Submitted to *Operations Research*
manuscript (Please, provide the manuscript number!)

Capacity Constrained Assortment Optimization under the Markov Chain based Choice Model

Antoine Désir
Department of Industrial Engineering and Operations Research, Columbia University, ad2918@columbia.edu

Vineet Goyal
Department of Industrial Engineering and Operations Research, Columbia University, vg2277@columbia.edu

Danny Segev
Department of Statistics, University of Haifa, Haifa 31905, Israel, segevd@stat.haifa.ac.il

Chun Ye
Department of Industrial Engineering and Operations Research, Columbia University, cy2214@columbia.edu

Assortment optimization is an important problem that arises in many practical applications such as retailing and online advertising. In such settings, the goal is to select a subset of items to offer from a universe of substitutable items in order to maximize expected revenue when consumers exhibit a random substitution behavior. We consider a capacity constrained assortment optimization problem under the Markov Chain based choice model, recently considered in Blanchet et al. (2016). In this model, the substitution behavior of customers is modeled through transitions in a Markov chain. Capacity constraints arise naturally in many applications to model real-life constraints such as shelf space or budget limitations. We show that the capacity constrained problem is APX-hard even for the special case when all items have unit weights and uniform prices, i.e., it is NP-hard to obtain an approximation ratio better than some given constant. We present constant factor approximations for both the cardinality and capacity constrained assortment optimization problem for the general Markov chain model. Our algorithm is based on a "local-ratio" paradigm that allows us to transform a non-linear revenue function into a linear function. The local-ratio based algorithmic paradigm also provides interesting insights towards the optimal stopping problem as well as other assortment optimization problems.

Key words: 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 this problem, the goal is to select a subset from a universe of substitutable items to offer to customers in order to maximize the expected revenue. The demand of any item depends on the substitution behavior of the customers that is captured mathematically by a choice model that specifies the probability that a random consumer selects a particular item from any given offer set. The objective of the decision maker is to identify an offer set that maximizes expected revenue.

Many parametric choice models have extensively been studied in the literature in diverse areas including marketing, transportation, economics, and operations management. The Multinomial logit (MNL) model is by far the most popular model in practice due to its tractability (Talluri and Van Ryzin 2004). However, some of the simplifying assumptions behind this model, such as the Independence of Irrelevant Alternatives property, make it inadequate for many applications. Consequently, more complex choice models have been developed to capture a richer class of substitution behaviors. Such models include the nested logit model (Williams 1977) and the mixture of Multinomial logit model (McFadden et al. 2000). Nonetheless, the increase in model complexity makes their estimation and assortment optimization problems significantly more difficult. Hence, one of the key challenges in assortment planing is choosing a model that strikes a good balance between its predictability and tractability, as there is a fundamental tradeoff between these desirable properties.

In a recent paper, Blanchet et al. (2016) consider a Markov chain based choice model. Here, customer substitution is captured by a Markov chain, where each item (including the no-purchase option) corresponds to a state, and substitutions are modeled using transitions in the Markov chain. 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 known choice models in the literature including MNL, Generalized Attraction Model (GAM) (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 is polynomial time solvable. Zhang and Cooper (2005) also consider the Markov chain model in the context of airline revenue management, and present a simulation study. In a recent paper, Feldman and Topaloglu (2014b) study the network revenue management problem under the Markov chain model and give a linear programming based algorithm.

In this paper, we consider the capacity constrained assortment problem under the Markov chain model. In this problem, 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, we can formulate the capacity constrained assortment optimization problem as

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

where $\mathcal{N}$ denotes the universe of substitutable items and $R(S)$ denotes the expected revenue for any assortment $S\subseteq\mathcal{N}$ under the Markov chain model. For the special case of uniform item weights (i.e. $w_i = 1$ for all i), the capacity constraint reduces to a constraint on the number of items in the assortment. We refer to this setting as the cardinality constrained assortment optimization problem:

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

The 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. Capacity constrained assortment optimization has been studied in the literature for many parametric choice models. Davis et al. (2013) give an exact algorithm for MNL under cardinality constraint, and more generally, under totally-unimodular constraints. Gallego and Topaloglu (2014) propose an exact algorithm for the cardinality constrained problem for a special case of the nested logit model. More recently, Feldman and Topaloglu (2014a) 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 (PTAS) for the cardinality constrained assortment problem under a mixture of MNL choice model. Désir and Goyal (2014) propose a fully polynomial-time approximation scheme (FPTAS) for the capacity constrained assortment problem under both the nested logit and the mixture of MNL models.

1.1. Our contributions

Hardness of Approximation. We show that the capacity constrained assortment optimization problem under the Markov chain model 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, i.e. to a cardinality constraint. It is interesting to note that, while the unconstrained assortment optimization problem under the Markov chain choice model can be solved optimally in polynomial time, the cardinality constrained problem is 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.

We also consider the case of totally-unimodular (TU) constraints on the assortment. Note that a cardinality constraint is a special case of TU constraints. These capture a wide range of practical constraints such as precedence, display locations, and quality consistent pricing constraints (Davis et al. (2013)). We show that the assortment optimization problem under general totally-unimodular (TU) constraints for the Markov chain choice model 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.

Approximation Algorithms: Uniform Prices. The above hardness results motivate us to consider approximation algorithms for the capacity constrained assortment optimization problem under the Markov chain choice model. For the special case, when all item prices are equal, we show that the revenue function is 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 of uniform prices, we can get a $(1-1/e)$-approximation for more general constraints such as a constant number of capacity constraints (Kulik et al. (2013)) and matroid constraint (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 products are horizontally-differentiated, i.e., 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.

Approximation Algorithms: General Prices. For the general case of non-uniform item prices, the revenue function is neither submodular nor monotone. Moreover, the performance of the greedy algorithm can be arbitrarily bad even for the cardinality constrained problem. Our main contribution in this paper is to present a "local-ratio" based algorithm to obtain a $(1/2-\epsilon)$-approximation for the cardinality constrained assortment optimization problem under the Markov chain model. The running time of our algorithm is polynomial in the input size and $1/\epsilon$. The algorithm is based on a "local-ratio" paradigm that builds the solution iteratively. In each iteration, the algorithm makes an appropriate greedy choice and then constructs a modified instance such that the final objective value is the sum of the objective value of the current solution and the objective value of the solution in the modified instance. Therefore, the local-ratio paradigm allows us to capture the externality of our action in each iteration on the remaining instance by constructing an appropriate modified instance; thereby, linearizing the revenue function even though the original objective function is non-linear. This technique may be of independent interest. We also obtain a $(1/3-\epsilon)$-approximation for the general capacity constrained assortment optimization problem using the local-ratio paradigm. Our approach also provides an alternative strongly-polynomial exact algorithm for the unconstrained assortment optimization problem under the Markov chain model.

Computational Results. We conduct a computational study to compare the numerical performance of our algorithm. We focus on two particular issues: performance and computational efficiency. We present an exact mixed-integer programming (MIP) formulation of the problem to compute the exact optimal solution for comparison. In the numerical experiments, we observe that the practical performance of our algorithm is significantly better than its worst-case theoretical guarantee. Specifically, although the theoretical guarantee 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 in our experiments. With respect to computational efficiency, our algorithm is scalable and terminates in a few seconds, and in fact, within one minute in the worst case over all large instances tested. On the other hand, the MIP does not terminate even within a time limit of 2 hours on most of these large instances ($n=200$).

1.2. The Markov chain model and Notations

We denote the universe of n products by the set $\mathcal{N}=\{1,2,\dots,n\}$ and the no-purchase option by 0, 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 the transition probabilities ρ_{ij} for all $i\in\mathcal{N}_+, j\in\mathcal{N}_+$. If a retailer chooses to offer a subset of products S to consumers, then the corresponding states in S of the Markov chain become absorbing states. A customer arrives in state i with probability λ_i if the state is absorbing. Otherwise, the customer transitions to a different state $j\neq i$ and the process continues until the customer reaches an absorbing state. In other words, the probability of a random customer purchasing product i with S being the offer set of products 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 0 with non-zero probability. For a given offer set $S\subseteq\mathcal{N}$, let $\pi(i,S)$ be the choice 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, the 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 $\mathbb{P}(U\prec V\prec W)=\sum_{j=1}^{n}\lambda_j\mathbb{P}_j(U\prec V\prec W)$. Note that 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, $W=\emptyset$).

1.3. Outline

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

2. 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.

2.1. APX-hardness for cardinality constraint with uniform prices

We show that Cardinality-Assort is APX-hard, i.e., it is NP-hard to approximate within a given constant. In particular, we prove this result even when all items have uniform prices.

Theorem 1. *Cardinality-Assort is APX-hard, even when all items have equal prices.*

Our proof is based on gap preserving reduction from the minimum vertex cover problem on 3-regular (or cubic) graphs, to which we refer to as VCC. This problem is known to be APX-hard (see Alimonti and Kann (2000)). In other words, for some constant $\alpha > 0$, it is NP-hard to distinguish whether the minimum-cardinality vertex cover is of size at most k or at least $(1+\alpha)k$ for cubic graphs. Given any instance $\mathcal{I}$ of the VCC problem with a cubic graph $G = (V, E)$ and $k > |E|/3$, we construct an instance $\mathcal{M}(\mathcal{I})$ of Cardinality-Assort as follows. We consider a Markov chain with states corresponding to each vertex in G and an additional state 0 corresponding to the no-purchase item 0. Each state has a transition to state 0 with probability 1/4. In addition, each state has transitions to the states corresponding to their neighbors in G with probability 1/4 each (since G is a 3-regular graph, the sum of transition probabilities out of any state is one).

To prove the hardness result, we establish the following two properties: *i*) if the minimum vertex cover of instance $\mathcal{I}$ has size at most k, then the optimal expected revenue for instance $\mathcal{M}(\mathcal{I})$ is at least $\left(\frac{3}{4} + \frac{k}{4n}\right)$, and *ii*) if the minimum vertex cover of instance $\mathcal{I}$ has size at least $(1+\alpha)k$, then the optimal expected revenue for instance $\mathcal{M}(\mathcal{I})$ is at most $\left(\frac{3}{4} + \frac{k}{4n} - \frac{\alpha}{16}\right)$. Therefore, there is a constant gap between the optimal objective value for instance $\mathcal{M}(\mathcal{I})$ of Cardinality-Assort for the two cases. Since it is NP-hard to distinguish between the two cases for instance $\mathcal{I}$, this implies that it is NP-hard to approximate Cardinality-Assort better than some constant (strictly smaller than 1); thereby, proving the APX-hardness of Cardinality-Assort. We would like to note that Cardinality-Assort is APX-hard even for the special case of uniform item prices. Furthermore, our hardness reduction provides interesting insights towards the structure of difficult instances of the problem.

We present a detailed proof of Theorem 1 in Appendix A.

2.2. Totally-unimodular constraints

We consider the assortment optimization under the Markov chain model for the more general case of totally-unimodular constraints. Let $x^S \in \{0,1\}^{|\mathcal{N}|}$ denote the incidence vector for any assortment $S \subseteq \mathcal{N}$ 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 follows:

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

Here, A is a totally-unimodular matrix, and b is an integer vector. Note that the cardinality constraint in Cardinality-Assort is a special case of TU-Assort. We show that TU-Assort is NP-hard to approximate within factor $O(n^{1/2-\epsilon})$, for any fixed $\epsilon > 0$ for the Markov chain model. 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.

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. (2015) 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 $\epsilon > 0$ (n is the number of substitutable items). In an instance of the assortment optimization problem over the distribution over permutations model, we are given a collection of items $\mathcal{N} = \{1,\dots,n\}$ with prices $p_1 \le \cdots \le p_n$, respectively. In addition, we are given an arbitrary (known) distribution on K preference lists, $L_1,\dots,L_K$, 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 his/her list. The goal is to find an assortment such that the expected revenue is maximized.

Theorem 2. *TU-Assort cannot be approximated in polynomial-time within a factor $O(n^{1/2-\epsilon})$, for any fixed $\epsilon > 0$, unless $P = NP$.*

We present the proof in Appendix A.

3. Special Case: Uniform Price Items

In this section, we consider a special case of Cardinality-Assort when item prices are uniform, and prove that this setting can be efficiently approximated within factor $1 - 1/e$.

3.1. Constant factor approximation algorithm

When all prices are equal, we show that the revenue function is submodular and monotone. Using the classical result of Nemhauser and Wolsey (1978), we have that a greedy algorithm guarantees a $(1-1/e)$-approximation for Cardinality-Assort for this special case of uniform prices. We start with a few definitions.

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 $R(S \cup \{i\}) - R(S) \geq R(T \cup \{i\}) - R(T)$.

Theorem 3. *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 $\mathcal{N}$. Since item prices are identical, for every subset S and item $i \in \mathcal{N} \backslash S$, we have

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

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 due to 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 non-negative. Moreover, the submodularity of R follows from the fact that for all $S \subseteq T$, we have

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

□

Therefore, from the classical result of Nemhauser and Wolsey (1978) for maximizing a monotone submodular function subject to a cardinality constraint, we know that the greedy algorithm gives a $(1-1/e)$-approximation bound for Cardinality-Assort with uniform prices. Algorithm 1 describes this procedure in detail. Note that for uniform prices, when $|S| < k < n$, the condition in Step 2 that there exist $i \in \mathcal{N} \backslash S$ such that $R(S \cup \{i\}) - R(S) \geq 0$ is redundant as the revenue function is monotone, which is not necessarily true for the case of arbitrary prices. Therefore, we include this condition to describe the greedy algorithm for the general case to discuss implications for arbitrary prices.

Algorithm 1 Greedy Algorithm

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

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

3.2. Bad examples for arbitrary prices

The approximation guarantees we establish for uniform prices do not extend to the more general setting with arbitrary prices, even for Cardinality-Assort. In what follows, we point out the drawbacks of the natural greedy heuristics, including Algorithm 1, in approximating Cardinality-Assort for arbitrary prices. Intuitively, the performance of Algorithm 1 for general prices can be bad since it can make a low-price item absorbing that subsequently blocks all probabilistic transitions going into high price items. We formalize this intuition in the following lemma.

Lemma 1. *For arbitrary instances of Cardinality-Assort with a cardinality constraint of k, Algorithm 1 can compute solutions whose expected revenue is only $O(1/k)$ times the optimum.*

Proof. 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 0, except for λ_s which is equal to 1. Moreover

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

where $\epsilon \leq 1/(2k)$. Figure 1 provides a graphical representation of this instance. Algorithm 1 first picks item s as $R(\{s\}) = (1/k) + \epsilon$ while $R(\{i\}) = (1/k)$, 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

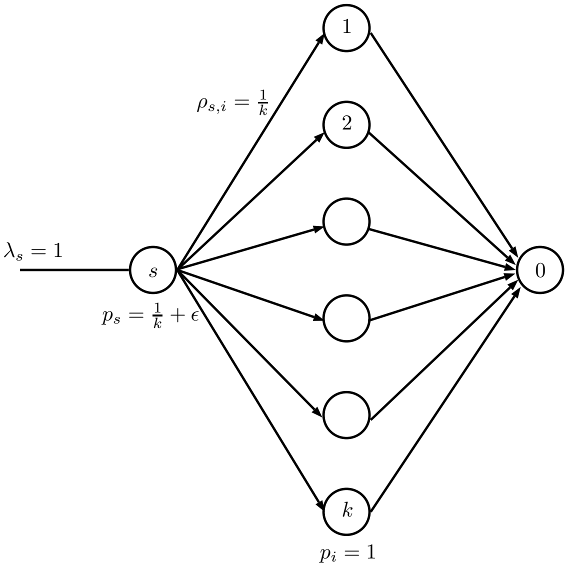

Figure 1 A bad example for Algorithm 1.

$(1/k)+\epsilon$. However, the optimal solution is to offer items 1 to k, which gives a revenue of 1 in total. When ϵ tends to 0, the approximation ratio goes to $1/k$. □

In fact, we can show that the above example is the worst possible and Algorithm 1 gives a $1/k$-approximation for Cardinality-Assort.

Lemma 2. *Algorithm 1 guarantees a $1/k$-approximation for Cardinality-Assort.*

We present the proof of the above lemma in Appendix B.

Modified Greedy Algorithm. The bad instance for Algorithm 1 shows that the algorithm may focus too much 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, where there is no restriction on the number of items picked. This solution can be computed via an algorithm proposed by Blanchet et al. (2016) (we also give an alternative strongly-polynomial algorithm for the unconstrained problem in Section 4.4). 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 an unconstrained optimal solution, and sets these states to be absorbing. Algorithm 2 describes this procedure.

Algorithm 2 Greedy Algorithm on Optimal Unconstrained Assortment

```
Let $U^*$ be an optimal solution to the unconstrained problem.
Sort items of $U^*$ in decreasing order of $\mathbb{P}(i \prec U^*_+ \backslash \{i\}) \cdot p_i$.
Return $S = \{$top $k$ items in the sorted order$\}$.
```

We show in the following lemma that even Algorithm 2 performs poorly in the worst case. In fact, we present an example where every subset of k items of the optimal solution U^* has revenue a factor k away from the optimal.

Lemma 3. *There are instances where the revenue obtained by Algorithm 2 is far from optimal by a factor of $k/|U^*|$ where k is the upper bound in the cardinality constraint.*

Proof. Consider the following instance of the problem with $n+2$ items (or states). We have a state s and states $i = 1, \ldots, n$ and state 0 corresponding to the no-purchase option. The arrival rates are all equal to 0, except for λ_s which is equal to 1. Moreover

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

where $\epsilon > 0$. Figure 2 provides a graphical representation of this instance. For this example, the unconstrained optimal assortment is $U^* = \{1, \ldots, n\}$, and the 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 0, the approximation ratio goes to $k/|U^*|$. □

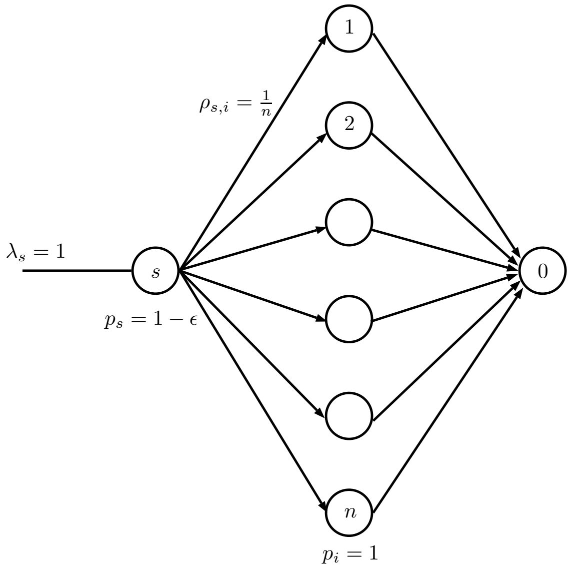

Figure 2 A bad example for Algorithm 2.

The poor performance of Algorithm 2 on the above example 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. It is worth noting that the lower bound of $k/|U^*|$ for Algorithm 2 is tight, as stated in the following lemma, whose proof is given in Appendix C.

LEMMA 4. *Algorithm 2 guarantees a $k/|U^*|$-approximation algorithm to* Cardinality-Assort*.*

The analysis of the two greedy variants for the cardinality constrained assortment optimization under the Markov chain model provides important insights that we use towards designing a good algorithm for the problem.

4. Local Ratio based Algorithm Design

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

4.1. High-level ideas

As the example in Figure 1 illustrates, Algorithm 1 could end up with a highly suboptimal solution due to picking items that cannibalize, i.e. block, 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 in the example in Figure 2. When there is 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 that builds the solution iteratively. Let $\mathcal{M}_t$ denote the problem instance in any iteration t. The algorithm (ALG) considers the following two steps in each iteration t:

1. *Greedy Selection.* Define an appropriate consideration set C_t of items, and pick the “highest price” item from C_t.

2. *Instance Update.* 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 from the item selected in iteration t.

The instance update step linearizes the revenue function even though the original revenue function is non-linear, which is crucial for our iterative solution approach. We can also view the update rule as a framework to capture the externality of our actions in each iteration of the algorithm. To completely specify the algorithm, we need to provide a precise definition for the consideration set in the greedy 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 4.2. The consideration set, however, depends on the particular optimization problem being considered and will be defined later on. The intuition is to include items whose incremental revenue is above an appropriately chosen threshold. Our algorithm can 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)). Therefore, we will interchangeably refer to the instance updates as local-ratio updates. However, we would like to note that the local-ratio framework does not provide a general recipe for designing an update rule or analyzing the performance bound. In most algorithms in 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 do not even know of any good LP formulation and the instance update rule requires new ideas.

4.2. Instance update in local ratio algorithm

Notation. Given an instance $\mathcal{M}$ of the Markov chain model, we define an updated instance $\mathcal{M}(S)$ given that S is made absorbing by modifying the item prices as well as the probability transition matrix. Note that we index the updates by a set S. Therefore, the instance $\mathcal{M}_t$ introduced in the preceding discussion is going to 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 will denote by p_i^S the updated price of item i, and by ρ_{ij}^S the updated transition probabilities for every $i \in \mathcal{N}, j \in \mathcal{N}_+$. Note that we do not change the arrival rate to any state, i.e., $\lambda_i^S = \lambda_i$ 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 the instance $\mathcal{M}(S)$.

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

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

The above 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, provided 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. Note that the adjusted prices can be negative, corresponding to the situation where adding an item decreases the overall revenue. The price update is explicitly described in Figure 3.

Transition probabilities update. Since the subset of states S is set to be absorbing, we will simply redirect the outgoing probabilities from all states in S to 0. This is described in Figure 3.

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:

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

Figure 3 Instance update in local-ratio algorithm.

We would like to note that 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)).

4.3. Structural properties of the updates

We first show that the local-ratio updates allow us to linearize the revenue function.

Lemma 5. *$R(S_1 \cup S_2) = R(S_1) + R^{S_1}(S_2)$ for every $S_1, S_2 \subseteq \mathcal{N}$.*

Proof. Assume without lost of generality that $S_1 \cap S_2 = \emptyset$, since the items in $S_1 \cap S_2$ all have 0 as their adjusted price and we can then apply the proof to $S_2 \backslash S_1$. Using the definition of the local ratio updates, we have

$$R^{S_1}(S_2) = \sum_{i \in S_2} \mathbb{P}^{S_1}(i \prec S_{2+} \backslash \{i\}) p_i^{S_1}$$

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

With the definition of ρ^{S_1}, note that all items of S_1 are redirected to 0. 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,

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

where the second equality holds since

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

as by the Markov property, both the left and right terms in the above equality denote the probability that we will 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+}. □

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

LEMMA 6. *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.*

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 the above identity clearly holds for the transition matrix updates. The proof is similar to that of Lemma 5, and is presented in Appendix D. Putting the previous two lemmas together gives the following claim.

LEMMA 7. $R^{S_1}(S_2 \cup S_3) = R^{S_1}(S_2) + R^{S_1 \cup S_2}(S_3)$ *for any pairwise-disjoint sets $S_1, S_2, S_3 \subseteq \mathcal{N}$.*

4.4. Warm-up: Exact algorithm for the unconstrained problem

As a warmup, we first present an alternative exact algorithm for the unconstrained assortment optimization problem under the Markov chain model by using the local-ratio 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 8. *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. From Lemma 5, we have that

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

Now, $p_i \geq [\max_{j\in S} p_j]^+$ and

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

which implies $R(S \cup \{i\}) \geq R(S)$. □

The Algorithm. Based on the above lemma, we present an alternative exact algorithm for the unconstrained assortment optimization problem under the Markov chain model. In particular, we define the consideration set in each iteration to be the set of all items. Therefore, we select the highest adjusted price item in every iteration (breaking ties arbitrarily) and update the prices and transition probabilities according to the local ratio updates described in Figure 3. This selection and updating process is repeated until all adjusted prices are non-positive, as explained in Algorithm 3.

Algorithm 3 Local Ratio 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 4. *Algorithm 3 computes an optimal solution for the unconstrained assortment optimization problem under the Markov chain model.*

Proof. The correctness of Algorithm 3 is based on the observation that it is always optimal to offer the highest adjusted price item, as long as this price is non-negative. Suppose item 1 is the highest price item. From Lemma 8, 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. From Lemma 5, 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 where our current absorption set is X, and the adjusted price of every state in the modified instance $\mathcal{M}(X)$ is non-positive, then X is an optimal solution to $\mathcal{M}$. To see this, by repeated applications of Lemmas 5 and 6, we have

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

However, since the adjusted price of every state in the instance $\mathcal{M}(X)$ is non-positive, 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 $\mathcal{M}$. □

Implications. Our algorithm for the unconstrained assortment optimization over the Markov chain model provides interesting insights for some known results about the optimal stopping problem and the assortment optimization over the MNL model. Blanchet et al. (2016) relate the unconstrained assortment problem to the optimal stopping time on a Markov chain (see 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 transition according to the transition probabilities of the Markov chain. Moreover, there is an absorbing state 0 with price $p_0 = 0$. Algorithm 3 for the unconstrained assortment optimization problem gives an alternative strongly polynomial time algorithm for the optimal stopping problem.

Blanchet et al. (2016) prove that the MNL choice model is a special case of the Markov chain based choice model. Therefore, by analyzing Algorithm 3 to solve the assortment optimization over the MNL model, we can recover the structure of the optimal assortment being nested by prices, i.e., the optimal assortment consists of the ℓ top-priced items for some ℓ. We give an explicit expression for our local ratio updates when the underlying choice model is MNL in Appendix E.

5. Cardinality Constrained Assortment Optimization for General Case

In this section, we present a $(1/2 - \epsilon)$-approximation for the cardinality constrained assortment optimization under the Markov chain model, for any fixed $\epsilon > 0$. Following the local-ratio framework described in Section 4, our algorithm for the cardinality constrained case also selects a state with high adjusted price in each step from an appropriate consideration set. The consideration set is defined to avoid picking states that have a high adjusted price but capture very little demand. In particular, the consideration set includes only items whose incremental revenue is at least a certain threshold.

The Algorithm. Our algorithm is iterative and 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 $S_t = \{\sigma_1, \dots, \sigma_t\}$. 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 local ratio update described in Figure 3, 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.

Guessing the value of $R(S^*)$. Since the optimal revenue $R(S^*)$ is not known a-priori, we need to describe how the value of $R(S^*)$ is approximately guessed to complete the algorithm's description. A natural upper bound for $R(S^*)$ is $R(U^*)$, when U^* is the optimal unconstrained solution. From Lemma 4, we know that $R(S^*) \geq \frac{k}{|U^*|} R(U^*)$. Now, given an accuracy parameter $0 < \epsilon < 1$, let

$$\begin{aligned} B_j &= \frac{k}{|U^*|} R(U^*)(1+\epsilon)^j, \quad j = 1, \dots, J \\ J &= \min\{ j \in \mathbb{N} \,:\, B_j \geq R(U^*) \}. \end{aligned} \tag{2}$$

Note that $J = O(\frac{1}{\epsilon} \log k)$. 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 runs. Algorithm 4 describes the resulting procedure for a particular choice of B_j and threshold α for the consideration set. Algorithm 5 describes the full procedure for any given $\epsilon > 0$.

Algorithm 4 Algorithm with guess B_j and threshold α

```
Let $S$ be the set of states picked so far, starting with $S = \emptyset$.
For all $S$, let $C(S) = \{i \in \mathcal{N} \backslash S : R^S(\{i\}) \geq \frac{\alpha \cdot B_j}{k}\}$.
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 $S$.
Return $S$.
```

Algorithm 5 Local-ratio Algorithm for Cardinality-Assort with threshold α

```
Given any $\epsilon > 0$, let $J$ and $B_j$, $j \in [J]$ be as defined in (2).
For all $j \in [J]$, let $S_j$ be the solution returned by Algorithm 4 with guess $B_j$ and threshold $\alpha$
Return $\arg\max_{j \in [J]} R(S_j)$.
```

5.1. Technical Lemmas

Prior to analyzing the performance guarantee of our algorithm, we present two technical lemmas. We start by arguing that the revenue function is sublinear for general item prices.

Lemma 9. *For all $S_1, S_2 \subseteq \mathcal{N}$ consisting only of non-negative priced items, $R(S_1 \cup S_2) \leq R(S_1) + R(S_2)$.*

Proof. We have that

$$\begin{aligned} R(S_1 \cup S_2) &= \sum_{j \in S_1} \mathbb{P}(j \prec (S_1 \cup S_2)_+ \setminus \{j\}) \cdot p_j + \sum_{j \in S_2 \setminus S_1} \mathbb{P}(j \prec (S_1 \cup S_2)_+ \setminus \{j\}) \cdot p_j \\ &\leq \sum_{j \in S_1} \mathbb{P}(j \prec (S_1)_+ \setminus \{j\}) \cdot p_j + \sum_{j \in S_2} \mathbb{P}(j \prec (S_2)_+ \setminus \{j\}) \cdot p_j \\ &= R(S_1) + R(S_2), \end{aligned}$$

where the first inequality follows as for any $j \in S_i$ $(i = 1, 2)$, $\mathbb{P}(j \prec (S_1 \cup S_2)_+ \setminus \{j\}) \leq \mathbb{P}(j \prec (S_i)_+ \setminus \{j\})$. □

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 allows us to partition the items of S^* according to the moment they disappear from the consideration set (since either their adjusted revenue becomes too small or they get picked by the algorithm). More precisely, let $Z_0 = S^*$ and for all $t \geq 1$, we define the following sets:

- $Z_t = S^* \cap C_t$ denotes the items of S^* which are in the consideration set C_t.
- $Y_t = Z_{t-1} \backslash Z_t$ denotes the items of S^* which 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 which have a non-negative 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 10. *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. Since our algorithm picks the highest adjusted price item, σ_t, in the consideration set C_t, we have $p_{\sigma_t}^{S_{t-1}} \geq p_i^{S_{t-1}} \geq 0$ for all items $i \in Z_t$. Therefore, by Lemma 8,

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

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 7, $R^{S_{t-1}}(Z_t \cup \{\sigma_t\}) = R^{S_{t-1}}(\{\sigma_t\}) + R^{S_t}(Z_t)$. 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 since removing all negative adjusted price items can only increase net revenue, and the last inequality follows from Lemma 9. Adding $R(S_{t-1})$ on both sides of the inequality yields the desired inequality by Lemma 5.

Case (b): $\sigma_t \in S^*$. From Lemma 7, $R^{S_{t-1}}(Z_t) = R^{S_{t-1}}(\{\sigma_t\}) + R^{S_t}(Z_t \backslash \{\sigma_t\})$. Then, similar to the previous case, we have

$$R^{S_t}(Z_t \backslash \{\sigma_t\}) \leq R^{S_t}((Z_{t+1} \cup Y_{t+1}^+) \backslash \{\sigma_t\}) \leq R^{S_t}(Z_{t+1}) + R^{S_t}(Y_{t+1}^+ \backslash \{\sigma_t\}).$$

Note that $R^{S_t}(Y_{t+1}^+ \backslash \{\sigma_t\}) = R^{S_t}(Y_{t+1}^+)$ since $p_{\sigma_t}^{S_t} = 0$ 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 the above result, we obtain the following claim.

Lemma 11. *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 10 over $j = 1, \ldots, t$, we obtain a telescopic sum which yields

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

Since every item in S^* must have non-negative price and $S^* = Z_1 \cup Y_1$ by definition, we have $R(S^*) \leq R(Z_1) + R(Y_1)$ by sublinearity of the revenue function (see Lemma 9). Combining these two inequalities concludes the proof. □

5.2. Analysis of the Local-Ratio Algorithm

We show that the local-ratio algorithm gives a $(1/2-\epsilon)$-approximation for Cardinality-Assort for any fixed $\epsilon > 0$. In particular, we have the following theorem.

THEOREM 5. *For any fixed $\epsilon > 0$, Algorithm 5 gives a $(1/2 - \epsilon/2)$-approximation for Cardinality-Assort. Moreover, the running time is polynomial in the input size and $1/\epsilon$.*

Proof. For a fixed $\epsilon > 0$, let j^* be such that $\frac{R(S^*)}{1+\epsilon} \leq B_{j^*} \leq R(S^*)$. Let $B = B_{j^*}$ and 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.

Case 1. If the algorithm stops after completing step k, then by linearity of the revenue when using the local ratio updates (Lemmas 5 and 6), the resulting solution S_k has a revenue of

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

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

Case 2. Now, suppose the algorithm stops at the end of step $k' < k$, after discovering that $C_{k'+1} = \emptyset$. From Lemma 11, 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, since $C_{k'+1} = \emptyset$, this implies that $Z_{k'+1} = \emptyset$. Moreover, from Lemma 9, 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 since $\sum_{j=1}^{k'+1} |Y_j^+| \leq k$ 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.

Running time. Algorithm 5 considers $J = O(\frac{1}{\epsilon}\log n)$ guesses for $R(S^*)$. For any given guess B_j, the running time of Algorithm 4 is polynomial in the input size. Therefore, the overall running time of Algorithm 5 is polynomial in the input size and $1/\epsilon$. □

Tight example. We show that Algorithm 5 is tight in the following sense: consider Algorithm 4 with input guess as the true value of $R(S^*)$ and threshold $\alpha = 1/2$, then there are instances for which the approximation ratio is 1/2. In particular, we consider an instance with 3 items. The Markov chain has 4 states $\mathcal{N}_+ = \{s, 1, 2, 0\}$. The prices are: $p_s = 1, p_1 = p_2 = 2$. 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 4. Consider the cardinality constrained assortment problem with cardinality bound, $k = 1$. 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 4 picks either 1 or 2, obtaining a revenue of $R(S^*)/2$.

We would like to note that our algorithm runs Algorithm 4 for different guesses $B_j, j = 1, \ldots, J$ and returns the best solution across all runs. Therefore, the performance bound of our algorithm is at least $(1/2 - O(\epsilon))$ and possibly better. In fact, in our computational study, we observe that the empirical performance of our algorithm is significantly better than the theoretical bound of $(1/2 - O(\epsilon))$. We describe the computational study in Section 7. It is an interesting open question to provide a tighter analysis of the approximation bound for Algorithm 5 that returns the best solution among several guesses of $R(S^*)$.

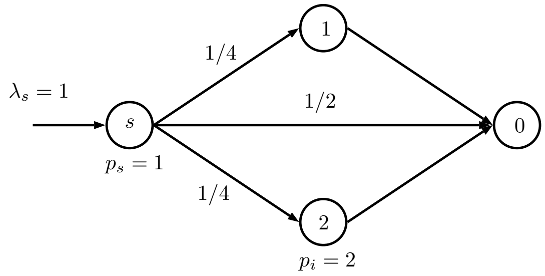

Figure 4 A tight example for Algorithm 5.

6. Capacity Constrained Assortment Optimization for General Case

In this section, we show how to approximate the capacity constrained problem under the Markov chain model 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 W on the available capacity. We assume without loss of generality that each item individually satisfies the capacity constraint, i.e., $w_i \leq W$ for all $i \in \mathcal{N}$.

The Algorithm. We describe a local-ratio 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.

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 $t \geq 1$, we select the highest adjusted price item (with respect to $p^{S_{t-1}}$, breaking ties arbitrarily) among items in the following consideration set:

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

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 local ratio update described in Figure 3. This selection and update process is repeated in every step until either the consideration set becomes empty or adding the current item violates the capacity constraint. Let t' be such a step. 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.

Guessing $R(S^*)$. As in the case of cardinality constraints, since the value of $R(S^*)$ is unknown, we need to approximately guess the value $R(S^*)$. We will use a procedure similar to the one given in Section 5, with the exception of utilizing $\frac{1}{|U^*|} R(U^*)$ as a lower bound (see proof of Lemma 2 in Appendix B), where U^* is the optimal unconstrained solution. In particular, we consider the following guesses for $R(S^*)$.

$$\begin{aligned} B_j &= \frac{1}{|U^*|} R(U^*)(1+\epsilon)^j, \; j = 1, \ldots, J \\ J &= \min\{j \in \mathbb{N} \, : \, B_j \geq R(U^*)\}. \end{aligned} \tag{4}$$

Note that $J = O(\frac{1}{\epsilon} \log n)$. Algorithm 6 provides a description of our approximation algorithm for Capacity-Assort, given a particular guess B_j for $R(S^*)$ and threshold α, while Algorithm 7 describes the complete procedure.

6.1. Analysis

To analyze the above algorithm, it is convenient to have a technical lemma similar to Lemma 11. 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, the exact same lemma holds. We therefore do not restate this claim and its proof, as these are identical to those of Lemma 11. The following theorem shows that the local-ratio algorithm gives a $(1/3 - \epsilon)$-approximation for Cardinality-Assort for any fixed $\epsilon > 0$.

Algorithm 6 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 $S$, let $C(S) = \{i \in \mathcal{N} : \frac{R^S(\{i\})}{w_i} \geq \alpha \cdot \frac{B_j}{W}\}$.
3: While $\sum_{i \in S} w_i < W$ 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) If $\sum_{i \in S \cup \{i^*\}} w_i < W$, add $i^*$ to $S$.
     (c) Else return the highest revenue set among $\{i^*\}$ and $S$.
4: Return $S$.
```

Algorithm 7 Local-ratio Algorithm for Capacity-Assort with threshold α

```
1: Given any $\epsilon > 0$, let $J$ and $B_j$, $j \in [J]$ be as defined in (4).
2: For all $j \in [J]$, let $S_j$ be the solution returned by Algorithm 6 with guess $B_j$ and threshold $\alpha$
3: Return $\arg\max_{j \in [J]} R(S_j)$.
```

Theorem 6. *For any fixed $\epsilon > 0$, Algorithm 7 gives a $(1/3 - \epsilon/3)$-approximation for Capacity-Assort. Moreover, the running time is polynomial in the input size and $1/\epsilon$.*

Proof. For a fixed $\epsilon > 0$, let j^* be such that $\frac{R(S^*)}{1+\epsilon} \leq B_{j^*} \leq R(S^*)$. Let $B = B_{j^*}$ and consider the solution returned by Algorithm 6 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.

Case 1. Suppose we stop the algorithm since 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 $\alpha R(S^*)/2$, since

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

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

Case 2. On the other hand, suppose the algorithm terminates since $C_{t'+1} = \emptyset$. Using Lemma 11 adapted to the capacitated case, we have

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

Since $C_{t'+1} = \emptyset$, this implies that $Z_{t'+1} = \emptyset$. Moreover, from Lemma 9, for all $j = 1, \ldots, t'+1$, we have

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

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

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

As a result, the approximation ratio attained by our algorithm is

$$\min\left\{(1-\epsilon)\frac{\alpha}{2},\, 1-\alpha\right\}.$$

By setting $\alpha = 2/3$, we obtain an approximation factor of $(1/3 - \epsilon/3)$.

Running Time . Algorithm 7 considers $J = O(\frac{1}{\epsilon}\log n)$ guesses of $R(S^*)$. Each run of Algorithm 6 for a given guess is polynomial time. Therefore, the overall running time of Algorithm 7 is polynomial in the input size and $1/\epsilon$. □

Tight example. Our analysis is tight in the following sense. When Algorithm 7 is run with the true value of $R(S^*)$, there are instances for which the approximation ratio is 1/3. For example, consider the instance given in Figure 5. For a capacity bound of $W = 1$, the optimal assortment is $S^* = \{b, c\}$. Initially, all the 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 $S = \{a\}$ since $R(\{a\}) > R(\{d\})$ and yields a revenue of $R(S^*)/3 + O(\epsilon)$. When ϵ goes to 0, the approximation ratio goes to 1/3.

7. Computational Experiments

In this section, we present our results from a computational study to test the performance of Algorithm 5 for the cardinality constrained assortment optimization for the Markov chain choice model. In particular, we focus on testing: i) the performance of our algorithm with respect to an optimal algorithm, and ii) the running time of this algorithm. We first present a mixed-integer programming (MIP) formulation of Cardinality-Assort.

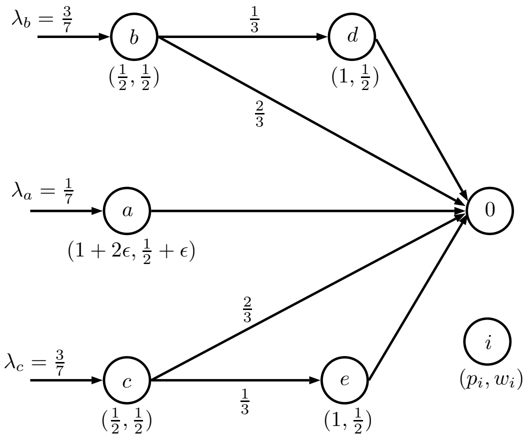
$\lambda_b = \frac{3}{7}$
b
$(\frac{1}{2}, \frac{1}{2})$
$\frac{1}{3}$
d
$(1, \frac{1}{2})$
$\frac{2}{3}$
$\lambda_a = \frac{1}{7}$
a
$(1+2\epsilon, \frac{1}{2}+\epsilon)$
0
$\frac{2}{3}$
i
(p_i, w_i)
$\lambda_c = \frac{3}{7}$
c
$(\frac{1}{2}, \frac{1}{2})$
$\frac{1}{3}$
e
$(1, \frac{1}{2})$

Figure 5 A tight example for Algorithm 7.

7.1. A mixed-integer programming formulation

We show that the following mixed-integer program (MIP) is an exact reformulation of Cardinality-Assort.

$$\begin{aligned} \max \ & \sum_{i=1}^{n} \alpha_i p_i \\ \text{s.t. } & \alpha_i + \beta_i - \sum_{j=1}^{n} \rho_{ji}\beta_j = \lambda_i, && \forall i = 1,\dots,n \\ & y_i \geq \alpha_i, && \forall i = 1,\dots,n \\ & \sum_{i=1}^{n} y_i \leq k \\ & \alpha_i \geq 0, \beta_i \geq 0, y_i \in \{0,1\}, && \forall i = 1,\dots,n. \end{aligned} \tag{5}$$

LEMMA 12. *The mixed-integer program* (5) *is an exact reformulation of Cardinality-Assort.*

Proof. Consider the following LP:

$$\begin{aligned} \max \ & \sum_{i=1}^{n} \alpha_i p_i \\ \text{s.t. } & \alpha_i + \beta_i - \sum_{j=1}^{n} \rho_{ji}\beta_j = \lambda_i, && \forall i = 1,\dots,n \\ & \alpha_i \geq 0, \beta_i \geq 0, && \forall i = 1,\dots,n. \end{aligned} \tag{6}$$

Let (α, β) be an extreme point solution to the above LP, and let $S = \{i : \alpha_i > 0\}$. Feldman and Topaloglu (2014b) show that α_i is the choice probability $\pi(i, S)$ when the assortment S is offered under the Markov chain choice model. Hence, the objective value $\sum_{i=1}^{n} \alpha_i r_i$ equals to $R(S)$. By adding the indicator variables y_i, we are restricting ourselves to the subset of feasible solutions where at most k of the α_i-s are allowed to be strictly positive. Note that the extreme points of this polytope, corresponding to the projection of the feasible space of the MIP down to the

(α, β) coordinates, are exactly the set of assortments S with cardinality at most k. Hence, (5) is a mixed-integer formulation of the cardinality constrained assortment optimization problem. □

7.2. Settings tested

We proceed 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 since we present statistics regarding approximation factors, any constant here will give identical results, so the choice of 1 is arbitrary. In 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 1 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 $\rho_{ij} = X_{ij} / \sum_{j=0}^{n} X_{ij}$ for all i, j such that $i \neq j$. Since 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 $i \neq 0$.
2. In this setting, we sparsify the transition matrix of setting 1. More precisely, we additionally generate n^2 independent random variable Z_{ij}, each following a Bernoulli distribution with parameter 0.2. For all i, j such that $i \neq j$, we set $\rho_{ij} = Z_{ij} X_{ij} / \sum_{j=0}^{n} Z_{ij} X_{ij}$, 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 in this last setting is one of a random walk. More precisely, we generate n^2 independent random variable 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 $i \neq j$. 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 $i \neq 0$.

7.3. Results

We examine how our algorithm performs in term of both approximation and running time. Table 1 shows the approximation ratio of Algorithm 5 (with $\epsilon = 0.1$) for the different settings and the different values of n. We use the MIP formulation given in (5) to compute the optimal assortment. As can be observed, the actual performance of our algorithm is significantly better than its worst case theoretical 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 5 in terms of running time for setting 2. The running times are very similar for the other

Setting	n	Approximation Ratio		# instances within x% of OPT				# instances
		Average	Minimum	2%	5%	10%	20%	
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

Table 1 Performance of Algorithm 5 for Cardinality-Assort.

settings. On the other hand, while the MIP running time can be competitive in some cases, it blows up when the number of products n gets large (see Table 2). Note that for $n = 100$, 12 out of the 100 instances had a running time of at least 30 minutes. For $n = 200$, we set a time limit of 2 hours for the MIP. Out of the 20 random instances generated, 16 reached the time limit without terminating. These numerical experiments suggest that Algorithm 5 is computationally efficient and that its numerical performance is significantly better than the theoretical worst-case guarantee.

n	Average Running Time		Maximum Running Time		# instances
	Algorithm 5	MIP	Algorithm 5	MIP	
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

Table 2 Running time of Algorithm 5 and the MIP for setting 2. ** Denotes the cases when we set a time limit of 2 hours.

We also compare the performance of Algorithm 5 with the best solution found by the MIP solver within a time limit that is equal to the running time for Algorithm 5 for the corresponding instance. Table 3 shows the ratio between the performance of Algorithm 5 and that of the best feasible solution found by the MIP within the allowed time limit as well as the duality gaps for the best feasible solution. We observe that although the solver might not even terminate, it finds good solutions within the time limit allowed. On average, the best MIP solution computed within the time limit is slightly better than the solution computed by Algorithm 5. Although, for several instances, Algorithm 5 outperforms the best MIP solution within the time limit (about 20% instances for $n = 30$ and 10% for $n = 60$). Therefore, the MIP solver spends a significant fraction of the time in reducing the duality gap and proving optimality for large instances. It is interesting open question to find a stronger LP formulation for the problem.

n	Algorithm 5/(Best MIP Solution)		# instances with ratio > 1	Duality gap		# instances
	Average	Maximum		Average	Maximum	
30	0.9800	1.0851	200	2.64 %	34.85 %	1,000
60	0.9805	1.0108	101	10.42 %	69.68 %	1,000
100	0.9787	1.0100	6	15.21 %	52.18 %	100
200	0.9821	1.0029	4	19.06 %	99.60 %	100

Table 3 Comparison of Algorithm 5 with the best MIP solution when we allow the solver the same time limit.

Acknowledgments

V. Goyal is supported in part by NSF Grants CMMI-1201116 and CMMI-1351838 (CAREER) and an IBM Faculty Award.

References

Alimonti, Paola, Viggo Kann. 2000. Some APX-completeness results for cubic graphs. *Theoretical Computer Science* **237**(1) 123–134.

Aouad, Ali, Vivek Farias, Retsef Levi, Danny Segev. 2015. The approximability of assortment optimization under ranking preferences. Working paper, available online as SSRN report 2612947.

Bar-Yehuda, Reuven, Keren Bendel, Ari Freund, Dror Rawitz. 2005. The local ratio technique and its application to scheduling and resource allocation problems. *Graph Theory, Combinatorics and Algorithms*. Springer, 107–143.

Bar-Yehuda, Reuven, Shimon Even. 1985. A local-ratio theorem for approximating the weighted vertex cover problem. *North-Holland Mathematics Studies* **109** 27–45.

Bar-Yehuda, Reuven, Dror Rawitz. 2006. A tale of two methods. *Theoretical Computer Science: Essays in Memory of Shimon Even*. Springer, 196–217.

Blanchet, Jose H., Guillermo Gallego, Vineet Goyal. 2016. A markov chain approximation to choice modeling. *Operations Research (forthcoming)* .

Buchbinder, Niv, Moran Feldman, Joseph Seffi Naor, Roy Schwartz. 2014. Submodular maximization with cardinality constraints. *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*. 1433–1452.

Calinescu, Gruia, Chandra Chekuri, Martin Pál, Jan Vondrák. 2011. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing* **40**(6) 1740–1766.

Chow, Yuan Shih, Herbert Robbins, David Siegmund. 1971. *Great expectations: The theory of optimal stopping*. Houghton Mifflin Boston.

Davis, James, Guillermo Gallego, Huseyin Topaloglu. 2013. Assortment planning under the multinomial logit model with totally unimodular constraint structures. Technical Report.

Désir, Antoine, Vineet Goyal. 2014. Near-optimal algorithms for capacity constrained assortment optimization. Available at SSRN 2543309.

Feldman, Jacob B, Huseyin Topaloglu. 2014a. Capacity constraints across nests in assortment optimization under the nested logit model. Technical report, Cornell University, School of Operations Research and Information Engineering.

Feldman, Jacob B, Huseyin Topaloglu. 2014b. Revenue management under the markov chain choice model.

Gallego, Guillermo, Richard Ratliff, Sergey Shebalov. 2015. A general attraction model and sales-based linear program for network revenue management under customer choice. *Operations Research* **63**(1) 212–232.

Gallego, Guillermo, Huseyin Topaloglu. 2014. Constrained assortment optimization for the nested logit model. *Management Science* **60**(10) 2583–2601.

Kök, A. Gürhan, Marshall L. Fisher. 2007. Demand estimation and assortment optimization under substitution: Methodology and application. *Operations Research* **55**(6) 1001–1021. doi:10.1287/opre.1070.0409.

Kulik, Ariel, Hadas Shachnai, Tami Tamir. 2013. Approximations for monotone and nonmonotone submodular maximization with knapsack constraints. *Mathematics of Operations Research* **38**(4) 729–739.

Lee, Jon, Maxim Sviridenko, Jan Vondrák. 2010. Submodular maximization over multiple matroids via generalized exchange properties. *Mathematics of Operations Research* **35**(4) 795–806.

McFadden, Daniel, Kenneth Train, et al. 2000. Mixed mnl models for discrete response. *Journal of applied Econometrics* **15**(5) 447–470.

Nemhauser, G. L., L. A. Wolsey. 1978. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research* **3**(3) 177–188.

Rusmevichientong, Paat, Zuo-Jun Max Shen, David B Shmoys. 2010. Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Operations research* **58**(6) 1666–1680.

Schrijver, Alexander. 1986. *Theory of Linear and Integer Programming*. John Wiley & Sons, Chichester.

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

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

Zhang, Dan, William L Cooper. 2005. Revenue management for parallel flights with customer-choice behavior. *Operations Research* **53**(3) 415–431.

Appendix A: Proofs of Theorems 1 and 2

Proof of Theorem 1. Consider an instance $\mathcal{I}$ of VCC, consisting of a cubic graph $G=(V,E)$ on n vertices $V=\{v_1,\ldots,v_n\}$. We can assume that $k>|E|/3$, or otherwise, the distinction between the two cases above is easy. We construct an instance $\mathcal{M}(\mathcal{I})$ of $\mathsf{Cardinality\text{-}Assort}$ 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, i.e., $N(v) = \{u : (u,v) \in E\}$, consisting of exactly 3 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 instance $\mathcal{M}(\mathcal{I})$ satisfies the following properties:

(a) $|U^*| \le k \quad \Rightarrow \quad R(S^*) \ge \dfrac{3}{4} + \dfrac{k}{4n}$,

(b) $|U^*| \ge (1+\alpha)k \quad \Rightarrow \quad R(S^*) \le \dfrac{3}{4} + \dfrac{k}{4n} - \dfrac{\alpha}{16}$,

where S^* is the optimal assortment for $\mathcal{M}(\mathcal{I})$. This implies that Cardinality-Assort cannot be approximated within 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. Since all prices are equal to 1, we can write the expected revenue of this set as

$$R(S) = \mathbb{P}(S \prec 0) = \sum_{i \in S} \lambda_i + \sum_{i \notin S} \lambda_i \mathbb{P}_i(S \prec 0) = \frac{k}{n} + \frac{1}{n} \sum_{i \notin S} \mathbb{P}_i(S \prec 0). \tag{7}$$

When starting at any state $i \notin S$, the Markov chain moves to 0 with probability 1/4 and gets absorbed. With probability 3/4, the Markov chain moves from i to one of the vertices in $N(i)$. Since 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 (7) 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 $N(i) \nsubseteq S$, 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 0. Consequently, for such items, $\mathbb{P}_i(S \prec 0) \le \frac{3}{4} - \frac{|N(i) \backslash S|}{16}$. Therefore,

$$\begin{aligned} R(S) &= \frac{k}{n} + \frac{1}{n} \sum_{i \notin S, N(i) \subseteq S} \frac{3}{4} + \frac{1}{n} \sum_{i \notin S, N(i) \nsubseteq S} \mathbb{P}_i(S \prec 0) \\ &\le \frac{3}{4} + \frac{k}{4n} - \frac{1}{16n} \sum_{i \notin S, N(i) \nsubseteq S} |N(i) \backslash S|. \end{aligned} \tag{8}$$

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

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

where the second inequality follows from $k>|E|/3$, and the last equality holds since $|E|=3n/2$, as G is cubic. By inequality (8), we have

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

Since 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^*. □

Proof of Theorem 2. Aouad et al. (2015) show that unconstrained assortment optimization over the distribution over permutations model is hard to approximate within factor $O(n^{1-\epsilon})$ for any fixed $\epsilon>0$ even for the case where the number of preference lists is equal to the number of items, i.e., $K=n$.

We consider an instance $\mathcal{I}$ of the assortment optimization problem over 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 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 $1\leq j\leq|L_i|$, we have a state (j,i) corresponding to the j-th 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:

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

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 1) over its corresponding states in decreasing order of preference, ending at the no-purchase option. This is illustrated in Figure 6. Finally, the arrival rates are defined by

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

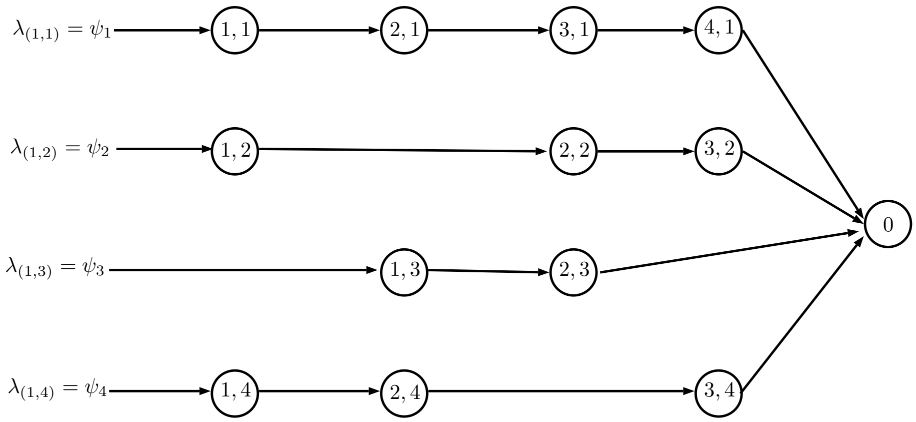
$\lambda_{(1,1)} = \psi_1$
$\lambda_{(1,2)} = \psi_2$
$\lambda_{(1,3)} = \psi_3$
$\lambda_{(1,4)} = \psi_4$
1,1
2,1
3,1
4,1
1,2
2,2
3,2
1,3
2,3
1,4
2,4
3,4
0

Figure 6 Sketch of our construction for an instance on 4 items, where $L_1 = (1 \succ 2 \succ 3 \succ 4)$, $L_2 = (1 \succ 3 \succ 4)$, $L_3 = (2 \succ 3)$, and $L_4 = (1 \succ 2 \succ 4)$. Note, for example, that the state $(2,2)$ corresponds to the second item of L_2, but actually corresponds to item 3.

where ψ_i is the probability of list L_i. With this construction, each row corresponds to a list, and each column correspond to an item.

In order 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 $i, \ell \in \{1, \dots, n\}$ such that $j \leq |L_i|, k \leq |L_\ell|$ and such that the j^{th} item in L_i is the k^{th} 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, where all other entries are 0. 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. Since 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: Proof of Lemma 2

This result is an immediate corollary of the following (more general) claim: Let S^g be the solution returned by Algorithm 1, and let S be any subset of states. Then,

$$R(S^g) \geq \frac{R(S)}{|S|}.$$

To prove this claim, let g be the first item selected by Algorithm 1, which necessarily exists as long as there is an item i with $p_i > 0$. Then, by definition of the greedy algorithm, we have $R(\{g\}) \geq R(\{i\})$ for every item $i \in S$. Therefore,

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

where the last inequality follows from the sublinearity of the revenue function (Lemma 9).

Appendix C: Proof of Lemma 4

Let S^{gu} be the set of states selected by Algorithm 2. Note that for every $i \in S^{gu}$, we have that $\mathbb{P}(i \prec S^{gu}_+ \backslash \{i\}) \geq \mathbb{P}(i \prec U^*_+ \backslash \{i\})$ since S^{gu} is a subset of U^*. Thus,

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

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

Appendix D: Proof of Lemma 6

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 the above identity clearly hold for the transition matrix updates. We have

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

Using the definition of the updated prices,

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

We can now combine A and C,

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

Putting everything together, we get

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

□

Appendix E: Application of Algorithm 3 to MNL

In the MNL model, we are given a collection of items, $1, \ldots, n$, along with the no-purchase option, which is denoted by 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, each item $i \in S$ is picked with probability

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

making the expected revenue

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

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 local ratio updates can be written as

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

Note that in the above update, the subtracted term is independent of i. Therefore, the ordering of the prices does not change after each update. Since we are picking the highest adjusted price item at each step, it follows that the optimal assortment is nested by price, i.e., consists of the top ℓ priced items, for some ℓ. This is a well known structural property that we recover here as a direct consequence of our algorithm. Moreover, the updated prices provide a criteria for when to stop adding items to the assortment.