P}^{S_1}(i \prec S_{2+} \backslash \{i\}) \cdot \left(p_i - \sum_{j \in S_1} \mathbb{P}_i(j \prec S_{1+} \backslash \{j\}) p_j \right) \\ &= \sum_{i \in S_2} \mathbb{P}^{S_1}(i \prec S_{2+} \backslash \{i\}) p_i \\ &\quad - \sum_{j \in S_1} \sum_{i \in S_2} \mathbb{P}^{S_1}(i \prec S_{2+} \backslash \{i\}) \mathbb{P}_i(j \prec S_{1+} \backslash \{j\}) p_j. \end{aligned}$$

By definition of the transition probabilities ρ^{S_1}, note that all items of S_1 are redirected to zero. This, together with the fact that $S_1 \cap S_2 = \emptyset$, implies that for all $i \in S_2$ we have $\mathbb{P}^{S_1}(i \prec S_{2+} \backslash \{i\}) = \mathbb{P}(i \prec (S_2 \cup S_1)_+ \backslash \{i\})$. Consequently,

$$\begin{aligned} &R(S_1) + R^{S_1}(S_2) \\ &= \sum_{j \in S_1} \Bigg(\mathbb{P}(j \prec S_{1+} \backslash \{j\}) \\ &\qquad - \sum_{i \in S_2} \mathbb{P}(i \prec (S_2 \cup S_1)_+ \backslash \{i\}) \mathbb{P}_i(j \prec S_{1+} \backslash \{j\}) \Bigg) p_j \\ &\quad + \sum_{i \in S_2} \mathbb{P}(i \prec (S_2 \cup S_1)_+ \backslash \{i\}) p_i \\ &= \sum_{j \in S_1} \left(\mathbb{P}(j \prec S_{1+} \backslash \{j\}) - \mathbb{P}(S_2 \prec j \prec S_{1+} \backslash \{j\}) \right) p_j \\ &\quad + \sum_{i \in S_2} \mathbb{P}(i \prec (S_2 \cup S_1)_+ \backslash \{i\}) p_i \\ &= \sum_{j \in S_1} \mathbb{P}(j \prec (S_2 \cup S_1)_+ \backslash \{j\}) p_j \\ &\quad + \sum_{i \in S_2} \mathbb{P}(i \prec (S_2 \cup S_1)_+ \backslash \{i\}) p_i \\ &= R(S_1 \cup S_2), \end{aligned}$$

where the second equality holds because

$$\begin{aligned} &\sum_{i \in S_2} \mathbb{P}(i \prec (S_2 \cup S_1)_+ \backslash \{i\}) \mathbb{P}_i(j \prec S_{1+} \backslash \{j\}) \\ &\quad = \mathbb{P}(S_2 \prec j \prec S_{1+} \backslash \{j\}), \end{aligned}$$

as by the Markov property, both the left and right terms in this equality denote the probability that we visit some state in S_2 before any state in S_{1+}, followed by state $j \in S_1$ before any other state in S_{1+}. □

Figure 1. Instance Update in Our Algorithm

Price update:

$$p_i^S = \begin{cases} 0, & \text{if } i \in S \\ p_i - \sum_{j \in S} \mathbb{P}_i(j \prec S_+ \backslash \{j\}) p_j, & \text{otherwise.} \end{cases}$$

Transition probabilities update:

$$\rho_{ij}^S = \begin{cases} 1, & \text{if } i \in S \text{ and } j = 0 \\ 0, & \text{if } i \in S \text{ and } j \neq 0 \\ \rho_{ij}, & \text{otherwise.} \end{cases}$$

The next lemma shows that the composition of two externality-adjustment updates over subsets S_1 and S_2 is equivalent to a single externality-adjustment update over $S_1 \cup S_2$. This property is crucial for repeatedly applying externality-adjustment updates.

Lemma 5. *Let $S_1 \subseteq \mathcal{N}$ be some assortment, and let $\mathcal{M}_1 = \mathcal{M}(S_1)$. For any S_2 with $S_1 \cap S_2 = \emptyset$, the instance $\mathcal{M}_1(S_2)$ is identical to the instance $\mathcal{M}(S_1 \cup S_2)$ in terms of item prices and transition probabilities.*

To establish this result, it suffices to verify that $(p_i^{S_1})^{S_2} = p_i^{S_1 \cup S_2}$ for all S_1, S_2 and $i \notin S_1 \cup S_2$ as this identity clearly holds for the transition-matrix updates. The proof is similar to that of Lemma 4 and is presented in Appendix A.4. Putting the previous two lemmas together gives the following claim.

Lemma 6. *$R^{S_1}(S_2 \cup S_3) = R^{S_1}(S_2) + R^{S_1 \cup S_2}(S_3)$ for any pairwise-disjoint sets $S_1, S_2, S_3 \subseteq \mathcal{N}$.*

3.4. Warm-up: Exact Algorithm for the Unconstrained Problem

As a warm-up, we first present an alternative exact algorithm for the unconstrained assortment-optimization problem under the Markov chain model by using the externality-adjustment framework. Our algorithm is based on the observation that it is always optimal to offer the highest-price item for the unconstrained problem as it does not cannibalize the demand of other items. The latter property is implied by a slightly more general claim, formalized as follows. For any $x \in \mathbb{R}$, let $[x]^+ = \max(x, 0)$.

Lemma 7. *Let $S \subseteq \mathcal{N}$. For any item $i \notin S$ with price $p_i \geq [\max_{j \in S} p_j]^+$, we have $R(S \cup \{i\}) \geq R(S)$.*

Proof. By Lemma 4,

$$R(S \cup \{i\}) = R(S) + R^S(\{i\}) = R(S) + \mathbb{P}^S(i \prec 0) \cdot p_i^S,$$

meaning that to prove the claim it remains to show $p_i^S \geq 0$. Indeed, by definition of the updated price in Figure 1,

$$p_i^S = p_i - \sum_{j \in S} \mathbb{P}_i(j \prec S_+ \setminus \{j\}) \cdot p_j \geq 0,$$

where the last inequality holds because $p_i \geq [\max_{j \in S} p_j]^+$. □

3.4.1. Algorithm Overview. Based on Lemma 7, we present an alternative exact algorithm for the unconstrained assortment-optimization problem under the Markov chain model. At a high level, our algorithm builds an assortment iteratively. In each step, we select the highest adjusted-price item (breaking ties arbitrarily) and update the prices and transition probabilities according to the externality-adjustment updates described in Figure 1. This selection and updating process is repeated until all adjusted prices are nonpositive.

3.4.2. Algorithm and Analysis. The specifics of our approach are formally described in Algorithm 1. Note that this algorithm indeed falls within the general framework of Sections 3.1 and 3.2 by defining the consideration set in each iteration to be the entire set of items.

Algorithm 1 (Algorithm for Unconstrained Assortment)

1: Let S be the set of states picked so far, starting with $S = \emptyset$.
2: While there exists $i \in \mathcal{N} \setminus S$ such that $p_i^S \geq 0$:
  a. Let i^* be the item for which p_i^S is maximized, breaking ties arbitrarily.
  b. Add i^* to S.
3: Return S.

Theorem 2. *Algorithm 1 computes an optimal solution for the unconstrained assortment-optimization problem under the Markov chain model.*

Proof. The correctness of Algorithm 1 is based on the observation that it is always optimal to offer the highest adjusted price item as long as this price is nonnegative. Suppose item 1 is the highest-price item. From Lemma 7, we get $R(S \cup \{1\}) \geq R(S)$ for any assortment S. Therefore, we can assume that item 1 belongs to the optimal assortment. By Lemma 4, we can write

$$\max_{S \subseteq \mathcal{N}} R(S) = R(\{1\}) + \max_{S' \subseteq \mathcal{N} \setminus \{1\}} R^{\{1\}}(S').$$

It remains to show that, when we get to an iteration in which our current absorption set is X and the adjusted price of every state in the modified instance $\mathcal{M}(X)$ is nonpositive, then X is an optimal solution to $\mathcal{M}$. To see this, by repeated applications of Lemmas 4 and 5, we have

$$\max_{S \subseteq \mathcal{N}} R(S) = R(X) + \max_{S' \subseteq \mathcal{N} \setminus X} R^X(S').$$

However, because the adjusted price of every state in the instance $\mathcal{M}(X)$ is nonpositive, we must have $R^X(S') \leq 0$ for all $S' \subseteq \mathcal{N} \setminus X$. Hence, it is optimal not to make any state in $\mathcal{M}(X)$ absorbing, which implies that X is an optimal solution to $\mathcal{M}$. □

3.4.3. Implications. Our algorithm provides interesting insights for some known results about the optimal-stopping problem and the assortment optimization under the MNL choice model. Blanchet et al. (2016) relate the unconstrained assortment problem to the optimal stopping time on a Markov chain (Chow et al. 1971). In this problem, we need to decide at each state i whether to stop and get the reward p_i or to proceed according to the transition probabilities of the Markov chain. Moreover, there is an absorbing state 0 with price $p_0 = 0$. Algorithm 1 for the unconstrained assortment-optimization problem gives an alternative strongly polynomial-time algorithm for this optimal-stopping problem.

Blanchet et al. (2016) also prove that the MNL choice model is a special case of the Markov chain–based model. Therefore, by analyzing Algorithm 1 to solve the assortment-optimization problem under the MNL model, we can recover the structure of the optimal assortment being nested by price; that is, the optimal assortment consists of the top ℓ-priced items for some ℓ. Additional details on this application, including an explicit expression for our externality-adjustment updates, are given in Appendix B.

4. Cardinality-Constrained Assortment Optimization

In this section, we present a $(1/2 - \epsilon)$-approximation for the cardinality-constrained assortment-optimization problem under the Markov chain model for any fixed $\epsilon > 0$. Following the externality-adjustment framework described in Section 3, our algorithm for the cardinality-constrained case also selects in each step a state with a high adjusted price from an appropriate consideration set. The latter set is defined to avoid picking states that have a high adjusted price but capture very little demand. This effect is ensured by only considering items whose incremental revenue exceeds a certain threshold.

4.1. The Algorithm

4.1.1. Overview. Before diving into the fine technical details, we give an informal high-level overview. Our algorithm builds an assortment iteratively in a similar spirit to Algorithm 1. The main difference resides in the selection step. Specifically, in each step, we select the highest nonpositive adjusted price item *only* among items whose marginal revenue increment with respect to the assortment picked so far exceeds some predetermined threshold. We then proceed by updating the prices and transition probabilities according to the externality-adjustment updates described in Figure 1. This selection and updating process is repeated until either k items have been picked or all remaining items have insufficient marginal increments.

4.1.2. Algorithm. Our algorithm is iterative and selects a single item in each step, following the framework described in Section 3.1. Let S_t be the set of selected items by the end of step t, starting with $S_0 = \emptyset$. We use σ_t to denote the item picked in step t, meaning that $S_t = \{\sigma_1, \ldots, \sigma_t\}$. At every step $t \geq 1$, we select the highest adjusted price item (with respect to $p^{S_{t-1}}$, breaking ties arbitrarily) among items in the following consideration set:

$$C_t = \left\{ i \in \mathcal{N} \backslash S_{t-1} : R^{S_{t-1}}(\{i\}) \geq \alpha \frac{R(S^*)}{k} \right\},$$

where S^* is the optimal solution, k is the cardinality bound, and $\alpha \in (0,1)$ is a parameter whose value will be optimized later. Note that C_t is defined at the beginning of step t, whereas S_t is defined at the end of step t and includes the item selected in this step. Once the item σ_t is selected, we recompute the adjusted prices via the externality-adjustment update described in Figure 1 and update the consideration set to get C_{t+1}. The algorithm terminates when either k items have already been picked (i.e., upon the completion of step k) or when the consideration set C_t becomes empty.

4.1.3. Guessing the Value of $R(S^*)$. Because the optimal revenue $R(S^*)$ is not known a priori, we explain how the value of $R(S^*)$ is approximately guessed to complete the algorithm's description. A natural upper bound on $R(S^*)$ is $R(U^*)$, where U^* is the optimal unconstrained solution. On the other hand, by Lemma 3, we know that $R(S^*) \geq \frac{k}{n} \cdot R(U^*)$. Now, given an accuracy parameter $0 < \epsilon < 1$, let

$$B_j = \frac{k}{n} \cdot R(U^*) \cdot (1+\epsilon)^j, \qquad j = 1, \ldots, J, \qquad (2)$$

where $J = \min\{j \in \mathbb{N} : B_j \geq R(U^*)\} = O(\frac{\log(n/k)}{\epsilon})$. For each guess B_j for the true value of $R(S^*)$, we run the algorithm and eventually return the best solution found over all guesses. Algorithm 2 describes the resulting procedure for a particular choice of B_j and threshold α for the consideration set. Algorithm 3 describes the full procedure for any given $\epsilon > 0$.

Algorithm 2 (Algorithm with Guess B_j and Threshold α)

1: Let S be the set of states picked so far, starting with $S = \emptyset$.
2: For all $S \subseteq \mathcal{N}$, let $C(S) = \{i \in \mathcal{N} \backslash S : R^S(\{i\}) \geq \frac{\alpha \cdot B_j}{k}\}$.
3: While $|S| < k$ and $C(S) \neq \emptyset$:
  a. Let i^* be the item of $C(S)$ for which p_i^S is maximized, breaking ties arbitrarily.
  b. Add i^* to S.
4: Return S.

Algorithm 3 (Algorithm for (Cardinality-Assort) with threshold α)

1: Given an error parameter $\epsilon > 0$, let J and $\{B_j\}_{j \in [J]}$ be defined according to (2).
2: For $j \in [J]$, let S_j be the solution returned by Algorithm 2 with guess B_j and threshold α.
3: Return $\arg\max_{j \in [J]} R(S_j)$.

4.2. Technical Lemmas

Prior to analyzing the performance guarantee of our algorithm, we present two technical lemmas. We first prove that our revenue function is subadditive.

Lemma 8. *For all $S_1, S_2 \subseteq \mathcal{N}$ consisting only of nonnegative priced items, $R(S_1 \cup S_2) \leq R(S_1) + R(S_2)$.*

Proof. We have that

$$\begin{aligned} R(S_1 \cup S_2) &= \sum_{j \in S_1} \mathbb{P}(j \prec (S_1 \cup S_2)_+ \backslash \{j\}) \cdot p_j \\ &\quad + \sum_{j \in S_2 \backslash S_1} \mathbb{P}(j \prec (S_1 \cup S_2)_+ \backslash \{j\}) \cdot p_j \\ &\leq \sum_{j \in S_1} \mathbb{P}(j \prec (S_1)_+ \backslash \{j\}) \cdot p_j \\ &\quad + \sum_{j \in S_2} \mathbb{P}(j \prec (S_2)_+ \backslash \{j\}) \cdot p_j \\ &= R(S_1) + R(S_2). \qquad \square \end{aligned}$$

Next, we establish a technical lemma that allows us to compare the revenue of the optimal solution $R(S^*)$ with the revenue of the set returned by our algorithm, $R(S_t)$. First, note that the consideration sets along different steps are nested (i.e., $C_1 \supseteq C_2 \supseteq \cdots$). Therefore, once an item disappears from the consideration set, it never reappears. This enables us to partition the items of S^* according to the moment they disappear from the consideration set (because either their adjusted revenue becomes too small or they get picked by the algorithm). More precisely, letting $Z_0 = S^*$, for all $t \geq 1$ we define the following sets:

- $Z_t = S^* \cap C_t$ denotes the items of S^* that are in the consideration set C_t.
- $Y_t = Z_{t-1} \backslash Z_t$ denotes the items of S^* that disappear from the consideration set during step $t-1$.
- $Y_t^+ = \{i \in Y_t : p_i^{S_{t-1}} \geq 0\}$ denotes the items of Y_t that have a nonnegative adjusted price at step t.

Note that these sets are all defined at the beginning of step t. The following lemma relates the adjusted revenue of items in Z_{t-1} and Z_t in terms of the marginal change in revenue, $R(S_t) - R(S_{t-1})$.

Lemma 9. *For all $t \geq 1$, $R(S_t) - R(S_{t-1}) \geq R^{S_{t-1}}(Z_t) - (R^{S_t}(Z_{t+1}) + R^{S_t}(Y_{t+1}^+))$.*

Proof. Recall that, by definition, Z_t contains the items of S^* that are in the consideration set at the beginning of step t. Because our algorithm picks the highest adjusted price item, σ_t, in the consideration set C_t, we have $p^{S_{t-1}}_{\sigma_t} \geq p^{S_{t-1}}_i \geq 0$ for all items $i \in Z_t$. Therefore, by Lemma 7,

$$R^{S_{t-1}}(Z_t) \leq R^{S_{t-1}}(Z_t \cup \{\sigma_t\}). \tag{3}$$

We now consider two cases, depending on whether the item σ_t appears in the optimal solution S^* or not.

Case (a). $\sigma_t \notin S^*$. From Lemma 6, $R^{S_{t-1}}(Z_t \cup \{\sigma_t\}) = R^{S_{t-1}}(\{\sigma_t\}) + R^{S_t}(Z_t)$. Consequently, from inequality (3), we have

$$\begin{aligned} R^{S_{t-1}}(Z_t) &\leq R^{S_{t-1}}(\{\sigma_t\}) + R^{S_t}(Z_t) \\ &= R^{S_{t-1}}(\{\sigma_t\}) + R^{S_t}(Z_{t+1} \cup Y_{t+1}) \\ &\leq R^{S_{t-1}}(\{\sigma_t\}) + R^{S_t}(Z_{t+1} \cup Y^+_{t+1}) \\ &\leq R^{S_{t-1}}(\{\sigma_t\}) + R^{S_t}(Z_{t+1}) + R^{S_t}(Y^+_{t+1}), \end{aligned}$$

where the second inequality holds because removing all negative adjusted price items can only increase net revenue and the last inequality follows from Lemma 8. Adding $R(S_{t-1})$ on both sides of the inequality yields the desired inequality by Lemma 4.

Case (b). $\sigma_t \in S^*$. From Lemma 6, $R^{S_{t-1}}(Z_t) = R^{S_{t-1}}(\{\sigma_t\}) + R^{S_t}(Z_t \backslash \{\sigma_t\})$. Then, similar to the previous case, we have

$$\begin{aligned} R^{S_t}(Z_t \backslash \{\sigma_t\}) &\leq R^{S_t}((Z_{t+1} \cup Y^+_{t+1}) \backslash \{\sigma_t\}) \\ &\leq R^{S_t}(Z_{t+1}) + R^{S_t}(Y^+_{t+1} \backslash \{\sigma_t\}). \end{aligned}$$

Note that $R^{S_t}(Y^+_{t+1} \backslash \{\sigma_t\}) = R^{S_t}(Y^+_{t+1})$ because $p^{S_t}_{\sigma_t} = 0$ and σ_t is an absorbing state in $\mathcal{M}(S_t)$. Adding $R(S_{t-1})$ on both sides of the inequality concludes the proof. □

From this result, we obtain the following claim.

Lemma 10. *For all $t \geq 0$, we have $R(S_t) \geq R(S^*) - (R^{S_t}(Z_{t+1}) + \sum_{j=1}^{t+1} R^{S_{j-1}}(Y^+_j))$.*

Proof. By summing the inequality stated in Lemma 9 over $j = 1, \ldots, t$, we obtain a telescopic sum that yields

$$R(S_t) \geq R(Z_1) - \left(R^{S_t}(Z_{t+1}) + \sum_{j=2}^{t+1} R^{S_{j-1}}(Y^+_j) \right).$$

Because every item in S^* must have a nonnegative price and $S^* = Z_1 \cup Y_1$ by definition, we have $R(S^*) \leq R(Z_1) + R(Y_1)$ by subadditivity of the revenue function (see Lemma 8). Combining these two inequalities concludes the proof. □

4.3. Analysis

In the following theorem, we show that the externality adjustment–based algorithm gives a $(1/2 - \epsilon)$-approximation for (Cardinality-Assort).

Theorem 3. *For any $\epsilon > 0$, Algorithm 3 computes a $(1/2 - \epsilon/2)$-approximation for (Cardinality-Assort). Moreover, the running time is polynomial in the input size and $1/\epsilon$.*

Proof. Given a fixed error parameter $\epsilon > 0$, let j^* be the unique integer for which $\frac{R(S^*)}{1+\epsilon} \leq B_{j^*} \leq R(S^*)$. Setting $B = B_{j^*}$, consider the solution returned by Algorithm 2 with guess B and threshold α. We consider two cases based on the condition by which the algorithm terminates.

1. If the algorithm stops after completing step k, then, by linearity of the revenue when using the externality-adjustment updates (Lemmas 4 and 5), the resulting solution S_k has a revenue of

$$\begin{aligned} R(S_k) &= \sum_{t=1}^{k} R^{S_{t-1}}(\{\sigma_t\}) \geq \alpha B \geq \frac{\alpha}{1+\epsilon} \cdot R(S^*) \\ &\geq (1-\epsilon)\alpha R(S^*), \end{aligned}$$

where the inequality holds because the item σ_t belongs to the consideration set C_t, and therefore, $R^{S_{t-1}} \cdot (\{\sigma_t\}) \geq \alpha B/k$.

2. Now, suppose the algorithm stops at the end of step $k' < k$ after discovering that $C_{k'+1} = \emptyset$. From Lemma 10, we get

$$R(S_{k'}) + R^{S_{k'}}(Z_{k'+1}) \geq R(S^*) - \sum_{j=1}^{k'+1} R^{S_{j-1}}(Y^+_j).$$

Now, because $C_{k'+1} = \emptyset$, this implies that $Z_{k'+1} = \emptyset$. Moreover, from Lemma 8, we also have $R^{S_{j-1}}(Y^+_j) < |Y^+_j| \cdot \alpha \cdot B/k$ for all $j = 1, \ldots, k'+1$. Therefore,

$$\sum_{j=1}^{k'+1} R^{S_{j-1}}(Y^+_j) \leq \alpha \cdot \frac{B}{k} \cdot \sum_{j=1}^{k'+1} |Y^+_j| \leq \alpha B \leq \alpha R(S^*),$$

where the second inequality holds because $\sum_{j=1}^{k'+1} |Y^+_j| \leq k$ and the last inequality holds as $B \leq R(S^*)$. Therefore,

$$R(S_{k'}) \geq R(S^*) - \alpha R(S^*) = (1-\alpha) \cdot R(S^*).$$

This shows that the approximation ratio attained by our algorithm is $\min\{(1-\epsilon)\alpha, 1-\alpha\}$. Picking $\alpha = 1/2$, we obtain a $(1/2 - \epsilon/2)$-approximation for (Cardinality-Assort). In terms of running time, Algorithm 3 considers $J = O(\frac{1}{\epsilon}\log(n/k))$ guesses for $R(S^*)$, and for any given guess, the running time of Algorithm 2 is polynomial in the input size. Therefore, the overall running time of Algorithm 3 is polynomial in the input size and $1/\epsilon$. □

4.3.1. Tight Example. Theorem 3 shows that running Algorithm 2 with an input guess $B = R(S^*)$ and threshold $\alpha = 1/2$ guarantees an approximation ratio of at least $1/2$. Here, we show that, for these input parameters, there exists an instance in which the approximation ratio of $1/2$ is tight. For this purpose, we consider an instance with three items. The Markov chain has four states $\mathcal{N}_+ = \{s, 1, 2, 0\}$. The prices are $p_s = 1$ and $p_1 = p_2 = 2$. The arrival rate for state s is $\lambda_s = 1$, and all other states have an arrival rate of zero. The transition probabilities are given in Figure 2. Consider the cardinality-constrained assortment problem with $k = 1$. The optimal assortment is $S^* = \{s\}$ with $R(S^*) = 1$. With guess $R(S^*)$ and $\alpha = 1/2$, the consideration set in the first step is $\{s, 1, 2\}$, and therefore, Algorithm 2 picks either one or two, obtaining a revenue of $R(S^*)/2$.

We would like to note that Algorithm 2 is employed as a subroutine in Algorithm 3 for each of the guesses $\{B_j\}_{j\in[J]}$ and returns the best solution across all guesses. Therefore, the performance bound of our algorithm is at least $1/2 - O(\epsilon)$ and possibly better. In fact, in our computational study (see Section 7), we observe that the empirical performance of this algorithm is significantly better than its theoretical worst-case bound of $1/2 - O(\epsilon)$. It is an interesting open question to provide a tighter analysis of the approximation bound for Algorithm 3 that returns the best solution among all guesses of $R(S^*)$.

5. Capacity-Constrained Assortment Optimization

In this section, we show that the capacity-constrained assortment problem under the Markov chain model can be approximated within factor $1/3 - \epsilon$ for any fixed $\epsilon > 0$. Recall that, unlike the simpler cardinality case, now each item i has an arbitrary weight w_i, and we have an upper bound of W on the total weight of items picked. We assume without loss of generality that each item individually satisfies the capacity constraint; that is, $w_i \le W$ for all $i \in \mathcal{N}$.

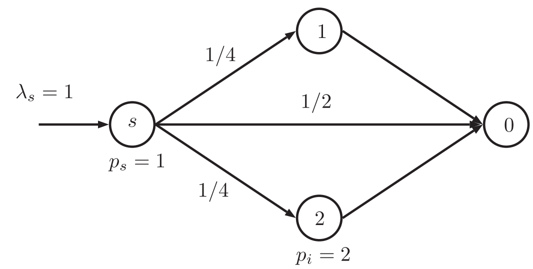
5.1. The Algorithm

5.1.1. Overview. In what follows, we describe an externality adjustment–based algorithm, similar in spirit to the one for the cardinality-constrained problem by suitably adapting the way consideration sets are defined. For this purpose, instead of considering items whose incremental absorption revenue exceeds a certain threshold, we only consider items whose incremental absorption revenue per unit of weight exceeds a certain threshold.

Our algorithm is similar in spirit to Algorithm 3 other than its selection step. In particular, we select in each step the highest nonpositive adjusted price item only among items that increase the revenue of the assortment picked so far by at least some predetermined threshold *multiplied by the weight of the item*. We then proceed by updating the prices and transition probabilities according to the externality-adjustment updates described in Figure 1. This selection and updating process is repeated until either the capacity constraint is violated or no further items can be picked.

5.1.2. The Algorithm. Again, our algorithm selects a single item in each step. Let S_t be the set of selected items by the end of step t, starting with $S_0 = \emptyset$. We use σ_t to denote the item picked in step t, meaning that $S_t = \{\sigma_1, \ldots, \sigma_t\}$. At every step $t \ge 1$, we select the highest adjusted price item (with respect to $p^{S_{t-1}}$, breaking ties arbitrarily) among items in the following consideration set:

Figure 2. A Tight Example for Algorithm 3



$$C_t = \left\{ i \in \mathcal{N}\setminus S_{t-1} : \frac{R^{S_{t-1}}(\{i\})}{w_i} \ge \alpha \frac{R(S^*)}{W} \right\},$$

where S^* is the optimal solution, W is the capacity bound, and $\alpha \in (0, 1)$ is a parameter whose value will be optimized later. Once the item σ_t is selected, we recompute the adjusted prices via the externality-adjustment update described in Figure 1. This selection and update process is repeated until either the consideration set becomes empty or adding the current item violates the capacity constraint; the step by which this condition is met is denoted by t'. In the former case, we stop and return $S_{t'-1}$. In the latter case, we take either $S_{t'-1}$ or $\{\sigma_{t'}\}$, depending on which of these sets has a larger total revenue.

5.1.3. Guessing $R(S^*)$. As in the case of cardinality constraints, because the value of $R(S^*)$ is unknown, we explain how to approximately guess it. We use a procedure similar to the one given in Section 4.1 with the exception of utilizing $R(U^*)/|U^*|$ as a lower bound (see proof of Lemma 2 in Appendix A.2), where U^* is the optimal unconstrained solution. In particular, we consider the following guesses for $R(S^*)$:

$$B_j = \frac{1}{|U^*|} \cdot R(U^*) \cdot (1+\epsilon)^j, \qquad j = 1, \ldots, J, \tag{4}$$

where $J = \min\{j \in \mathbb{N} : B_j \ge R(U^*)\} = O(\frac{\log n}{\epsilon})$. Algorithm 4 provides a formal description of our approximation algorithm for (Capacity-Assort), given a particular guess B_j for $R(S^*)$ and threshold α, and Algorithm 5 describes the complete procedure.

Algorithm 4 (Algorithm with Guess B_j and Threshold α)

1: Let S be the set of states picked so far, starting with $S = \emptyset$.
2: For all $S \subseteq \mathcal{N}$, let $C(S) = \{i \in \mathcal{N} : \frac{R^S(\{i\})}{w_i} \ge \alpha \cdot \frac{B_j}{W}\}$.
3: While $\sum_{i\in S} w_i < W$ and $C(S) \ne \emptyset$:
   a. Let i^* be the item of $C(S)$ for which p_i^S is maximized, breaking ties arbitrarily.
   b. If $\sum_{i\in S\cup\{i^*\}} w_i < W$, add i^* to S.
   c. Else return the highest revenue set among $\{i^*\}$ and S.

Return S.

Algorithm 5 (Algorithm for (Capacity-Assort) with Threshold α)

1: Given an error parameter $\epsilon > 0$, let J and $\{B_j\}_{j\in[J]}$ be defined according to (4).
2: For $j \in [J]$, let S_j be the solution returned by Algorithm 4 with guess B_j and threshold α.
3: Return $\arg\max_{j\in[J]} R(S_j)$.

5.2. Analysis

To analyze this algorithm, it is convenient to have a technical claim similar to Lemma 10. By defining the same sets Y_t and Z_t with respect to the optimal assortment S^* to (Capacity-Assort) and the adapted consideration sets C_t, precisely the same lemma statement holds. We, therefore, do not repeat this claim and its proof as these are identical to those of Lemma 10. The following theorem, whose proof is presented in Appendix A.5, shows that the externality adjustment–based algorithm gives a $(1/3 - \epsilon)$-approximation for (Capacity-Assort).

Theorem 4. *For any $\epsilon > 0$, Algorithm 5 computes a $(1/3 - \epsilon/3)$-approximation for (Capacity-Assort). Moreover, the running time is polynomial in the input size and $1/\epsilon$.*

5.2.1. Tight Example. Theorem 4 shows that executing Algorithm 4 with the true value of $R(S^*)$ and threshold $\alpha = 2/3$ as its inputs generates a 1/3-approximate solution, which is the highest achievable worst-case ratio for Algorithm 4 if we must fix a set of input parameters a priori. Here we show that, for this guess of input parameters, there exists an instance in which the ratio approximation of 1/3 is tight.

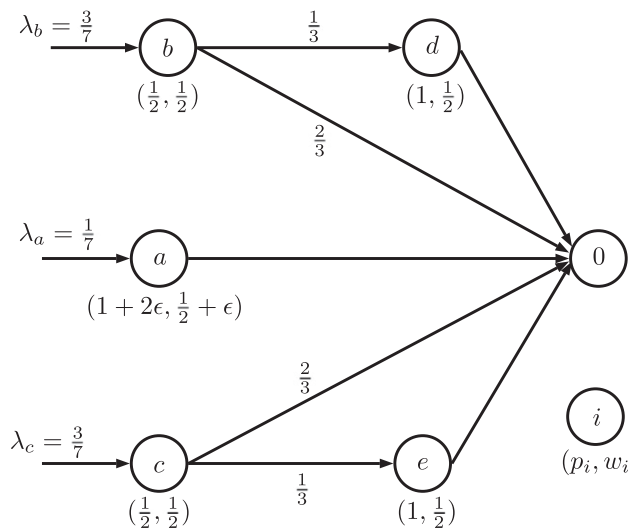
Consider the instance given in Figure 3. For a capacity bound of $W = 1$, the optimal assortment is $S^* = \{b, c\}$ with $R(S^*) = 3/7$. Initially, all items are in the consideration set, and the algorithm picks item a, the highest-price item. In the next step, no item can be added to the assortment. The algorithm, therefore, returns $S = \{a\}$ because $R(\{a\}) > R(\{d\})$ and yields a revenue of $(1 + 2\epsilon)/7 = R(S^*)/3 + O(\epsilon)$. When ϵ tends to zero, the approximation ratio tends to 1/3.

Similar to the cardinality-constrained setting, we would like to note that Algorithm 4 is employed as a subroutine of Algorithm 5 for each of the guesses $\{B_j\}_{j\in[J]}$ over which the best solution is returned. Therefore, the practical performance bound of our algorithm is possibly much better than $1/3 - O(\epsilon)$ as we observed in our computational study (see Section 7).

6. Hardness of Approximation

In this section, we present our hardness of approximation results for the constrained assortment-optimization problem under the Markov chain choice model.

Figure 3. A Tight Example for Algorithm 5



6.1. APX-Hardness for a Cardinality Constraint with Uniform Prices

We show that (Cardinality-Assort) is APX-hard; that is, this problem is NP-hard to approximate within a given constant, strictly smaller than one. In particular, we prove this result even when all items have uniform prices.

Theorem 5. (Cardinality-Assort) *is APX-hard even when all items have equal prices.*

Our proof, presented in Appendix A.6, is based on a gap-preserving reduction from the minimum vertex cover problem on three regular (or cubic) graphs. This problem is known to be APX-hard (Alimonti and Kann 2000). In other words, for some constant $\alpha > 0$, it is NP-hard to determine whether the minimum-cardinality vertex cover for a cubic graph is of size at most k or at least $(1 + \alpha)k$.

6.2. Totally Unimodular Constraints

We consider the assortment-optimization problem under the Markov chain model for the more general case of totally unimodular constraints. For an assortment $S \subseteq \mathcal{N}$, let $x^S \in \{0,1\}^{|\mathcal{N}|}$ denote its incidence vector, where $x_i^S = 1$ if $i \in S$ and $x_i^S = 0$ otherwise. The assortment-optimization problem subject to a totally unimodular constraint can be formulated as

$$\max_{S\subseteq\mathcal{N}} \{R(S) : Ax^S \le b\}, \tag{TU-Assort}$$

where A is a totally unimodular matrix and b is an integer vector. Note that (Cardinality-Assort) is a special case of (TU-Assort). In Appendix A.7, we show that (TU-Assort) for the Markov chain model is NP-hard to approximate within factor $O(n^{1/2-\epsilon})$ for any fixed $\epsilon > 0$. This result drastically contrasts that of Davis et al. (2013), who proved that the assortment-optimization problem with totally unimodular constraints can be solved in polynomial time when consumers choose according to the MNL model.

Theorem 6. *It is NP-hard to approximate* (TU-Assort) *in polynomial time within factor* $O(n^{1/2-\epsilon})$ *for any fixed* $\epsilon > 0$.

From a technical perspective, to establish our inapproximability results for (TU-Assort), we demonstrate that totally unimodular constraints in the Markov chain model capture the distribution-over-permutations model as a special case. Aouad et al. (2018) show that even unconstrained assortment optimization under a general distribution-over-permutations (or rankings) model is hard to approximate within factor $O(n^{1-\epsilon})$ for any fixed $\epsilon > 0$. In an instance of the latter model, we are given a collection of items $\mathcal{N} = \{1, \ldots, n\}$ with prices $p_1 \le \cdots \le p_n$, respectively. In addition, we are given an arbitrary (known) distribution on K preference lists, $L_1, \ldots, L_K$, each of which specifies a subset of the items listed in decreasing order of preference. A customer with a given preference list selects the most preferred item that is offered (possibly the no-purchase item) according to the customer's list. The goal is to find an assortment whose expected revenue is maximized. Further details are provided in Appendix A.7.

7. Computational Experiments

In this section, we present an extensive computational study to evaluate the performance of our algorithms for the cardinality- and capacity-constrained assortment-optimization problems under the Markov chain choice model. In particular, we focus in Section 7.1 on testing (i) the practical performance of our algorithm with respect to its theoretical worst-case guarantee and (ii) the running time of this algorithm. In Section 7.2, we conduct experiments in settings in which the instance parameters can be correlated to better understand which factors can potentially cause the practical performance of our algorithms to deteriorate. In Section 7.3, we present numerical experiments to showcase the benefits of using a Markov chain model over a simpler MNL model. In each of these settings, to evaluate the quality of the solution returned by our algorithms, we propose a MIP formulation for (Cardinality-Assort), whose specifics are provided in Online Appendix A.

7.1. Practical Performance and Running Time

7.1.1. Settings Tested. We begin by describing the families of random instances being tested in our computational experiments. Here, each item's price p_i is uniformly distributed over the interval $[0, 1]$. Note that because we present statistics regarding approximation factors, any constant here will give identical results, so the choice of one is arbitrary. For each instance, we compute the optimal unconstrained assortment U^* using the LP given by Blanchet et al. (2016). We then choose the cardinality constraint k uniformly between one and $|U^*|/2$. For the transition probabilities ρ_{ij} and the arrival rates λ_i, we test our algorithm on three different settings:

1. We generate n^2 independent random variables X_{ij}, each picked uniformly over the interval $[0, 1]$. We then set $\rho_{ij} = X_{ij}/\sum_{k=0}^{n} X_{ik}$ for all i, j such that $i \ne j$. Because we do not allow self-loops (i.e. $\rho_{ii} = 0$), the number of random variables needed is n^2. For the arrival rates, we then generate n independent random variables Y_i, each picked uniformly over the interval $[0, 1]$, and set $\lambda_i = Y_i/\sum_{j=1}^{n} Y_j$ for all $i \ne 0$.
2. In this setting, we sparsify the transition matrix of setting 1. More precisely, we additionally generate n^2 independent random variables Z_{ij}, each following a Bernoulli distribution with parameter 0.2. For all i, j such that $i \ne j$, we set $\rho_{ij} = Z_{ij}X_{ij}/\sum_{k=0}^{n} Z_{ik}X_{ik}$, where X_{ij} are generated as in setting 1. This is equivalent to eliminating each transition (i, j) with probability 0.8 and then renormalizing. The arrival rates are generated similarly to setting 1.
3. The transition matrix here is one of a random walk. More precisely, we generate n^2 independent random variables X_{ij}, each following a Bernoulli distribution with parameter 0.5. We then set $\rho_{ij} = X_{ij}/\sum_{j=0}^{n} X_{ij}$ for all i, j such that $i \ne j$. We also generate n random variables Y_i, each following a Bernoulli distribution with parameter 0.5 and set $\lambda_i = Y_i/\sum_{j=1}^{n} Y_j$ for all $i \ne 0$.

7.1.2. Results and Discussion. We examine how our algorithm performs in terms of both approximation and running time. Table 1 shows the approximation ratio of Algorithm 3 (with $\epsilon = 0.1$) for the different settings and the different values of n. We use the MIP formulation proposed in Online Appendix A to compute the optimal assortment. As can be observed, the actual performance of our algorithm is significantly better than its theoretical worst-case guarantee. Indeed, in all settings tested, the average approximation ratio is always above 0.97. Moreover, the worst approximation ratio over all instances is above 0.77.

The running time of our algorithm also scales nicely. Table 2 shows the performance of Algorithm 3 in terms of running time for setting 2. For the other settings, the running times are very similar and are, therefore, omitted. On the other hand, although the MIP running time can be competitive in some cases, it blows up when the number of items n increases (see Table 2). Note that 12 out of the 100 instances tested for $n = 100$ had a MIP running time of at least 30 minutes. For $n = 200$, we set a time limit of two hours for the MIP. Out of the 20 random instances generated, 16 reached the time limit without terminating. These numerical experiments suggest that Algorithm 3 is computationally

Table 1. Performance of Algorithm 3 for (Cardinality-Assort)

		Approximation ratio		Number of instances within x% of OPT				
Setting	n	Average	Minimum	2%	5%	10%	20%	Number of instances
1	30	0.9783	0.7771	664	812	972	998	1,000
2	30	0.9784	0.7734	662	858	956	995	1,000
3	30	0.9830	0.7693	708	884	976	998	1,000
1	60	0.9803	0.8671	622	838	997	1,000	1,000
2	60	0.9796	0.8094	621	888	982	1,000	1,000
3	60	0.9854	0.8885	693	941	998	1,000	1,000
1	100	0.9763	0.9132	52	79	100	100	100
2	100	0.9782	0.8882	59	91	99	100	100
3	100	0.9848	0.9142	70	97	100	100	100

efficient and that its numerical performance is significantly better than the theoretical worst-case guarantee. We would like to note that the mixed integer programming formulation (A.3) for the constrained assortment-optimization problem has a large integrality gap, and therefore, the corresponding LP relaxation is very weak, which explains the large running times as we increase the number of items. It is an interesting open question to develop a formulation with a small integrality gap for our problem.

We also test Algorithm 5 for (Capacity-Assort). To this end, we draw each item's weight w_i uniformly over the interval [0,1] and choose the capacity constraint W uniformly over $[2 \cdot \min_i w_i, w(U^*)]$, where $w(U^*)$ denotes the weight of the optimal unconstrained assortment. In Table 3, we report the empirical performance guarantee of Algorithm 5 for the three settings tested. As in the cardinality-constrained case, the numerical performance is significantly better than the theoretical worst-case guarantee. In particular, the ratio between the expected revenue computed by our algorithm and the optimal revenue is roughly 0.98, on average, and at least 0.74 over all instances tested; both are significantly better than the worst-case bound of $1/3 - \epsilon$. We also report the running time of our algorithm in Table 4. Again, our numerical experiments suggest that Algorithm 5 is also computationally efficient and significantly outperforms the theoretical worst-case guarantees.

These results are encouraging as they show that the average performance of our algorithms far exceeds their theoretical worst-case guarantee. In the next section, inspired by the bad example described in Figure 2, we explore instances whose parameters are correlated. In particular, we show that when the correlation is high and the size of the constraint is moderate, the performance of our algorithm slightly degrades.

7.2. Performance on Correlated Instances

The previous section highlights the very appealing performance of our algorithms when the problem parameters are generated independently. Moreover, their running times nicely scale with the number of items n. In this section, we further investigate which factors can potentially cause our algorithms to perform worse. For this purpose, we fix the number of items at $n = 30$ and consider a family of cardinality-constrained instances whose parameters are correlated.

7.2.1. Settings Tested. We generate the parameters of our random instances as follows.

- Prices: We generate n independent random variables Y_i, each picked uniformly over the interval $[0, 1]$. We denote by $Y_{(i)}$ the ith smallest realized value and set $p_i = Y_{(i)}$ for $i \in [n]$.
- Transition probabilities: We generate independent random variables X_{ij} for $i \neq j$, each picked uniformly over the interval $[0, 1]$. Further, we generate independent random variables Z_{ij} for $i \neq j$ and $j \neq 0$, each following a Bernoulli distribution with parameter μ if $i > j$ and with $1 - \mu$ if $i < j$. We then set $\rho_{ij} = Z_{ij}X_{ij}/\sum_{k=0}^{n} X_{ik}Z_{ik}$ for all i, j such that $i \neq j$.

Table 2. Running Time of Algorithm 3 and the MIP for Setting 2

	Average running time		Maximum running time		
n	Algorithm 3	MIP	Algorithm 3	MIP	Number of instances
30	0.18	0.17	0.67	0.25	1,000
60	0.74	0.67	1.25	29.34	1,000
100	3.18	278.20	9.16	10,226.98	100
200	31.98	**	47.38	**	20

Note. We use "**" to denote cases hitting the time limit of two hours for $n = 200$.

Management Science, 2020, vol. 66, no. 2, pp. 698–721, © 2019 INFORMS

Table 3. Performance of Algorithm 5 for (Capacity-Assort)

		Approximation ratio		Number of instances within x% of OPT				
Setting	n	Average	Minimum	2%	5%	10%	20%	Number of instances
1	30	0.9854	0.7447	814	926	965	993	1,000
2	30	0.9833	0.7828	776	890	963	995	1,000
3	30	0.9850	0.7668	801	902	969	999	1,000
1	60	0.9870	0.7587	836	930	973	996	1,000
2	60	0.9844	0.8091	772	900	981	1,000	1,000
3	60	0.9847	0.7321	775	913	974	998	1,000
1	100	0.9917	0.8868	88	97	98	100	100
2	100	0.9852	0.8966	77	92	99	100	100
3	100	0.9851	0.8892	74	91	99	100	100

Table 4. Running Time of Algorithm 5 and the MIP for Setting 2

	Average running time		Maximum running time		
n	Algorithm 5	MIP	Algorithm 5	MIP	Number of instances
30	0.36	1.23	0.03	0.09	1,000
60	2.69	6.86	0.18	1.99	1,000
100	9.33	17.97	2.54	101.14	100
200	46.59	69.21	463.72	2,275.52	20

- Arrival rates: We generate n independent random variables W_i, each picked uniformly over the interval $[0,1]$, and set $\lambda_i = W_i/\sum_{j=1}^{n} W_j$ for all $i \neq 0$.

It is worth noting that the parameter μ captures certain correlation between the generated parameters. Indeed, because the prices are ordered, the only allowed transitions are toward cheaper (respectively, more expensive) items when $\mu = 1$ (respectively, $\mu = 0$). For each instance, we compute the optimal unconstrained assortment U^* and choose a cardinality constraint k uniformly between one and $|U^*|$. Letting $\rho = k/|U^*|$, this parameter represents the relative size of the constraint with respect to the size of the optimal unconstrained assortment. Intuitively, the problem is expected to be harder for moderate values of ρ. Indeed, when $\rho = 1$, the optimal unconstrained assortment is also optimal for the constrained problem; when ρ is very small, the number of feasible assortments is small as well. For each value of $\mu \in \{0, 0.1, \ldots, 1\}$, we generate 10,000 random instances and report the average and minimum approximation ratio of Algorithm 3 (with $\epsilon = 0.5$) as a function of ρ.

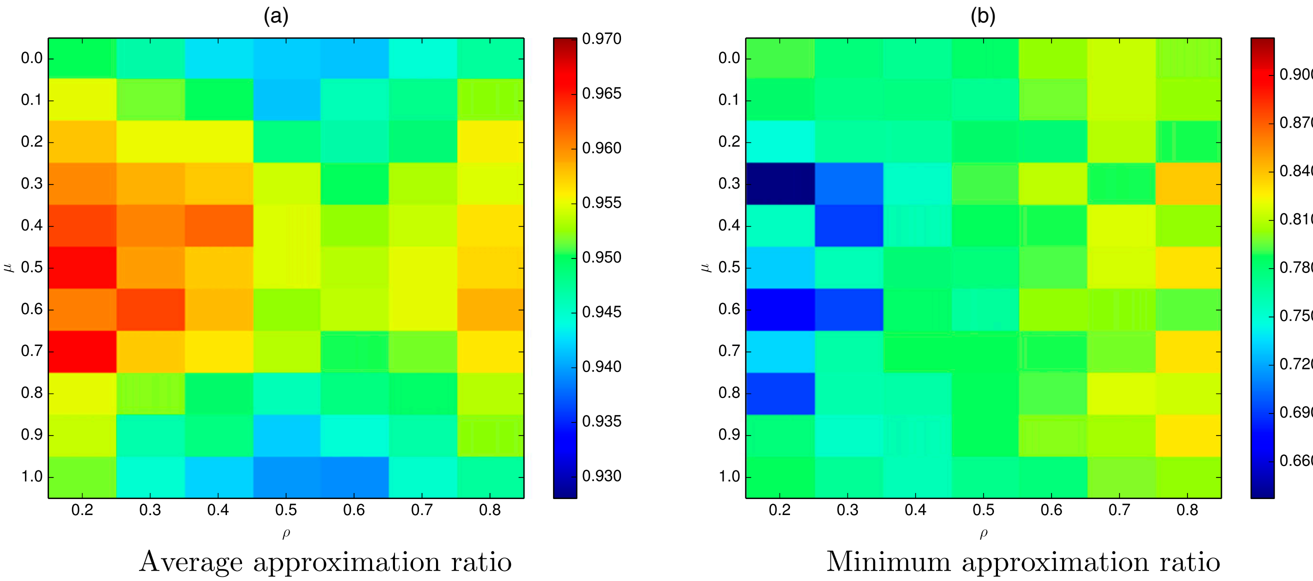
7.2.2. Results and Discussion. Figure 4 reports the results of these experiments via a heat map. The constraint size parameter ρ varies along the x-axis and the correlation parameter μ along the y-axis. As hinted by the worst-case example in Figure 2, the average performance of our algorithm indeed depends on the correlation present in the instance. We observe that the average performance degrades when μ is close to zero or one. Additionally, one can notice that the performance worsens for moderate values to ρ. In particular, the worst parameter combinations for our algorithm occurs at $(\rho, \mu) \in \{(0.5, 1), (0.5, 0)\}$, that is, precisely when there is a strong correlation and when the relative size of the constraint is moderate. In this regime, the average approximation ratio can decrease down to 93.5% from up to 97% for other combinations of the parameters.

Perhaps more surprisingly, the minimum approximation ratio occurs in a different regime. As shown in Figure 4(b), we incur the worst minimum approximation ratio when ρ is very small, that is, when the constraint is relatively small. In those cases, because our algorithm can only pick a small number of items, a bad choice can potentially lead to significant losses in revenue. That said, a simple search over the feasible assortments would be sufficient to identify the optimal assortment. These results highlight some of the factors that can drive down the performance of our algorithm.

7.3. Comparison Between the Markov Chain and MNL Models

In this section, we aim to better understand the value of utilizing the Markov chain model in comparison with a simpler option, such as an MNL model. Although, Blanchet et al. (2016) show that the Markov chain model generalizes MNL, this question is particularly interesting in our setting given that the constrained assortment problem can be solved exactly under an MNL model but only approximately under a Markov chain model. It is,

Figure 4. (Color online) Average and Minimum Approximation Ratio of Algorithm 5 as a Function of the Relative Size of the Constraint ρ (x-Axis) and the Correlation Parameter μ (y-Axis)



therefore, unclear a priori whether using the Markov chain model, the more complex option, is advantageous in this setting.

To conduct our experiments, we use a mixture-of-MNL model [which can approximate any random utility model (McFadden and Train 2000)] as a ground truth model and evaluate the quality of the assortments returned when employing an MNL model or a Markov chain model as an approximation. It is worth emphasizing that we focus here on the performance of the assortment optimization algorithm and not on the prediction performance or on the estimation procedure itself. Both of these questions have been considered in previous literature (Blanchet et al. 2016, Feldman and Topaloglu 2017a, Simsek and Topaloglu 2017).

7.3.1. Settings Tested. In a mixture-of-MNL model, each item i (including the no-purchase option) has K utility parameters, $u_{i,k}$, one for each class $k = 1, \ldots, K$. We are also given probabilities θ_k for each class k that represents the fraction of customers belonging to that particular class. For any given assortment S, each item $i \in S$ is picked with probability

$$\pi^{\text{truth}}(i, S) = \sum_{k=1}^{K} \theta_k \cdot \frac{u_{i,k}}{u_{0,k} + \sum_{j \in S} u_{j,k}}.$$

To generate the mixture-of-MNL parameters, similar to Feldman and Topaloglu (2017a), we assume that there is an intrinsic correlation between the quality of an item and its price. In particular, we generate n independent random numbers V_i uniformly in $[0,1]$ and let $p_i = V_{(i)}$ for $i \in [n]$, where $V_{(i)}$ is the ith smallest generated number. For the utilities, we generate n independent random number U_i uniformly in $[0,1]$. We also select, for each class k, two integers $1 \le \ell_k \le L_k \le K$ and set

$$u_{i,k} = \begin{cases} U_{(i)}, & \text{if } 1 \le \ell_k \le i \le L_k \\ 1, & \text{if } i = 0 \\ 0, & \text{otherwise,} \end{cases}$$

where $U_{(i)}$ denotes the ith smallest generated number. For a given k, we generate the pair (ℓ_k, L_k) by first sampling ℓ_k uniformly in $[1, n]$ and then L_k uniformly in $[\ell_k, n]$. Note that we normalize the no-purchase utility to one in all classes. To make sure that each item has a nonzero probability of being purchased, we assume that $\ell_1 = 1$ and $L_1 = n$. This choice models a situation in which each class of customers has a maximal threshold on prices and a minimal threshold on quality and resembles the consideration sets considered by Aouad et al. (2015). The probabilities θ_k are generated uniformly at random in $[0,1]$ and are normalized to have a sum of one. To generate the cardinality constraint k, we compute the optimal unconstrained assortment U^* under the ground-truth model and set k to be a random integer uniformly in $[|U^*|/2]$. We mention in passing that even the unconstrained assortment optimization problem is NP-hard under a mixture of MNL model (Rusmevichientong et al. 2014). However, the latter problem admits an MIP formulation (see, for instance, Méndez-Díaz et al. 2014).

For every generated ground-truth instance, the revenue of any assortment under the true underlying model can be computed as

$$R^{\text{truth}}(S) = \sum_{i \in S} p_i \cdot \pi^{\text{truth}}(i, S).$$

We compare $R^{\text{truth}}(S^{MNL})$ and $R^{\text{truth}}(S^{MC})$, where S^{MNL} (respectively, S^{MC}) is the assortment obtained using an MNL (respectively, Markov chain) approximation. More precisely, to obtain S^{MNL}, we solve (Cardinality-Assort) assuming the underlying model is an MNL model, in which each utility parameter is set to its expected value; that is, for each item i, we let $\hat{u}_i = \sum_k \theta_k \cdot u_{i,k}$. Note that the optimal constrained assortment S^{MNL} can be computed via linear programming (Davis et al. 2013). To obtain S^{MC}, we solve (Cardinality-Assort) using Algorithm 3 (with $\epsilon = 0.5$), assuming the underlying model is a Markov chain model, whose parameters are obtained using the procedure proposed by Blanchet et al. (2016). Even though the Markov chain model generalizes MNL, it is unclear a priori whether we would indeed get $R^{\text{truth}}(S^{MC}) \geq R^{\text{truth}}(S^{MNL})$ because of solving the Markov chain model in an approximate way.

7.3.2. Results and Discussion. Table 5 reports the average, minimum, and maximum performance of S^{MC} over S^{MNL} over 1,000 randomly generated instances. More precisely, for each set of parameters (n, K), we compute the average, minimum, and maximum value of $R^{\text{truth}}(S^{MC})/R^{\text{truth}}(S^{MNL})$. We also report in the last column the percentage of instances in which $R^{\text{truth}}(S^{MC}) \geq R^{\text{truth}}(S^{MNL})$.

We observe that, for a wide range of parameters, the Markov chain model outperforms the MNL model. This dominance is exhibited despite the fact that S^{MC} is computed using Algorithm 3, meaning that it is an approximate solution to (Cardinality-Assort) under the Markov chain model. On average, using a Markov chain model increases the revenue by more than 12% in all settings tested and up to 50% on average for $n = 60$ and $K = 10$. The increase can be as substantial as 1,200% in the best case; that is, there is at least one instance for which the gap is as wide. Looking more deeply at the results, the Markov chain model outperforms the MNL model in 60%–80% of the instances for all sets of parameters but one. Finally, the performance gap between the two models widens as the number of classes K in the ground truth increases. This confirms our intuition that MNL is a fair approximation for mixture-of-MNL when the number of mixtures is small. However, as K increases, MNL can no longer approximate the heterogeneity of the utility parameters as well as the Markov Chain model can.

Table 5. Performance of the Markov Chain Model Against the MNL Model for (Cardinality-Assort)

		$R^{\text{truth}}(S^{MC})/R^{\text{truth}}(S^{MNL})$			
n	K	Average	Minimum	Maximum	MC ≥ MNL
30	5	1.121	0.658	5.494	79.9%
30	10	1.160	0.615	4.314	81.1%
60	5	1.126	0.816	10.538	60.3%
60	10	1.529	0.799	12.096	83.5%
100	5	1.163	0.782	22.756	47.9%
100	10	1.728	0.790	17.097	62.5%
100	15	1.809	0.734	14.212	83.8%

Table 6. Running Time Comparison Between Algorithm 3 and the MIP for Mixture of MNL

	$t_{\text{MIP}}/t_{\text{ALG}}$	
K	Average	Maximum
5	0.33	6.08
10	0.89	21.24
15	8.39	3,629.47
20	35.20	696.75

From a computational efficiency perspective, as n and K increase, the running time of our algorithm outperforms the one needed to solve the MIP under the mixture-of-MNL model. In Table 6, we present the average and maximum $t_{\text{MIP}}/t_{\text{ALG}}$ ratios, where t_{MIP} and t_{ALG} are the running times (in seconds) incurred by the MIP and Algorithm 3, respectively. For $k \in \{5, 10, 15\}$, this performance measure is taken over 1,000 instances, and for $k = 20$ over only 50 instances (because of significantly high running times), with $n = 100$ items in all cases. As can easily be observed, the time needed to solve the latter MIP (ground-truth model) can be as much as eight times slower, on average, for $K = 15$ and as much as several thousand times slower in the worst case for $K = 15$. These results suggest that the Markov chain model not only outperforms a simpler model, such as MNL, in terms of expected revenue but also provides a more tractable alternative (in terms of running time) over a more complex model, such as a mixture of MNL.

8. Conclusions

In this paper, we consider the cardinality- and capacity-constrained assortment-optimization problems under the Markov chain model. We prove that this problem is APX-hard even when all item prices are uniform. We present a $(1/2 - \epsilon)$-approximation for the cardinality-constrained assortment-optimization problem and a $(1/3 - \epsilon)$-approximation for the capacity-constrained version. Our algorithmic approach is based on a new externality-adjustment paradigm that allows us to exactly capture the externality of adding an item to any given assortment on the remaining set of items. This approach enables us to linearize the revenue function, which is generally nonlinear, nonmonotone, and nonsubmodular. Our overall framework also provides new insights toward the optimal stopping problem as well as for assortment optimization in additional models such as MNL.

Acknowledgments

This work was done before C. Ye joined Amazon.

Appendix A. Additional Proofs

A.1. Proof of Lemma 1

Let us consider the following example with three items; that is, $\mathcal{N} = \{1, 2, 3\}$.

- Prices: $p_1 = 1$, $p_2 = 1/2$, $p_3 = 1$.
- Arrival probabilities: $\lambda_1 = 1$, $\lambda_2 = \lambda_3 = 0$.
- Transition probabilities: $\rho_{1,2} = 1$, $\rho_{2,3} = 1$, $\rho_{3,0} = 1$, and all other unspecified transition probabilities are zero.

Now consider $S = \{3\}$ and $T = \{2, 3\}$. Then, $R(S \cup \{2\}) = 1/2 < 1 = R(S)$, meaning that the revenue function is not monotone. Furthermore, $R(T \cup \{1\}) - R(T) = 1/2$ and $R(S \cup \{1\}) - R(S) = 0$. Therefore, this example illustrates that the revenue function is not submodular as well.

A.2. Proof of Lemma 2

We prove the desired result in two steps. We first show that the incremental greedy algorithm guarantees a $1/k$-approximation for (Cardinality-Assort) We then exhibit an instance in which incremental greedy obtains only an $1/k$-fraction of the optimal revenue, thus completing the proof.

Step 1. *Approximation guarantee*. The fact that the incremental greedy algorithm guarantees a $1/k$-approximation for (Cardinality-Assort) is an immediate corollary of the following (more general) claim. Let S^g be the solution returned by the incremental greedy algorithm, and let S be any subset of states. Then, $R(S^g) \geq R(S)/|S|$. To prove this claim, recall that the incremental greedy algorithm iteratively builds the assortment, and in each iteration, it adds the item that increases the expected revenue by the most. Let j be the first item selected by the algorithm, which necessarily exists as long as there is an item i with $p_i > 0$. Then, by definition of the greedy algorithm, we have $R(\{j\}) \geq R(\{i\})$ for every item $i \in S$. Therefore,

$$R(S^g) \geq R(\{j\}) \geq \frac{1}{|S|} \cdot \sum_{i \in S} R(\{i\}) \geq \frac{R(S)}{|S|},$$

where the last inequality follows from the subadditivity of the revenue function (see Lemma 8).

Step 2. *Bad example*. Consider the following instance of (Cardinality-Assort) with $n = k + 1$ items, where k is the upper bound specified by the cardinality constraint. We have a state s and states $i = 0, \ldots, k$. The arrival rates are all equal to zero, except for $\lambda_s = 1$. Moreover,

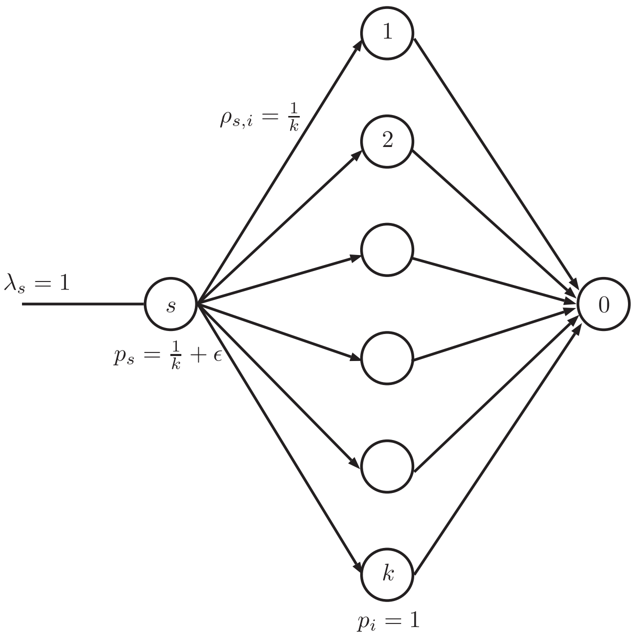
$$p_i = \begin{cases} (1/k) + \epsilon, & \text{if } i = s \\ 1, & \text{if } i = 1, \ldots, k, \end{cases}$$

$$\rho_{ij} = \begin{cases} 1/k, & \text{if } i = s \text{ and } j = 1, \ldots, k \\ 1, & \text{if } i = 1, \ldots, k \text{ and } j = 0 \\ 0, & \text{otherwise,} \end{cases}$$

where $\epsilon \leq 1/(2k)$. Figure A.1 provides a graphical representation of this instance.

The incremental greedy algorithm first picks item s as $R(\{s\}) = (1/k) + \epsilon$, and $R(\{i\}) = 1/k$ for $i = 1, \ldots k$. Once s is selected, adding any other state cannot increase the revenue. Therefore, the greedy algorithm gives a revenue of $(1/k) + \epsilon$. However, the optimal solution is to offer item 1 to k with a revenue of one in total. When ϵ tends to zero, the approximation ratio goes to $1/k$.

Figure A.1. A Bad Example for the Incremental Greedy Algorithm



A.3. Proof of Lemma 3

Step 1. *Approximation guarantee*. Let S^{mg} be the set of states selected by the modified greedy algorithm. Note that, for every $i \in S^{mg}$, we have that $\mathbb{P}(i \prec S_+^{mg} \backslash \{i\}) \geq \mathbb{P}(i \prec U_+^* \backslash \{i\})$ because S^{mg} is a subset of U^*. Thus,

$$\begin{aligned} R(S^{mg}) &= \sum_{i \in S^{mg}} \mathbb{P}(i \prec S_+^{mg} \backslash \{i\}) p_i \\ &\geq \sum_{i \in S^{mg}} \mathbb{P}(i \prec U_+^* \backslash \{i\}) p_i \\ &\geq \frac{k}{|U^*|} \cdot \sum_{i \in U^*} \mathbb{P}(i \prec U_+^* \backslash \{i\}) p_i \\ &= \frac{k}{|U^*|} \cdot R(U^*) \\ &\geq \frac{k}{n} \cdot R(S^*), \end{aligned}$$

where S^* is the optimal solution to (Cardinality-Assort). Here, the second inequality holds because of picking the top k states in terms of $\mathbb{P}(i \prec U_+^* \backslash \{i\}) \cdot p_i$ values. The last inequality holds because the optimal unconstrained revenue provides an upper bound on the optimal revenue in the constrained case.

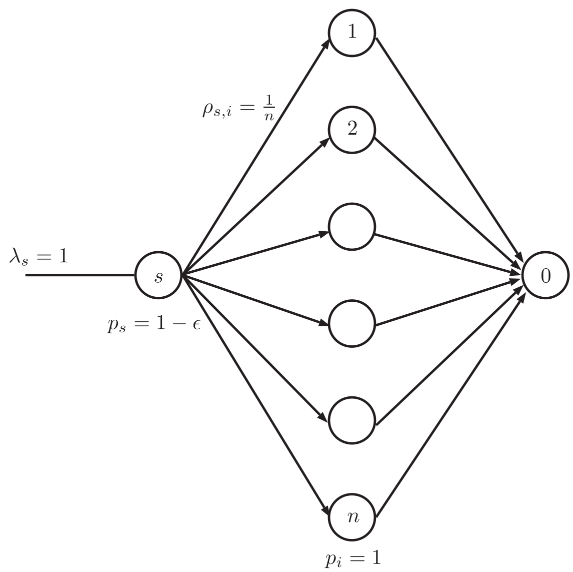
Step 2. *Bad example.* We present an example in which the revenue of every subset of k items within the optimal solution U^* is a factor k/n away from optimal. Consider the following instance of the problem with $n + 2$ items (or states). We have a state s, states $i = 1, \ldots, n$, and state 0 corresponding to the no-purchase option. The arrival rates are all equal to zero except for $\lambda_s = 1$. Moreover,

$$p_i = \begin{cases} 1 - \epsilon, & \text{if } i = s \\ 1, & \text{if } i = 1, \ldots, n, \end{cases}$$

$$\rho_{ij} = \begin{cases} 1/n, & \text{if } i = s \text{ and } j = 1, \ldots, n \\ 1, & \text{if } i = 1, \ldots, n \text{ and } j = 0 \\ 0, & \text{otherwise.} \end{cases}$$

Figure A.2 provides a graphical representation of this instance.

Figure A.2. A Bad Example for the Modified Greedy Algorithm



For this example, the unconstrained optimal assortment is $U^* = \{1, \ldots, n\}$, and the modified greedy algorithm on U^* selects k items among U^*, meaning that a total revenue of k/n is obtained. However, the optimal solution of the constrained problem is to only offer item s, which gives a revenue of $1 - \epsilon$. As ϵ tends to zero, the approximation ratio goes to k/n.

A.4. Proof of Lemma 5

To verify that $(p_i^{S_1})^{S_2} = p_i^{S_1 \cup S_2}$ for all S_1, S_2 and $i \notin S_1 \cup S_2$, note that

$$\begin{aligned}(p_i^{S_1})^{S_2} &= p_i^{S_1} - \sum_{j \in S_2} \mathbb{P}_i^{S_1}(j \prec S_{2+} \backslash \{j\}) p_j^{S_1} \\ &= p_i - \underbrace{\sum_{l \in S_1} \mathbb{P}_i(l \prec S_{1+} \backslash \{l\}) p_l}_{A} - \underbrace{\sum_{j \in S_2} \mathbb{P}_i^{S_1}(j \prec S_{2+} \backslash \{j\}) p_j^{S_1}}_{B}.\end{aligned}$$

Using the definition of the updated prices,

$$\begin{aligned}B &= \sum_{j \in S_2} \mathbb{P}_i^{S_1}(j \prec S_{2+} \backslash \{j\}) p_j \\ &\quad - \sum_{j \in S_2} \mathbb{P}_i^{S_1}(j \prec S_{2+} \backslash \{j\}) \sum_{l \in S_1} \mathbb{P}_j(l \prec S_{1+} \backslash \{l\}) p_l \\ &= \sum_{j \in S_2} \mathbb{P}_i(j \prec (S_2 \cup S_1)_+ \backslash \{j\}) p_j \\ &\quad \underbrace{- \sum_{j \in S_2} \mathbb{P}_i^{S_1}(j \prec S_{2+} \backslash \{j\}) \sum_{l \in S_1} \mathbb{P}_j(l \prec S_{1+} \backslash \{l\}) p_l}_{C}.\end{aligned}$$

We can now combine A and C,

$$\begin{aligned}A - C &= \sum_{l \in S_1} \Bigg(\mathbb{P}_i(l \prec S_{1+} \backslash \{l\}) \\ &\quad - \sum_{j \in S_2} \mathbb{P}_i(j \prec (S_2 \cup S_1)_+ \backslash \{j\}) \mathbb{P}_j(l \prec S_{1+} \backslash \{l\}) \Bigg) p_l \\ &= \sum_{l \in S_1} (\mathbb{P}_i(l \prec S_{1+} \backslash \{l\}) - \mathbb{P}_i(S_2 \prec l \prec S_{1+} \backslash \{l\})) p_l \\ &= \sum_{l \in S_1} \mathbb{P}_i(l \prec (S_2 \cup S_1)_+ \backslash \{j\}) p_l.\end{aligned}$$

Putting everything together, we get

$$(p_i^{S_1})^{S_2} = p_i - \sum_{j \in (S_2 \cup S_1)} \mathbb{P}_i(j \prec (S_2 \cup S_1)_+ \backslash \{j\}) p_j = p_i^{S_1 \cup S_2}.$$

A.5. Proof of Theorem 4

Given an error parameter $\epsilon > 0$, let j^* be the unique integer for which $\frac{R(S^*)}{1+\epsilon} \le B_{j^*} \le R(S^*)$. Letting $B = B_{j^*}$, consider the solution returned by Algorithm 4 with guess B and threshold α. We consider two cases based on the condition by which the algorithm terminates. Let t' be the step at which the algorithm terminates.

1. Suppose we stop the algorithm because adding the item $\sigma_{t'}$ violates the capacity constraint; that is, $\sum_{t=1}^{t'} w_{\sigma_t} > W$. In this case, we return either $S_{t'-1}$ or $\{\sigma_{t'}\}$, depending on which of these sets has a larger revenue. We argue that this choice guarantees a revenue of at least $\alpha R(S^*)/2$ because

$$\begin{aligned}\max\{R(S_{t'-1}), R(\{\sigma_{t'}\})\} &\ge \max\left\{\sum_{t=1}^{t'-1} R^{S_t}(\{\sigma_t\}), R^{S_{t'-1}}(\{\sigma_{t'}\})\right\} \\ &\ge \max\left\{\alpha \frac{B}{W} \sum_{t=1}^{t'-1} w_{\sigma_t}, \alpha \frac{B}{W} w_{\sigma_{t'}}\right\} \\ &= \alpha \frac{B}{W} \cdot \max\left\{\sum_{t=1}^{t'-1} w_{\sigma_t}, w_{\sigma_{t'}}\right\} \\ &\ge \alpha \frac{B}{2} \\ &\ge \alpha \cdot \frac{R(S^*)}{2(1+\epsilon)} \\ &\ge (1-\epsilon)\alpha \cdot \frac{R(S^*)}{2},\end{aligned}$$

where the third inequality holds because $\max\{\sum_{t=1}^{t'-1} w_{\sigma_t}, w_{\sigma_{t'}}\} \ge W/2$ and the fourth inequality follows as $B \ge R(S^*)/(1+\epsilon)$.

2. Suppose the algorithm terminates because $C_{t'+1} = \emptyset$. Using Lemma 10 adapted to the capacitated case, we have

$$R(S_{t'}) + R^{S_{t'}}(Z_{t'+1}) \ge R(S^*) - \sum_{j=1}^{t'+1} R^{S_{j-1}}(Y_j^+).$$

Because $C_{t'+1} = \emptyset$, this implies that $Z_{t'+1} = \emptyset$. Moreover, from Lemma 8, for all $j = 1, \ldots, t'+1$, we have

$$R^{S_{j-1}}(Y_j^+) < \alpha B \cdot \frac{\sum_{i \in Y_j^+} w_i}{W}.$$

Because our algorithm stopped prior to reaching the capacity constraint, we have $\sum_{j=1}^{t'+1} \sum_{i \in Y_j^+} w_i \le W$. Consequently, $\sum_{j=1}^{t'+1} R^{S_{j-1}}(Y_j^+) < \alpha B \le \alpha R(S^*)$, and therefore,

$$R(S_{t'}) \ge R(S^*) - \alpha R(S^*) = (1-\alpha) R(S^*).$$

As a result, the approximation ratio attained by our algorithm is $\min\{(1-\epsilon)\frac{\alpha}{2}, 1-\alpha\}$. By setting $\alpha = 2/3$, we obtain an approximation factor of $1/3 - \epsilon/3$. From a running time perspective, Algorithm 5 considers $J = O(\frac{1}{\epsilon} \log n)$ guesses of $R(S^*)$. Each run of Algorithm 4 for a given guess terminates in polynomial time. Therefore, the overall running time of Algorithm 5 is polynomial in the input size and $1/\epsilon$.

A.6. Proof of Theorem 5

Our proof is based on a gap-preserving reduction from the minimum vertex cover problem on three regular (or cubic) graphs to which we refer as VCC. This problem is known to be APX-hard (Alimonti and Kann 2000). In other words, for some constant $\alpha > 0$, it is NP-hard to determine whether the minimum-cardinality vertex cover for a cubic graph is of size at most k or at least $(1+\alpha)k$.

Consider an instance $\mathcal{I}$ of VCC, consisting of a cubic graph $G = (V, E)$ on n vertices $V = \{v_1, \dots, v_n\}$. We can assume that $k > |E|/3$, or otherwise, the distinction between the two cases is easy. We construct an instance $\mathcal{M}(\mathcal{I})$ of (Cardinality-Assort) as follows. Each vertex $v_i \in V$ corresponds to an item i of $\mathcal{N}$. In addition, we also have the no-purchase item 0. For each vertex $v \in V$, let $N(v)$ denote the neighborhood of v in G; that is, $N(v) = \{u : (u, v) \in E\}$, consisting of exactly three vertices. Now, for all $(i, j) \in \mathcal{N} \times \mathcal{N}_+$ the transition probabilities are defined as

$$\rho_{ij} = \begin{cases} 1/4, & \text{if } v_j \in N(v_i) \text{ or } j = 0 \\ 0, & \text{otherwise.} \end{cases}$$

Finally, for all items $i \in \mathcal{N}$, we have an arrival rate of $\lambda_i = 1/n$ and a price of $p_i = 1$. Out of these items, at most k can be selected.

The goal in VCC is to choose a minimum-cardinality set of vertices such that every edge is incident to at least one of the chosen vertices. Let $U^* \subseteq V$ be a minimum vertex cover in G. We show that the optimal assortment for the instance $\mathcal{M}(\mathcal{I})$ satisfies the following properties:

a. $R(S^*) \ge \frac{3}{4} + \frac{k}{4n}$ when $|U^*| \le k$.
b. $R(S^*) \le \frac{3}{4} + \frac{k}{4n} - \frac{\alpha}{16}$ when $|U^*| \ge (1+\alpha)k$.

This implies that (Cardinality-Assort) cannot be approximated within a factor larger than $1 - \frac{\alpha}{16}$ unless P = NP. To see this, note that the ratio between $\frac{3}{4} + \frac{k}{4n} - \frac{\alpha}{16}$ and $\frac{3}{4} + \frac{k}{4n}$ is monotone-increasing in k, meaning that the maximum value attained is $1 - \frac{\alpha}{16}$.

Case (a). $|U^*| \le k$. In this case, we can augment U^* with $k - |U^*|$ additional vertices chosen arbitrarily from $V \setminus U^*$ and obtain a (not necessarily minimum) vertex cover U with $|U| = k$. Now, consider the assortment $S = \{i : v_i \in U\}$, which is indeed a feasible solution. Because all prices are equal to one, we can write the expected revenue of this set as

$$R(S) = \mathbb{P}(S \prec 0) = \sum_{i \in S} \lambda_i + \sum_{i \notin S} \lambda_i \mathbb{P}_i(S \prec 0) = \frac{k}{n} + \frac{1}{n} \sum_{i \notin S} \mathbb{P}_i(S \prec 0). \tag{A.1}$$

When starting at any state $i \notin S$, the Markov chain moves to zero with probability 1/4 and gets absorbed. With probability 3/4, the Markov chain moves from i to one of the vertices in $N(i)$. Because U is a vertex cover, it follows that $N(i) \subseteq S$. Therefore, $\mathbb{P}_i(S \prec 0) = 3/4$ for all $i \notin S$. Based on these observations, for the optimal assortment S^* we have

$$R(S^*) \ge R(S) = \frac{k}{n} + \frac{3(n-k)}{4n} = \frac{3}{4} + \frac{k}{4n}.$$

Case (b). $|U^*| \ge (1+\alpha)k$. Let S be some assortment consisting of k items. In this case, Equation (A.1) is still a valid decomposition of $R(S)$, and we need to consider two cases for items $i \notin S$. If $N(i) \subseteq S$, then $\mathbb{P}_i(S \prec 0) = 3/4$ as in Case (a). However, when $N(i) \not\subseteq S$, there exists $j \in N(i)$ such that $j \notin S$. Therefore, there is a probability of 1/16 that, starting from i, the Markov chain moves to j and from there to zero. Consequently, for such items, $\mathbb{P}_i(S \prec 0) \le \frac{3}{4} - \frac{|N(i)\setminus S|}{16}$. Therefore,

$$\begin{aligned} R(S) &= \frac{k}{n} + \frac{1}{n} \sum_{i \notin S: N(i) \subseteq S} \frac{3}{4} + \frac{1}{n} \sum_{i \notin S: N(i) \not\subseteq S} \mathbb{P}_i(S \prec 0) \\ &\le \frac{3}{4} + \frac{k}{4n} - \frac{1}{16n} \sum_{i \notin S: N(i) \not\subseteq S} |N(i) \setminus S|. \end{aligned} \tag{A.2}$$

To upper bound the latter term, let $V(S)$ be the set of vertices of V corresponding to S; that is, $V(S) = \{v_i : i \in S\}$. Let $\bar{E}(S)$ be the set of edges that are not covered by $V(S)$. We have $2 \cdot |\bar{E}(S)| = \sum_{i \notin S: N(i) \not\subseteq S} |N(i) \setminus S|$. The important observation is that $|\bar{E}(S)| \ge \alpha k$. Otherwise, $V(S)$ can be augmented to a vertex cover via the addition of fewer than αk vertices, contradicting $|U^*| \ge (1+\alpha)k$. Now,

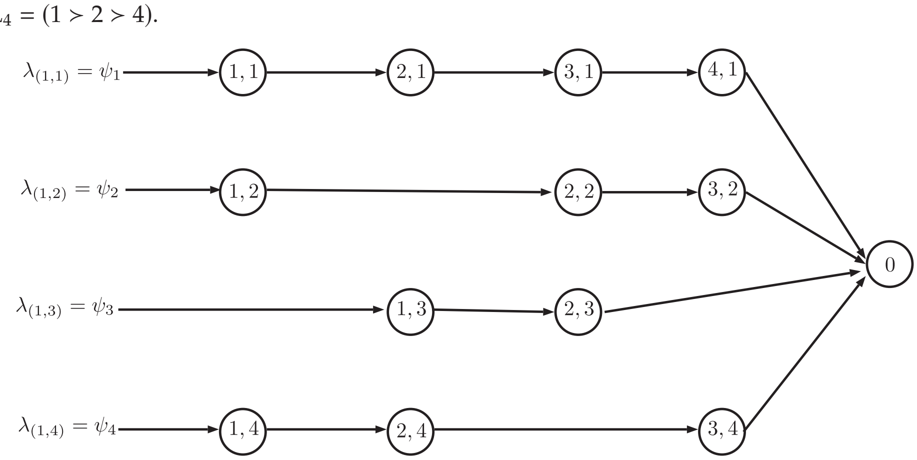
$$|\bar{E}(S)| \ge \alpha k \ge \frac{\alpha}{3} \cdot |E| = \frac{\alpha n}{2},$$

where the second inequality follows from $k > |E|/3$ and the last equality holds because $|E| = 3n/2$ as G is cubic. By inequality (A.2), we have

$$R(S) \le \frac{3}{4} + \frac{k}{4n} - \frac{|\bar{E}(S)|}{8n} \le \frac{3}{4} + \frac{k}{4n} - \frac{\alpha}{16}.$$

Because the above upper bound on $R(S)$ holds for any assortment S of k items, this must also be true for the maximum-revenue one, S^*.

Figure A.3. Sketch of Our Construction for an Instance of Four Items, in Which $L_1 = (1 \succ 2 \succ 3 \succ 4)$, $L_2 = (1 \succ 3 \succ 4)$, $L_3 = (2 \succ 3)$, and $L_4 = (1 \succ 2 \succ 4)$.



Note. Note, for example, that the state (2, 2) corresponds to the second item of L_2, which is item 3.

A.7. Proof of Theorem 6

Aouad et al. (2018) show that unconstrained assortment optimization for the distribution-over-permutations model is hard to approximate within factor $O(n^{1-\epsilon})$ for any fixed $\epsilon > 0$ even when the number of preference lists is equal to the number of items; that is, $K = n$.

We consider an instance $\mathcal{I}$ of the assortment-optimization problem for the distribution-over-permutations model with n preference lists: $L_1, \ldots, L_n$. We construct a corresponding instance $\mathcal{M}(\mathcal{I})$ of the assortment optimization problem under the Markov chain model as follows. Each of the original items in $\mathcal{N}$ has a separate copy as a state in $\mathcal{M}(\mathcal{I})$ for every list that contains it. More precisely, for every list L_i and for every $1 \le j \le |L_i|$, we have a state (j, i) corresponding to the jth most preferred item in L_i. In addition, there is a state 0 corresponding to the no-purchase option. Therefore, the set of states is

$$\mathcal{S} = \{0\} \cup \{(j,i) \; : \; 1 \le i \le n, \; 1 \le j \le |L_i|\}.$$

The transition probabilities between these states are given by

$$\rho_{((j,i),s)} = \begin{cases} 1, & \text{if } j < |L_i| \text{ and } s = (j+1, i) \\ 1, & \text{else if } j = |L_i| \text{ and } s = 0 \\ 0, & \text{otherwise.} \end{cases}$$

In other words, for each list, there is a directed path (with transition probabilities one) over its corresponding states in decreasing order of preference, ending at the no-purchase option. This construction is illustrated in Figure A.3, in which each row corresponds to a list and each column correspond to an item. Finally, the arrival rates are defined by

$$\lambda_{(j,i)} = \begin{cases} \psi_i, & \text{if } j = 1 \\ 0, & \text{otherwise,} \end{cases}$$

where ψ_i is the probability of list L_i.

To obtain a one-to-one correspondence between the solutions to $\mathcal{I}$ and $\mathcal{M}(\mathcal{I})$, it remains to ensure that, when item i is offered in $\mathcal{I}$, all of its corresponding copies (appearing in the same column) are offered in $\mathcal{M}(\mathcal{I})$ and vice versa. This restriction can be captured by the constraints $x_{(j,i)} = x_{(k,\ell)}$ for every $i, \ell \in \{1, \ldots, n\}$ such that $j \le |L_i|$, $k \le |L_\ell|$, and such that the jth item in L_i is the kth item in L_ℓ. This way, we guarantee that each column is either completely picked or completely unpicked in the instance $\mathcal{M}(\mathcal{I})$. The resulting set of inequalities specifies a constraint matrix with a single appearance of +1 and −1 in each row, and all other entries are zero. Such matrices are well known to be totally unimodular (see, for example, Schrijver 1986).

To complete the proof, note that the original instance $\mathcal{I}$ consists of n items and n preference lists, and therefore, the Markov chain instance $\mathcal{M}(\mathcal{I})$ has $O(n^2)$ states. Because the former problem is NP-hard to approximate within factor $O(n^{1-\epsilon})$ for any fixed $\epsilon > 0$, it follows that (TU-Assort) cannot be efficiently approximated within $O(n^{1/2-\epsilon})$ unless P = NP.

Appendix B. Application of Algorithm 1 to the MNL Model

In the MNL model, we are given a collection of items, $1, \ldots, n$, along with the no-purchase option, which is designated as item 0. Each item i has a utility parameter u_i and a price p_i. Without loss of generality, we can assume that $\sum_{i=0}^{n} u_i = 1$. For any given assortment S, the choice probability of each item $i \in S$ is given by

$$\pi(i, S) = \frac{u_i}{u_0 + \sum_{\ell \in S} u_\ell},$$

making the expected revenue

$$R(S) = \sum_{i \in S} \frac{u_i p_i}{u_0 + \sum_{\ell \in S} u_\ell}.$$

Blanchet et al. (2016) prove that the MNL choice model is a special case of the Markov chain model. More precisely, when $\rho_{ij} = u_j$ for all j and $\lambda_i = u_i$ for all i, the choice probabilities of the two models are identical. In this special case, our externality adjustment updates can be written as

$$p_i^S = \begin{cases} 0, & \text{if } i \in S \\ p_i - \displaystyle\sum_{j \in S} \frac{u_j p_j}{u_0 + \sum_{\ell \in S} u_\ell}, & \text{otherwise.} \end{cases}$$

Note that, in this update, the subtracted term is independent of i. Therefore, the ordering of the prices does not change after each update. Because we are picking the highest adjusted price item at each step, it follows that the optimal assortment is nested by price, that is, consists of the top ℓ priced items, for some ℓ. This is a well-known structural property (see, for instance, Talluri and van Ryzin 2004) that we recover here as a direct consequence of our algorithm. Moreover, the updated prices provide a criterion for when to stop adding items to the assortment.

References

Alimonti P, Kann V (2000) Some APX-completeness results for cubic graphs. *Theoret. Comput. Sci.* 237(1):123–134.

Aouad A, Farias VF, Levi R (2015) Assortment optimization under consider-then-choose choice models. Working paper, London Business School, London.

Aouad A, Farias V, Levi R, Segev D (2018) The approximability of assortment optimization under ranking preferences. *Oper. Res.* 66(6):1661–1669.

Bar-Yehuda R, Even S (1985) A local-ratio theorem for approximating the weighted vertex cover problem. Ausiello G, Lucertini M, eds. *North-Holland Mathematics Studies*, vol. 109 (North-Holland, Amsterdam), 27–45.

Bar-Yehuda R, Rawitz D (2006) A tale of two methods. Golumbic MC, Ben-Arroyo Hartman I, eds. *Theoretical Computer Science: Essays in Memory of Shimon Even* (Springer, Berlin), 196–217.

Bar-Yehuda R, Bendel K, Freund A, Rawitz D (2005) The local ratio technique and its application to scheduling and resource allocation problems. Golumbic MC, Ben-Arroyo Hartman I, eds. *Graph Theory, Combinatorics and Algorithms* (Springer, Boston), 107–143.

Blanchet JH, Gallego G, Goyal V (2016) A Markov chain approximation to choice modeling. *Oper. Res.* 64(4):886–905.

Buchbinder N, Feldman M, Naor JS, Schwartz R (2014) Submodular maximization with cardinality constraints. *Proc. 25th Annual ACM-SIAM Sympos. Discrete Algorithms* (Society for Industrial and Applied Mathematics, Portland, OR), 1433–1452.

Calinescu G, Chekuri C, Pál M, Vondrák J (2011) Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.* 40(6):1740–1766.

Chow YS, Robbins H, Siegmund D (1971) *Great Expectations: The Theory of Optimal Stopping* (Houghton Mifflin, Boston).

Davis J, Gallego G, Topaloglu H (2013) Assortment planning under the multinomial logit model with totally unimodular constraint structures. Technical Report, Cornell University, Ithaca, NY.

Davis JM, Gallego G, Topaloglu H (2014) Assortment optimization under variants of the nested logit model. *Oper. Res.* 62(2):250–273.

Désir A, Goyal V, Zhang J (2014) Near-optimal algorithms for capacity constrained assortment optimization. Working paper, Technology and Operations Management, INSEAD, Fontainebleau, France.

Désir A, Goyal V, Jagabathula S, Segev D (2016) Assortment optimization under the Mallows model. Lee DD, Sugiyama M, Luxburg UV, Guyon I, Garnett R, eds. *Advances in Neural Information Processing Systems*, vol. 29 (Curran Associates, Red Hook, NY), 4700–4708.

Farias VF, Jagabathula S, Shah D (2011) Assortment optimization under general choice. Working paper, Massachusetts Institute of Technology, Cambridge.

Farias VF, Jagabathula S, Shah D (2013) A nonparametric approach to modeling choice with limited data. *Management Sci.* 59(2):305–322.

Feldman J, Topaloglu H (2017a) Technical note–Capacitated assortment optimization under the multinomial logit model with nested consideration sets. *Oper. Res.* 66(2):380–391.

Feldman JB, Topaloglu H (2015) Capacity constraints across nests in assortment optimization under the nested logit model. *Oper. Res.* 63(4):812–822.

Feldman JB, Topaloglu H (2017b) Revenue management under the Markov chain choice model. *Oper. Res.* 65(5):1322–1342.

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

Gallego G, Iyengar G, Phillips R, Dubey A (2004) Managing flexible products on a network. CORC Technical Report TR-2004-01, Columbia University, New York.

Gallego G, Ratliff R, Shebalov S (2015) A general attraction model and sales-based linear program for network revenue management under customer choice. *Oper. Res.* 63(1):212–232.

Gallego G, Li A, Truong V-A, Wang X (2016) Online personalized resource allocation with customer choice. Working paper, Hong Kong University of Science and Technology, Hong Kong.

Golrezaei N, Nazerzadeh H, Rusmevichientong P (2014) Real-time optimization of personalized assortments. *Management Sci.* 60(6): 1532–1551.

Kök AG, Fisher ML (2007) Demand estimation and assortment optimization under substitution: Methodology and application. *Oper. Res.* 55(6):1001–1021.

Kulik A, Shachnai H, Tamir T (2013) Approximations for monotone and nonmonotone submodular maximization with knapsack constraints. *Math. Oper. Res.* 38(4):729–739.

Lee J, Sviridenko M, Vondrák J (2010) Submodular maximization over multiple matroids via generalized exchange properties. *Math. Oper. Res.* 35(4):795–806.

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

McFadden D (1973) Conditional logit analysis of qualitative choice behavior. Zarembka P, ed. *Frontiers in Econometrics* (Academic Press, New York), 105–142.

McFadden D (1978) Modelling the choice of residential location. *Transportation Res. Rec.* 673:72–77.

McFadden D, Train K (2000) Mixed MNL models for discrete response. *J. Appl. Econometrics* 15(5):447–470.

Méndez-Díaz I, Miranda-Bront JJ, Vulcano G, Zabala P (2014) A branch-and-cut algorithm for the latent-class logit assortment problem. *Discrete Appl. Math.* 164(Part 1):246–263.

Nemhauser GL, Wolsey LA (1978) Best algorithms for approximating the maximum of a submodular set function. *Math. Oper. Res.* 3(3): 177–188.

Nip K, Wang Z, Wang Z (2017) Assortment optimization under a single transition model. Working paper, Tsinghua University, Beijing.

Paul A, Davis JM, Feldman J (2017) Assortment optimization and pricing under a nonparametric tree choice model. *Manufacturing Service Oper. Management* 20(3):550–565.

Plackett RL (1975) The analysis of permutations. *J. Roy. Statist. Soc. Series C Appl. Statist.* 24(2):193–202.

Ragain S, Ugander J (2016) Pairwise choice Markov chains. Lee DD, Sugiyama M, Luxburg UV, Guyon I, Garnett R, eds. *Advances in Neural Information Processing Systems*, vol. 29 (Curran Associates, Red Hook, NY), 3198–3206.

Rusmevichientong P, Shen Z-JM, Shmoys DB (2010) Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Oper. Res.* 58(6):1666–1680.

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

Schrijver A (1986) *Theory of Linear and Integer Programming* (John Wiley & Sons, Chichester, UK).

Simsek AS, Topaloglu H (2017) Technical note—An expectation-maximization algorithm to estimate the parameters of the Markov chain choice model. *Oper. Res.* 66(3):748–760.

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

Thurstone LL (1927) A law of comparative judgment. *Psych. Rev.* 34(4):273–286.

Train KE (2009) *Discrete Choice Methods with Simulation* (Cambridge University Press, New York).

Williams HW (1977) On the formation of travel demand models and economic evaluation measures of user benefit. *Environ. Planning A* 9(3):285–344.

Zhang D, Cooper WL (2005) Revenue management for parallel flights with customer-choice behavior. *Oper. Res.* 53(3):415–431.