Assortment Optimization under a Random Swap based Distribution over Permutations Model

Antoine Désir, Department of Industrial Engineering and Operations Research, Columbia University
Vineet Goyal, Department of Industrial Engineering and Operations Research, Columbia University
Danny Segev, Department of Statistics, University of Haifa, Haifa 31905, Israel

Assortment planning is an important problem that arises in many industries such as retailing and airlines where one of the key challenges is to identify the right model for the consumer preferences and substitution behavior. Distribution over preference lists or permutations is the most general framework for modeling preferences but is intractable in general [Aouad et al. 2015]. In this paper, we present a parsimonious distribution over permutations model that is specified by an initial preference list (that can be intuitively thought of as the mode) and any random preference list is generated by random swaps starting from the initial list. This model is motivated by practical applications where consumer preference are more or less similar over most items and differ in the relative order of only a few items. We present near-optimal algorithms for the assortment optimization problem under the random swap based distribution over permutations. Our algorithm is based on a surprising sparsity property about near-optimal assortments, namely, that there exist small-sized assortments that can be efficiently completed into near-optimal ones, crucially utilizing certain symmetries in the distribution over permutations. We also show that our results can be extended to more general settings where we have capacity constraints on the assortment and when the distribution over permutations is generated from a mixture of several initial lists. Therefore, our model provides a tractable framework for capturing consumer preferences under fairly general settings.

CCS Concepts: •**Applied computing**→ *Law, social and behavioral sciences; Economics;*

General Terms: Algorithms, Economics, Management

Additional Key Words and Phrases: choice modeling, revenue management, assortment optimization

1. INTRODUCTION

Assortment optimization problems arise widely in many practical applications such as retailing and online advertising. One of the key operational decisions faced by a retailer is to select a subset of items to offer from a universe of n substitutable items, trying to maximize the expected revenue. The demand of any item depends on the set of offered items due to the substitution behavior of consumers. This feature is captured by a choice model that specifies the probability that a random consumer selects any item of a given offer set. The goal in the assortment optimization problem is to identify a subset of items that maximizes the total expected revenue from a single random consumer.

Several parametric choice models have extensively been studied in diverse areas including marketing, transportation, economics, and operations management (see [Ben-Akiva and Lerman 1985; McFadden 1980; Wierenga 2008], for example). The multinomial logit (MNL) model is by far the most popular model in practice. It was intro-

V. Goyal is supported by NSF Grants CMMI 1201116 and CMMI 1351838.
Author's addresses: A. Désir and V. Goyal, Industrial Engineering and Operations Research, Columbia University, New York 10027; Email: {ad2918,vg2277}@columbia.edu; D. Segev, Department of Statistics, University of Haifa, Haifa 31905, Israel; Email: segevd@stat.haifa.ac.il

EC'16, July 24–28, 2016, Maastricht, The Netherlands. ACM 978-1-4503-3936-0/16/07.
http://dx.doi.org/10.1145/XXXXXXX.XXXXXXX

duced independently by [Luce 1959] and [Plackett 1975], where it was referred to as the Plackett-Luce model, later to be known as the multinomial logit model following the work of [McFadden 1978]. The popularity of this model arises from the tractability of estimation and the corresponding assortment optimization problem [Talluri and Van Ryzin 2004], The assortment optimization problem under the MNL choice model has also been shown to be tractable under additional constraints [Davis et al. 2013; Rusmevichientong et al. 2010]. However, some of the justifications for the MNL model are not reasonable for many applications, such as the Independence of Irrelevant Alternatives (IIA) property [Ben-Akiva and Lerman 1985]. To address the limitations of the MNL model, more general models including the nested logit model [Williams 1977] and the mixture of multinomial logit models have been considered in the literature. However, these models become more complex to estimate and optimize over. We refer the reader to excellent surveys [Kök et al. 2015; Lancaster 1990; Ramdas 2003] and to the references therein for an exhaustive review of different choice models in the literature.

One of the fundamental challenges in this area is to select the right model to capture consumer preferences. This is particularly difficult as the preferences are latent and we only observe the eventual choices made by the consumers. A general framework to model preferences is to consider a distribution over preference lists or permutations. Here, a preference list specifies an ordering of the items and a consumer with a particular preference list selects the first available item (possibly be no-purchase) on the list. This is the most general model and generalizes all parametric choice models including multinomial logit, nested logit and mixture of multinomial logit models. However, assortment optimization with respect to a general distribution over permutations is strongly inapproximable. In particular, [Aouad et al. 2015] show that it is NP-hard to compute a subset of items (assortment) whose expected revenue is within factor better than $O(n^{1-\epsilon})$ of optimal, for any fixed accuracy level $\epsilon > 0$. Furthermore, even specifying a general distribution over permutations may require us to explicitly specify probabilities for exponentially-many preference lists. Therefore, while there have been some positive results under specific structural assumptions [Aouad et al. 2015; Goyal et al. 2016; Honhon et al. 2012], in general, distribution over permutations is a highly intractable framework for choice modeling and assortment optimization. We would like to note that the hard instance in the reduction in [Aouad et al. 2015] is a distribution over permutations model with a sparse support on only n different preference lists. Such a model does not arise naturally in practice. In particular, if there is a non-zero probability for a certain preference list, σ, then typically close preference lists that differ in only a few items from σ also have non-zero probability in practice.

The above observation motivates us to consider a special class of distribution over permutations to model consumer preferences. In this paper we consider a distribution over permutations that is induced by *random swaps* from a central preference list (which we also refer to as the *prototype list*). In particular, a random list from this distribution can be sampled by performing random swaps starting from the initial prototype list. The number of swaps is also random according to a given distribution. A random swap operation consists of selecting a random pair of items in the current list and swapping their positions. We consider two types of random swaps: $i)$ swapping an arbitrary pair of items where a pair is picked uniformly at random out of all $\binom{n+1}{2}$ possible pairs, and $ii)$ swapping an adjacent pair of items where a pair is picked out of n pairs of adjacent items. We also allow the initial preference list to contain only a subset of the items prior to the no-purchase option. In this case, we assume a random order on the items not included in the list to complete it into a permutation for the random swap process. The precise mathematical definition of this model and its dynamics are given in Section 1.2.

This model is motivated by practical applications where consumers preferences generally have many common items appear according to the same relative order and differ only in a small number of items. The prototype list used to generate a random preference list can intuitively be thought of as the mode of the distribution implied by the random swap model. If the support of the distribution over the number of swaps is sufficiently large, the distribution over permutations has non-zero probability on all preference lists. Therefore, this random swap based distribution gives a "smooth" distribution over all permutations. As a consequence, our model avoids a sparse distribution with non-zero probability on isolated permutations which is the structure of the hard instance instance in [Aouad et al. 2015]. Our distribution over permutations model is described by at most n parameters specifying the prototype list and a distribution over the number of random swaps. This is quite analogous to the multinomial logit model which is also specified by n utility parameters and the distribution over the random component of the utility (which for the case of MNL is standard Gumbel [Luce 1959]). Therefore, the random swap based distribution over permutations provides a parsimonious framework for modeling preferences directly as distribution over rankings and in some sense, it is quite analogous to the MNL model which is a parsimonious random utility based model.

We would also like to note that our random swap based model is quite analogous to the Mallow's distribution that was introduced in [Mallows 1957]. This is another parsimonious model widely used for modeling distributions over preferences [Doignon et al. 2004; Lu and Boutilier 2011]. As in our model, the Mallows model is specified by an initial preference list, L, with probability p_0, and the probability of any other preference list L' decreases exponentially with the Kendall-tau distance between L and L' (which is the number of disagreements on pairwise comparisons between L and L').

1.1. Our results and techniques

In this paper, we consider the assortment optimization problem over the distribution over permutations model described above, where the goal is to compute a subset of items to offer that maximizes the expected revenue from a single random conusmer, i.e.,

$$\max_{S\subseteq[n]} \sum_{j\in S} \Pr\left[j \succ S\right] \cdot r_j,$$

where $\Pr\left[j \succ S\right]$ denotes the probability that item j is the most preferred among all items in S, also referred to as the choice probability of $j \in S$. Note that the choice probability includes a sum over possibly exponentially many permutations and can not be computed in closed form. In this paper, we present a polynomial time approximation scheme (PTAS) for the above assortment optimization problem over this random swap based distribution over permutations model. We also show that our results can be extended to more general settings where we have capacity constraints on the assortment and when the distribution over permutations is a mixture of random swap based distributions. Therefore, we show that the random swap based distribution over permutations provides a tractable framework for modeling consumer preferences under fairly general settings. Our main contributions are summarized below.

Constant factor approximation We first present a simple algorithm that gives a $1/3$-approximation for random swap based distribution over permutations model for both adjacent and arbitrary swaps. In other words, our algorithm computes an assortment with expected revenue at least $1/3$ times the optimal. The algorithm is based on the structural properties about the relative orderings of items in the random swap based distribution over permutation models. In particular, we show that the best among $O(n)$ assortments gives a $1/3$-approximation for the problem.

Computing near-optimal assortments. We devise a polynomial-time approximation scheme (PTAS) for the assortment optimization problem under the random swap model for both types of random swaps. In other words, for any accuracy level $\epsilon > 0$, we compute an assortment with expected revenue at least $(1 - \epsilon)$ times the optimal. The running time of the algorithm is $O(n^{O(1/\epsilon)})$ that depends exponentially on $1/\epsilon$. Our algorithm is based on establishing a surprising sparsity property about near-optimal assortments, namely, that there exist small-sized assortments that can be efficiently completed into near-optimal ones, crucially utilizing certain symmetries in the distribution over permutations. In particular, we show that among the set of items that do not exist in the prototype list, it is optimal to select a revenue ordered set of items, i.e., the top-k highest revenue ones (for some k).

For the case of arbitrary swaps, we present an improved PTAS, based on a approximate dynamic program. In particular, we give an algorithm whose running time depends polynomially on n and $O((1/\epsilon)^{O((1/\epsilon)\cdot\log(1/\epsilon))})$. A result of this nature is also referred to as an EPTAS (see [Baker 1994; Epstein and Levin 2014; Fomin et al. 2011; Jansen 2010]). In particular, we present a dynamic programming algorithm, where in each step the probabilities of certain events (describing relative orders in random permutations) have to be computed. While computing these probabilities exactly leads to difficult counting problems, we give an efficient sampling algorithm to estimate these probabilities and show that using these estimates from simulation for the dynamic program (instead of the real probabilities) still leads to an efficient PTAS, with high probability.

Extensions to more general distributions and constrained assortment optimization. We show that our PTAS for the assortment optimization problem can be extended to significantly more general settings. In particular, we consider the following two generalizations.

Mixture of prototype lists. We consider a more general distribution over permutations model which is given by a mixture of random swap based distributions. This essentially reduces to considering a mixture of initial prototype lists. This is a significantly more general model that allows us to capture heterogeneity in consumer preferences where each prototype list corresponds to a particular consumer type. We would like to note that we require all prototype lists to contain the same set of items, even though their internal order within each list could be arbitrary (If we relax this assumption, the model can be used to capture an arbitrary distribution over preference lists and therefore becomes at least as hard to approximate as the independent set problem [Aouad et al. 2015]). We show that our algorithms (both PTAS and EPTAS) can be leveraged to obtain near-optimal solutions for this general mixture model. The running time of our PTAS does not depend on the number of initial prototype lists in the mixture model. However, the running time of our EPTAS for the arbitrary swap model depends exponentially on the number of lists, and we would require the number of lists to be constant for the EPTAS.

Capacity constraints on assortments. In a capacity constrained assortment optimization problem, each item has an associated weight and there is an upper bound on the total weight of the final assortment. Such capacity constraints arise naturally in many applications to model a budget restriction or a display space limitation and have been studied for many choice models (for example, [Gallego and Topaloglu 2014; Rusmevichientong et al. 2010]). Interestingly, we show that we can adapt our algorithms using several new ideas to obtain near-optimal approximations for capacity-constrained assortment optimization over random swap models.

Outline. The rest of the paper is organized as follows. In Section 3, we present the constant factor approximation for the assortment optimization problem. We present the PTAS and EPTAS in Sections 4 and 5. In Section 6, we present the extensions to the case of mixture of constant number of random swap based distribution over permutations, and we present the constrained assortment optimization over these models in Section 7.

1.2. Model

In what follows, we make use of $1, \ldots, n$ to denote the underlying set of items, with the convention that $\mathcal{X}$ stands for the no-purchase option. For $i \in [n]$, item i is associated with a revenue r_i. In this paper, we consider a special class of distributions over permutations generated by *random swaps*.

The prototype list. We model the preference list of a random consumer from the underlying population by picking an initial list L_0 (which we will refer to as *prototype list*) on which a random number of *random swaps* are performed. This prototype list can be intuitively thought of as the mode of the distribution implied by the random swap model. Moreover, we allow the prototype list to contain only a subset of the items. In this case, we assume a random order on the items not included in the list to complete it into a permutation for the random swap process. More precisely, the items are divided into a set of *desirable* items $\mathcal{D}$ and a set of *undesirable* items $\mathcal{U}$. Without loss of generality, let $\mathcal{D} = \{1, \ldots, d\}$. With this notation, the prototype list L_0 is given by

$$L_0 = [1 \succ 2 \succ \cdots \succ d \succ \mathcal{X} \succ \mathcal{U}_0],$$

where $\mathcal{U}_0$ is a random permutation of $\mathcal{U}$, drawn uniformly among the $(n-d)!$ possible permutations. Note that due to the randomness of $\mathcal{U}_0$, the prototype list L_0 is a random preference list by itself.

Random swaps. We then generate a random preference list by sequentially applying a random number of random swaps. The number of swaps is given by a random variable X, whose distribution is assumed to be known in advance. A random swap operation selects a random pair of items in the current list and swaps their positions. In particular, the value of X first realizes, and then, we sequentially perform X random swaps.

We consider two types of random swaps: (1) swapping an arbitrary pair of items, and (2) swapping an adjacent pair of items. In the arbitrary swap model, each of the $\binom{n+1}{2}$ pairs (i, j) with $i \neq j$ is picked with equal probability. In the adjacent swap model, each of the n pairs $(i, i+1)$ is picked with equal probability. We do not put any restriction on the number of swaps, and therefore, we can achieve any permutation starting from L_0 even in the adjacent swap model. Note that the ordering of the desired items $\mathcal{D}$ as well as the distribution of X completely defines the distribution over permutations.

Objective. We consider the assortment optimization problem under this new choice model. As previously mentioned, each item $i \in [n]$ is given a revenue r_i. The goal is to select the subset of items that maximizes the expected revenue. When a subset $S \subseteq [n]$ is offered, a consumer first picks a random permutation according to the process defined earlier, and then picks the highest-ranked offered item on his list which is also preferred to the no-purchase option. For a fixed list L, let $\mathcal{R}(L, S)$ be the revenue generated by the subset of items S. In addition, let $\mathcal{R}(S)$ be the expected revenue generated by the subset S, where the expectation is taken over the distribution over permutation described earlier. The assortment optimization problem can be written as

$$\max_{S \subseteq [n]} \mathcal{R}(S) = \max_{S \subseteq [n]} \mathbb{E}[\mathcal{R}(L, S)].$$

Computing the expected revenue. It is worth mentioning that, for a given subset S, we do not have any closed form expression to compute $\mathcal{R}(S)$. However, we show in Appendix A.1 how to efficiently estimate this quantity by sampling. Consequently, we assume in the remainder of the paper, that we have access to an oracle to computes $\mathcal{R}(S)$ for any given subset S.

2. RELATIVE ORDERS IN RANDOM PERMUTATIONS

This section introduces a number of structural properties regarding relative orders in random permutations, before we move to the more algorithmic part. In particular, we show that if an item i is preferred to an item j in the prototype list, then item i is preferred to item j in a random list that results from our swapping process with probability at least $1/2$, irrespective of the type of swap.

Notation. We denote by $r^k(\cdot)$ the random rank of any item after k random swaps. Also, $\succ_k$ denotes the random ordering after k swaps.

2.1. Probabilistic claims

We first show that for any pair of items (i, j) such that i is preferred to j in the prototype list L_0, item i is preferred to item j with probability at least $1/2$ after any number of random swaps for both arbitrary and adjacent swaps. Note that $\Pr[i \succ_0 j] = 1$ and $\lim_{k\to\infty} \Pr[i \succ_k j] = 1/2$ (since in the limit of infinite swaps the distribution tends to a uniform distribution over permutations).

CLAIM 2.1. *Let i and j be a pair of items such that $i \succ_0 j$. Then, for any integer k,*

$$\Pr\left[i \succ_k j\right] \geq \frac{1}{2}.$$

We also extend this result to a tuple of items $(i_1, \ldots, i_m)$. More precisely, we show that if item i_1 is the most preferred item of $(i_1, \ldots, i_m)$ in the prototype list, then item i_1 is the most preferred after any number of random adjacent swaps with probability at least $1/m$. Again, note that $\Pr[i_1 \succ_0 i_j, \forall j = 2, \ldots, m] = 1$ and $\lim_{k\to\infty} \Pr[i_1 \succ_0 i_j, \forall j = 2, \ldots, m] = 1/m$ when $k \to \infty$.

CLAIM 2.2. *Let $i_1, \ldots, i_m$ be a sequence of items such that $i_1 \succ_0 i_2 \succ_0 \cdots \succ_0 i_m$. Then, for any integer k,*

$$\Pr\left[i_1 \succ_k i_j, \forall j \geq 2\right] \geq \frac{1}{m}.$$

2.2. Proofs

We are going to prove Claim 2.1 separately for the two types of swaps. However, we first need an intermediate result.

CLAIM 2.3. *Consider the arbitrary swap model. For any integer k, there exists a function f^k such that for every pair of items i and j*

$$\Pr\left[i \succ_k j | r^0(i) = \ell, r^0(j) = m\right] = f^k(\ell - m).$$

PROOF. Consider ℓ' and m' such that $\ell - m = \ell' - m'$. We want to show that

$$\Pr\left[i \succ_k j | r^0(i) = \ell, r^0(j) = m\right] = \Pr\left[i \succ_k j | r^0(i) = \ell', r^0(j) = m'\right].$$

Table I. Coupling between swaps in L_1 and L_2.

	Swap in L_1	Swap in L_2
(a)	i is swapped to position k with $1 \leq k \leq n - m'$, $k \neq i$ and $k \neq j$	i is swapped to position $k + (\ell' - \ell)$
(b)	i is swapped to position k with $n - m' < k \leq n$	j is swapped to position $(k - m') - \ell'$
(c)	j is swapped to position k with $1 \leq k \leq n - m'$, $k \neq i$ and $k \neq j$	i is swapped to position $k + (\ell' - \ell)$
(d)	j is swapped to position k with $n - m' < k \leq n$	i is swapped to position $(k - m') - \ell'$

We do a single step analysis and show that after a single swap the distribution over the distance between i and j is the same when starting with L_1 where $(r^0(i), r^0(j)) = (\ell, m)$ and starting with L_2 where $(r^0(i), r^0(j)) = (\ell', m')$. The result then follows by induction.

Note that when a swap does not involve i and j or involves both, the distance between i and j remains the same in L_1 and L_2. We now couple the remaining possible swaps such that for every coupled swaps, the distance between i and j remains unchanged. Without loss of generality, assume that $\ell < \ell'$ and $\ell < m$. Consider the coupling described in Table 1 (see also Figures 1 and 2). For case (a),

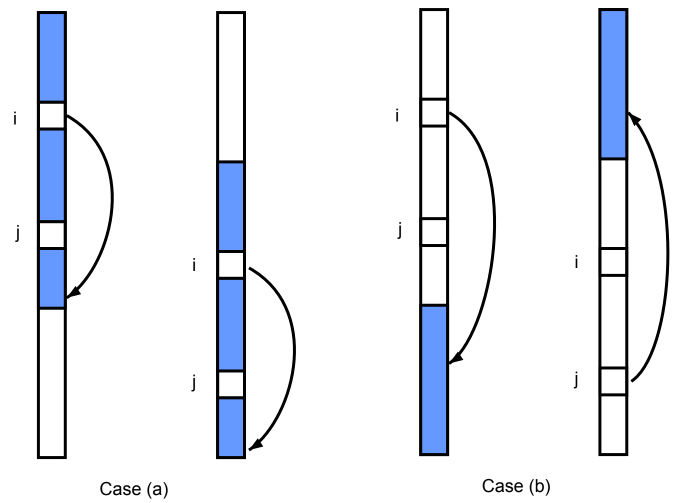

Fig. 1. Coupling when i is swapped in L_1

$$r_1^1(i) - r_1^1(j) = k - m,$$
$$r_2^1(i) - r_1^2(j) = k + \ell' - \ell - m' = k - \ell + (\ell' - m') = k - \ell + (\ell - m) = k - m$$

Similarly, for case (b),

$$r_1^1(i) - r_1^1(j) = k - m,$$
$$r_2^1(i) - r_1^2(j) = \ell + \ell' - (k - m') = k + \ell - (\ell' - m') = k + \ell - (\ell - m) = k - m$$

Cases (c) and (d) are similar. □

We now present a separate proof of Claim 2.1 for each type of swap.

PROOF OF CLAIM 2.1 (ARBITRARY SWAPS). Without loss of generality, we assume that $r^0(i) = i$ and $r^0(j) = m$. We condition on the first swap. Consider a swap such that $j \succ_1 i$.

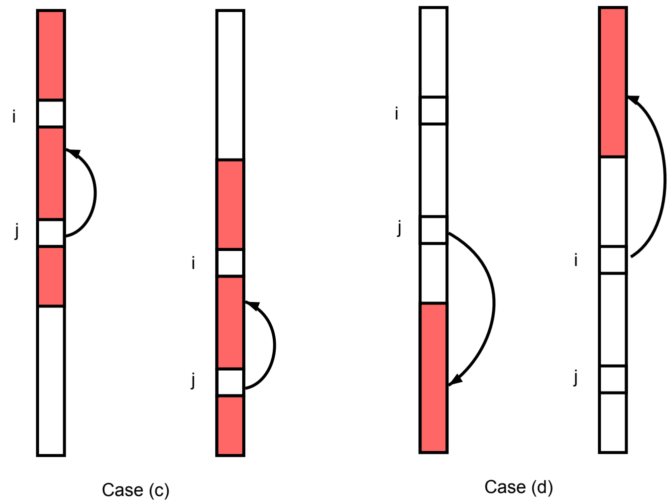

Fig. 2. Coupling when j is swapped in L_1

Case a. If j is swapped but not i, it implies that j is swapped to a position $\ell < i$. By Claim 2.3, we have

$$\begin{aligned}\Pr\left[i \succ_{k-1} j | r^0(j) = \ell, r^0(i) = i\right] &= \Pr\left[j \succ_{k-1} i | r^0(j) = m, r^0(i) = m + \ell - i\right] \\ &= 1 - \Pr\left[i \succ_{k-1} j | r^0(j) = m, r^0(i) = m + \ell - i\right].\end{aligned}$$

Let s_1 be the first swap. We have for $\ell < i$,

$$\begin{aligned}\Pr\left[i \succ_k j | s_1 \in \{(\ell, m), (i, m + \ell - i)\}\right] =& \frac{1}{2} \cdot \Pr\left[i \succ_{k-1} j | r^0(j) = \ell, r^0(i) = i\right] \\ &+ \frac{1}{2} \cdot \Pr\left[i \succ_{k-1} j | r^0(j) = m, r^0(i) = m + \ell - i\right] \\ =& \frac{1}{2}.\end{aligned}$$

Case b. The second possibility for $j \succ_1 i$ is that i is swapped but not j. This implies that i is swapped to a position $\ell > m$. In that case, we have

$$\Pr\left[i \succ_k j | s_1 \in \{(i, \ell), (m, i + \ell - m)\}\right] = \frac{1}{2}.$$

Case c. Finally, i and j can be swapped together. In that case, if there exists two other items ℓ and m, then by coupling the swap involving both i and j with the swap (ℓ, m), we have

$$\Pr\left[i \succ_k j | \text{the first swap is } (i, j) \text{ or } (\ell, m)\right] = \frac{1}{2}.$$

Putting together the three cases, let E be the set of swaps described in case (a), (b) and (c). We have

$$\Pr\left[i \succ_k j\right] = \frac{1}{2} \cdot \Pr\left[E\right] + \Pr\left[i \succ_{k-1} j | i \succ_0 j\right] \cdot (1 - \Pr\left[E\right]).$$

The proof is then completed by induction. $\square$

We present the proof of Claim 2.1 for adjacent swaps in Appendix A.2.

PROOF OF CLAIM 2.2 (ARBITRARY SWAPS). Note that the couplings introduced in the above proof of Claim 2.1 only involve a pair of items which are different than the items we are trying to compare. Consequently, for every $j' \geq 2$, the same proof implies that

$$\Pr\left[i_1 \succ_k i_j, \forall j \geq 2\right] \geq \Pr\left[i_{j'} \succ_k i_j, \forall j \neq j'\right]$$

which in turn implies the desired result. □

The proof for adjacent swaps is similar and therefore omitted.

3. WARM-UP: A CONSTANT FACTOR APPROXIMATION ALGORITHM

In order to introduce our main technical ideas incrementally, we first present a simple algorithm that guarantees a constant factor approximation. The algorithm returns the highest revenue item in $\mathcal{D}$ or the best nested assortment of $\mathcal{U}$. The correctness of our algorithm relies on the probabilistic claims of Section 2 as well as on additional observations that are presented in the sequel. Note that, since the algorithm returns the highest revenue item in $\mathcal{D}$, it is natural to conjecture that a nested assortment with sufficiently many items could result in a PTAS. However, we demonstrate that for both type of swaps, this conjecture is not true. In particular, we present families of instances for which any nested assortment attains only a constant fraction of the optimal revenue.

3.1. A $1/3$-approximation algorithm

Before presenting the algorithm, we establish an additional structural result about the items of $\mathcal{U}$ that are picked by an optimal solution. More precisely, we show that there always exists an optimal solution that picks the k highest revenue items of $\mathcal{U}$ for some k. As mentioned in Section 1.2, the items in $\mathcal{U}$ are randomly ordered in the prototype list. We exploit this symmetry to prove the result, which does not depend on the type of swaps.

CLAIM 3.1. *For any subset of desired items $D \subseteq \mathcal{D}$, there exists a revenue ordered subset of $\mathcal{U}$ such that $\mathcal{R}(D \cup \mathcal{S})$ is maximized over all $S \subseteq \mathcal{U}$.*

PROOF. For a fixed $D \subseteq \mathcal{D}$, let $S^* \subseteq \mathcal{U}$ be an optimal subset that maximizes $\mathcal{R}(D \cup \mathcal{S})$ over all $S \subseteq \mathcal{U}$. Suppose there exists a pair of items $i, j \in \mathcal{U}$ such that $r_i < r_j$ and $i \in S^*$ but $j \notin S^*$. Consider the bijection $\sigma \in \mathcal{S}_{n+1}$, where $\mathcal{S}_{n+1}$ is the set of permutation of $\{\mathcal{X}, 1, \ldots, n\}$, such that

$$\sigma(k) = \begin{cases} i & \text{if } k = j \\ j & \text{if } k = i \\ k & \text{otherwise} \end{cases} .$$

For any prototype list L, we construct another coupled prototype list L^σ. If $i_1 \succ \cdots \succ i_{|\mathcal{U}|}$ is the random order of $\mathcal{U}$ in L, then $\sigma(i_1) \succ \cdots \succ \sigma(i_{|\mathcal{U}|})$ is the random order of $\mathcal{U}$ in L^σ. This coupling exchanges the position of i and j in the prototype list. Therefore, for every L,

$$\mathcal{R}(L, S^*) \leq \mathcal{R}(L^\sigma, S^* \cup \{j\} \backslash \{i\}).$$

Since σ is a bijection, $\mathbb{E}\left[\mathcal{R}(L^\sigma, S)\right] = \mathbb{E}\left[\mathcal{R}(L, S)\right]$ for any subset S. Taking expectation in the previous inequality yields the desired result. □

The above result tells us how to complete any subset of $\mathcal{D}$ in an optimal way. We are now ready to describe our constant factor approximation algorithm.

Description of the algorithm. The algorithm considers two assortments S_1 and S_2, and returns the one with the highest expected revenue. We now describe the two candidate assortments. First, S_1 is the assortment consisting of the highest revenue item in $\mathcal{D}$. Second, S_2 is the best revenue-ordered subset of $\mathcal{U}$. More precisely, let $U_k \subseteq \mathcal{U}$ be the set of k highest revenue items in $\mathcal{U}$. We have $S_2 = \arg\max\{\mathcal{R}(U_k) : 0 \leq k \leq |\mathcal{U}|\}$. Note that S_1 consists of a single item from $\mathcal{D}$, whereas S_2 consists only of items from $\mathcal{U}$. Algorithm 1 details the algorithm.

ALGORITHM 1: Constant Factor Algorithm

Let $S_1 = \arg\max\{r_i : i \in \mathcal{D}\}$.
Let $S_2 = \arg\max\{\mathcal{R}(U_k) : 0 \leq k \leq |\mathcal{U}|\}$, where $\mathcal{U}_k$ consists of the highest k revenue items in $\mathcal{U}$.
return $\arg\max\{\mathcal{R}(S_1), \mathcal{R}(S_2)\}$.

This construction provides a constant factor approximation for both types of swaps. More precisely, we have the following result.

THEOREM 3.1. *Algorithm 1 guarantees a 1/3-approximation for the assortment optimization problem over the random swap based distribution over permutations model.*

PROOF. Let S^* be the optimal assortment. Let $S^*_U = S^* \cap \mathcal{U}$, and $S^*_D = S^* \cap \mathcal{D}$. The revenue function is sublinear and we have

$$\mathcal{R}(S^*) \leq \mathcal{R}(S^*_{\mathcal{D}}) + \mathcal{R}(S^*_{\mathcal{U}}).$$

We show that S_1 is a good solution compared to S^*_D and S_2 is a good solution compared to S^*_U.

Let i^* be the highest revenue item in $\mathcal{D}$, i.e., $i^* = \text{argmax}\{r_i : i \in \mathcal{D}\}$. By definition, $S_1 = \{i^*\}$ and $r_{i^*} \geq \mathcal{R}(S^*_{\mathcal{D}})$. Indeed, for any set of items S, the expected revenue $\mathcal{R}(S)$ is a convex combination of the revenues of the items in S (including $\mathcal{X}$). From Claim 2.1, there is a probability of at least $1/2$ that item i^* is preferred to the no-purchase option after any number of swaps. Therefore, we have

$$\mathcal{R}(S_1) = \mathcal{R}(\{i^*\}) = \Pr[i \succ \mathcal{X}] \cdot r_{i^*} \geq \frac{1}{2} \cdot \mathcal{R}(S^*_D).$$

In addition, S_2 is the best nested solution in $\mathcal{U}$. Therefore, from Claim 3.1, S_2 also maximizes the expected revenue among all assortment $S \subseteq \mathcal{U}$. Consequently,

$$\mathcal{R}(S_2) \geq \mathcal{R}(S^*_U).$$

Putting the two parts together, it follows that we obtain an expected revenue of

$$\max\{\mathcal{R}(S_1), \mathcal{R}(S_2)\} \geq \frac{2}{3} \cdot \mathcal{R}(S_1) + \frac{1}{3} \cdot \mathcal{R}(S_2) \geq \frac{1}{3} \cdot \mathcal{R}(S^*_D) + \frac{1}{3} \cdot \mathcal{R}(S^*_U) \geq \frac{1}{3} \cdot \mathcal{R}(S^*).$$

Note that since Claims 2.1 and 3.1 are true for both type of swaps, this proof applies to both types as well. □

Note Algorithm 1 requires computing the expected revenue of different candidate assortments. These quantities can be approximated efficiently using sampling (see Appendix A.1). Our algorithm either picks the highest revenue items in $\mathcal{D}$ or the k highest revenue item in $\mathcal{U}$ for some k. Therefore, one could conjecture that nested assortments in $\mathcal{D} \cup \mathcal{U}$ could be good candidates for an optimal solution. In Section 3.2, we show that this intuition is not true for both type of swaps.

3.2. Bad Example

In this section, we exhibit an example for the case of arbitrary swaps where a nested solution is not optimal. More precisely, no optimal solution picks the k highest revenue items for any k. Moreover, we prove that there is a constant gap in optimality. Therefore, this approach cannot provide a PTAS.

The construction. Consider the following example with $2n+2$ items and $d = n+2$. The revenues are as followed

$$r_i = \begin{cases} 1+\epsilon & \text{if } i = 1 \\ 0 & \text{if } 2 \le i \le n+1 \\ 1 & \text{if } i = n+2 \\ 2/(1-\alpha) & \text{if } i > n+2 \end{cases},$$

where $\epsilon > 0$ and $0 < \alpha < 1$. The prototype list is ordered as follows (here the revenues are used instead of the item numbering):

$$L_0 = \Big[1+\epsilon \succ \underbrace{0 \succ \cdots \succ 0}_{n \text{ items}} \succ 1 \succ \mathcal{X} \succ \underbrace{\frac{2}{1-\alpha} \succ \cdots \succ \frac{2}{1-\alpha}}_{n \text{ items}}\Big].$$

Note that since all the items in $\mathcal{U}$ have the same revenue, the prototype list is deterministic. Additionally, we consider the following distribution on the number of swaps:

$$X = \begin{cases} 0 & \text{with probability } \alpha \\ 1 & \text{otherwise} \end{cases}.$$

For this example, we show in Appendix A.3 that any nested assortment attains only a constant fraction of the optimal revenue. We also provide an example for the case of adjacent swaps in Appendix A.4.

4. A PTAS FOR THE ASSORTMENT OPTIMIZATION PROBLEM

In this section, we present a polynomial time approximation scheme (PTAS) for the assortment optimization problem under the swap model, which works for both types of swaps. Our algorithm is based on establishing a surprising sparsity property, proving the existence of small-sized assortments that can be efficiently completed into near-optimal ones, crucially utilizing certain symmetries in the distribution over permutations. In fact, our constant factor algorithm (see Section 3.1) also uses a small-sized assortment of the items in $\mathcal{D}$, by only considering the highest revenue one. Here the set of items that we consider are not necessarily the highest revenue items in general, but rather those picked by the optimal assortment.

Description of the algorithm. Let S^* be the optimal assortment and $S^*_U = S^* \cup \mathcal{U}$. From Claim 3.1, we know that S^*_U is nested by revenue. Therefore, by guessing its cardinality, we can assume that S^*_U is known.

Let $K = 1/\epsilon$, where without loss of generality, assume that K takes an integer value. We enumerate all possible subsets of $\mathcal{D}$ of size $k = 0, \ldots, K$, and construct a candidate assortment for each subset as follows. For each subset S_D, the candidate assortment is just $S^*_U \cup S_D$, i.e., we take the union of S^*_U and S_D. The algorithm returns the best candidate assortment. Algorithm 2 describes the procedure.

THEOREM 4.1. *Algorithm 2 is a PTAS for the assortment optimization problem under the swap model (with both arbitrary and adjacent swaps).*

PROOF. We first argue the correctness of the algorithm, i.e., that the assortment returned is a $(1-\epsilon)$-optimal solution. Again, let S^* be the optimal assortment, $S^*_U =$

ALGORITHM 2: PTAS

Let $\mathcal{A} = \emptyset$.
for all $S_{\mathcal{D}} \subseteq \mathcal{D}$ such that $|S_{\mathcal{D}}| \leq K$ **do**
  for all $\ell = 0, \ldots, |\mathcal{U}|$ **do**
    Let $\mathcal{A} = \{\mathcal{A}, S_{\mathcal{D}} \cup U_\ell\}$, where U_ℓ consists of the ℓ highest revenue items in $\mathcal{U}$.
  end
end
return $\arg\max\{\mathcal{R}(S), S \in \mathcal{A}\}$.

$S^* \cap \mathcal{U}$, and $S^*_D = S^* \cap \mathcal{D}$. Note that if $|S^*_D| < K$, then S^* is one of the candidate assortments we examine, and therefore the algorithm returns the optimal solution. We therefore assume that $S^*_D \geq K$. In this case, let S_D consist of the K highest revenue items of S^*_D. Consider the assortment $S_\epsilon = S_D \cup \mathcal{A}$, where $\mathcal{A}$ consists of all the items of S^*_U such that $r_i \geq \min\{r_j : j \in S_D\}$. Since $|S_D| = K$ and S^*_U is nested, note that S_ϵ is among the candidate assortment constructed by the algorithm. We show that S_ϵ is $(1-\epsilon)$-optimal using a sample-path analysis. In particular, let L be a fixed preference list. We consider two cases.

Case 1. We first assume that $\mathcal{X} \succ i$ in the preference list L for all $i \in S_\epsilon$. In this case, $\mathcal{R}(L, S_\epsilon) = 0$. On the other hand, offering a single item i is always a feasible solution. Therefore, by Claim 2.1, for all $i \in \mathcal{D}$,

$$\mathcal{R}(S^*) \geq \mathcal{R}(\{i\}) = \Pr[i \succ \mathcal{X}] \cdot r_i \geq \frac{r_i}{2}.$$

Moreover, note that all items in $S^*_U \backslash A$ have revenue smaller or equal to some item in $\mathcal{D}$. Therefore, any item in $S^* \backslash S_\epsilon$ has revenue less or equal than $2\mathcal{R}(S^*)$. This implies that $\mathcal{R}(L, S^*) \leq 2\mathcal{R}(S^*)$.

Case 2. In this case, we assume that in the permutation L, there exists an item $i \in S_\epsilon$ such that $i \succ \mathcal{X}$. We show that $\mathcal{R}(L, S_\epsilon) \geq \mathcal{R}(L, S^*)$. Indeed, suppose the chosen item i^* is in $S^*_D \backslash S_D$. Since S_D contains the K highest revenue items of S^*_D, it must be that $r_{i^*} \leq r_i$ for all $i \in S_\epsilon$. Therefore, $\mathcal{R}(L, S_\epsilon) \geq \mathcal{R}(L, S^*)$. On the other hand, if the chosen item i^* is in $S^*_U \backslash A$, it must be that $r_{i^*} \leq r_i$ for all $i \in S_\epsilon$. Consequently, we also have $\mathcal{R}(L, S_\epsilon) \geq \mathcal{R}(L, S^*)$ in that case.

We now combine the two cases. For case 1 to happen, note that $\mathcal{X}$ has to be preferred to all items from S_D. From Claim 2.2, this event occurs with probability at most $1/|S_D| = 1/K = \epsilon$. Consequently,

$$\begin{aligned}
\mathcal{R}(S^*) - \mathcal{R}(S_\epsilon) =& \Pr[\text{Case 1}] \cdot \underbrace{\mathbb{E}\left[\mathcal{R}(S^*) - \mathcal{R}(S_\epsilon) | \text{Case 1}\right]}_{\leq 0} \\
&+ \Pr[\text{Case 2}] \cdot \underbrace{\mathbb{E}\left[\mathcal{R}(S^*) - \mathcal{R}(S_\epsilon) | \text{Case 2}\right]}_{=\mathbb{E}[\mathcal{R}(S^*)|\text{Case 2}]} \\
\leq& \epsilon \cdot \mathbb{E}\left[\mathcal{R}(S^*) | \text{Case 2}\right] \\
\leq& 2\epsilon \cdot \mathcal{R}(S^*).
\end{aligned}$$

From a running time perspective, for each subset $S_U \subseteq U$, the number of candidate assortment is at most $|\mathcal{D}|^K \leq n^{1/\epsilon}$. Since we only consider nested assortment of $\mathcal{U}$, we have to consider at most n subsets of $\mathcal{U}$, and therefore the overall running time of the algorithm is $O(n^{O(1/\epsilon)})$. □

5. EPTAS FOR ARBITRARY SWAPS

For the case of arbitrary swaps, we present an improved PTAS, based on approximate dynamic programming. In particular, we devise an algorithm whose running time for any accuracy level $\epsilon > 0$ depends polynomially on n and $O((1/\epsilon)^{O((1/\epsilon)\cdot\log(1/\epsilon))})$. This is referred to as an efficient polynomial time approximation scheme (EPTAS).

Preprocessing. We first construct a modified instance of the problem whose optimal expected revenue is at least $(1-2\epsilon)\cdot\mathcal{R}(S^*)$, where S^* is the optimal solution to our original instance. The modified instance sets the revenue of inexpensive items in $\mathcal{D}$ to 0. More precisely, for every item $i \in \mathcal{D}$ such that $r_i \leq \epsilon r^*$, where $r^* = \max\{r_j : j \in \mathcal{D}\}$, its revenue in the modified instance is 0. Let $\tilde{S}^*$ be the optimal solution of the modified instance.

CLAIM 5.1. $\mathcal{R}(\tilde{S}^*) \geq (1-2\epsilon)\cdot\mathcal{R}(S^*)$.

PROOF. For all $i \in S^*$, let E_i be the event that item i is the first out of S^* and $\mathcal{X}$. Also, let B be the set of items whose revenue was decreased to 0 in the modified instance. We have

$$\begin{aligned}\mathcal{R}(S^*) - \mathcal{R}(\tilde{S}^*) &\leq \sum_{i\in B} \Pr[E_i]\, r_i \\ &\leq \left(\sum_{i\in B} \Pr[E_i]\right)\cdot \epsilon r^* \\ &\leq \epsilon r^* \\ &\leq 2\epsilon\cdot\mathcal{R}(S^*),\end{aligned}$$

where the last inequality uses an argument similar to the proof of Theorem 4.1. □

Description of the algorithm. Using the preprocessing step described above, we assume that all non-zero revenues of items in $\mathcal{D}$ are within the interval $I = [\epsilon r^*, r^*]$. Moreover, as in the PTAS (see Section 4), we assume that some nested $S_U \subseteq \mathcal{U}$ is known in advance. The algorithm starts by geometrically partitioning the interval I in powers of $1+\epsilon$. In particular, the first subinterval is $[\epsilon r^*, \epsilon(1+\epsilon)r^*]$, the second is $[\epsilon(1+\epsilon)r^*, \epsilon(1+\epsilon)^2 r^*]$, and so on. The number of subintervals is therefore $M = O((1/\epsilon)\cdot\log(1/\epsilon))$. This partition of the interval I induces a partition of the items of $\mathcal{D}$ according to their revenue. We denote by $C_1,\dots,C_M$ these induced classes of items.

For each class, let K_i be the number of items in $S^* \cap C_i$, where S^* is the optimal solution. For each class, we guess this number of item K_i, i.e., we enumerate all possible combinations of $(K_1,\dots,K_M)$ and run the algorithm for each M-tuple. Moreover, by the analysis of the PTAS in Section 4, we know that in order to obtain a $(1-\epsilon)$-optimal solution, it suffices to consider subsets of $\mathcal{D}$ of size at most $K = 1/\epsilon$. Consequently, the number of guesses for $(K_1,\dots,K_M)$ is $O((1/\epsilon)^M)$. For the remainder of this section, we assume that $(K_1,\dots,K_M)$ is already known.

We now solve a dynamic program. Recall that items from some nested $S_U \subseteq \mathcal{U}$ are assumed to be picked. In order to write down the dynamic program, we need more notation. For $k_m \in K_m$ for $m = 1,\dots,M$, let $\kappa = (k_1,\dots,k_M)$ and $\mathcal{K} = (K_1,\dots,K_M)$. For all i, j, κ and $\mathcal{K}$, let $E^i(j,\kappa,\mathcal{K})$ be the event that item j appear in positions i (in a random list), none of the other offered items appears in position $1,\dots,i-1$, and $\mathcal{X}$ appears after position i, given that we offer K_m items from class C_m and that k_m of them are among items $1,\dots,i$, for all $m \leq M$. With our notation,

$$E^i(j,\kappa,\mathcal{K}) = \{\sigma^{-1}(j) \leq i, j \succ_\sigma \mathcal{X}, j \succ_\sigma k, \forall k \in S\backslash\{j\}\},$$

where S is a subset of items containing K_m items from C_m among which k_m are in position $1, \ldots, i$ in L_0 for $m = 1, \ldots, M$. It turns out that $\Pr\left[E^i(j, \kappa, \mathcal{K}\right]$ does not depend on the choice of S but only on κ and $\mathcal{K}$. The proof is similar in spirit to that of Claim 3.1 and is therefore omitted. We can now properly define the states of our dynamic program.

Recall that we assume that $\mathcal{K}$ is known. For all $i \in [n+1]$ and $k_m \in [K_m]$ for all $m = 1, \ldots, M$, let $F(i, k_1, \ldots, k_M)$ be the maximum expected revenue we get from items that end up in positions $1, \ldots, i$ in a random list given that we pick k_m items from $C_m \cup \{1, \ldots, i\}$ for all $m \leq M$. Note that by the way the classes are defined, $F(n+1, K_1, \ldots, K_M)$ gives a $(1-\epsilon)$-optimal solution. Moreover, we have the following dynamic programming recursion:

$$
\begin{aligned}
&F(i, k_1, \ldots, k_M)\\
&= \max \left\{ \sum_{j=1}^{n} r_j \cdot \Pr\left[E^i(j, \kappa, \mathcal{K}) | \text{we offer item } i\right] + F(i-1, k_1, \ldots, k_{m(i)} - 1, \ldots, k_M), \right.\\
&\left. \sum_{j=1}^{n} r_j \cdot \Pr\left[E^i(j, \kappa, \mathcal{K}) | \text{we do not offer item } i\right] + F(i-1, k_1, \ldots, k_M) \right\},
\end{aligned} \tag{1}
$$

where $m(i)$ is the class to which item i belongs, i.e. $r_i \in C_{m(i)}$. When $i > d$, we have

$$
F(i, k_1, \ldots, k_M) = \sum_{j=1}^{n} r_j \cdot \Pr\left[E^i(j, \kappa, \mathcal{K})\right] + F(i-1, k_1, \ldots, k_{m(i)}, \ldots, k_M). \tag{2}
$$

The boundary condition $F(0, \kappa) = 0$ for all κ completes the description of the dynamic program and of the algorithm. Algorithm 3 describes the algorithm.

ALGORITHM 3: EPTAS

Let $\mathcal{A} = \emptyset$.
for all $K_1, \ldots, K_M$ such that $K_m \leq 1/\epsilon$ for all $m \leq M$ **do**
  Compute $F(i, k_1, \ldots, k_M)$ for all $i \in [n+1]$, $k_m \in [K_m]$ for all $m \leq M$ using (1) and (2).
  Let $\mathcal{A} = \{\mathcal{A}, S\}$, where S is the subset corresponding to $F(n+1, K_1, \ldots, K_M)$.
end
return $\arg\max\{\mathcal{R}(S), S \in \mathcal{A}\}$.

THEOREM 5.1. *Algorithm 3 is an EPTAS for the assortment optimization problem under the arbitrary swap model.*

The size of the dynamic program is $O((1/\epsilon)^M \cdot n) = O((1/\epsilon)^{O((1/\epsilon)\cdot\log(1/\epsilon))} \cdot n)$. Moreover, we need to enumerate over all possible nested assortments of $\mathcal{U}$ and guesses $\mathcal{K}$. Since there are at most n options for nested assortments of $\mathcal{U}$ and $O(1/\epsilon)^M$ for $\mathcal{K}$, the running time of the algorithm is $O((1/\epsilon)^{O((1/\epsilon)\cdot\log(1/\epsilon))} \cdot n^2)$ which makes our algorithm an EPTAS.

Computing the probabilities. In order to solve the dynamic program, we need to evaluate the probabilities $\Pr[E^i(j, \kappa, \mathcal{K})]$. However, these are complicated events for which we do not know how to compute the probabilities exactly. Therefore, our approach is to plug in estimators for these probabilities in the dynamic program, which are computed through sampling.

We next show that if the probability $\Pr[E^i(j,\kappa,\mathcal{K})]$ is too small, rounding it to zero does not affect the outcome of the dynamic program in a meaningful way. More precisely, let $\tilde{F}$ be the state of the dynamic program computed when rounding every probability with $\Pr[E^i(j,\kappa,\mathcal{K})] \leq \epsilon/n^4$ to zero. Note that an adaption of the proof of Lemma A.1 (see Appendix A.1) allows us to compute these probabilities efficiently using sampling. We prove the following result which proves that we can approximately solve the dynamic program.

LEMMA 5.2. $\tilde{F}(n, K_1, \dots, K_M) \geq (1-\epsilon)\cdot \mathcal{R}(S^*)$.

PROOF. Note that there exists a sequence of optimal $\kappa_{i,j}$ such that

$$F(n, K_1, \dots, K_M) = \sum_{i=1}^{n+1}\sum_{j=1}^{n} \Pr\left[E^i(j,\kappa_{i,j},\mathcal{K})\right]\cdot r_j.$$

Also, let $\widetilde{\Pr}[\cdot]$ be the rounded down probabilities. We bound two parts of this double sum separately.

Part 1. For every item j in $\mathcal{D}$, we have $r_j \leq r^* \leq 2\mathcal{R}(S^*)$ where $r^* = \max\{r_i : i \in \mathcal{D}\}$. Therefore,

$$\sum_{i=1}^{n+1}\sum_{j\in\mathcal{D}} \Pr\left[E^i(j,\kappa_{i,j},\mathcal{K})\right] r_j - \sum_{i=1}^{n+1}\sum_{j\in\mathcal{D}} \widetilde{\Pr}\left[E^i(j,\kappa_{i,j},\mathcal{K})\right] r_j \leq \sum_{i=1}^{n+1}\sum_{j\in\mathcal{D}} \frac{\epsilon}{n^4}2\mathcal{R}(S^*) \leq \frac{2\epsilon}{n^2}\mathcal{R}(S^*).$$

Part 2. We now look at items j in $\mathcal{U}$. In order to obtain a similar bound, consider what happens if we only offer the highest revenue item j^* in $\mathcal{U}$. In particular, $r_{j^*} = \max\{r_i : i \in \mathcal{U}\}$. In this case, given that there is at least one swap, $\Pr[j^* \succ_1 \mathcal{X} | X > 0] \geq 1/n^2$. When this happens, if there are additional swaps, $\Pr[j^* \succ_k \mathcal{X} | X > 0, j^* \succ_1 \mathcal{X}] \geq 1/2$ by Claim 2.1 for any $k \geq 1$. Consequently,

$$\mathcal{R}(S^*) \geq \mathcal{R}(\{j^*\}) \geq \Pr\left[X > 0\right]\cdot \frac{r_{j^*}}{2n^2},$$

and therefore, $\Pr[X > 0] r_j \leq 2n^2 \cdot \mathcal{R}(S^*)$ for all $j \in \mathcal{U}$. To finish the calculation, note that for $j \in \mathcal{U}$, $\Pr[E^i(j,\kappa,\mathcal{K})] = \Pr[X > 0]\cdot \Pr[E^i(j,\kappa,\mathcal{K}) | X > 0]$, since if there is no swap, this event happens with probability 0. We therefore get

$$\begin{aligned}
&\sum_{i=1}^{n+1}\sum_{j\in\mathcal{U}} \Pr\left[E^i(j,\kappa_{i,j},\mathcal{K})\right]\cdot r_j - \sum_{i=1}^{n+1}\sum_{j\in\mathcal{U}} \widetilde{\Pr}\left[E^i(j,\kappa_{i,j},\mathcal{K})\right]\cdot r_j \\
&\quad \leq \sum_{i=1}^{n+1}\sum_{j\in\mathcal{U}} \Pr\left[E^i(j,\kappa,\mathcal{K})\right]\cdot r_j \\
&\quad = \sum_{i=1}^{n+1}\sum_{j\in\mathcal{U}} \Pr\left[E^i(j,\kappa,\mathcal{K}) | X > 0\right] \Pr\left[X > 0\right]\cdot r_j \\
&\quad \leq n^2 \cdot \frac{\epsilon}{n^4}\cdot 2n^2 \cdot \mathcal{R}(S^*) \\
&\quad = 2\epsilon \cdot \mathcal{R}(S^*).
\end{aligned}$$

Combining the two cases, we get

$$\begin{aligned}F(n, K_1, \ldots, K_M) - \widetilde{F}(n, K_1, \ldots, K_M) &\leq F(n, K_1, \ldots, K_M) - \sum_{i=1}^{n+1} \sum_{j \in \mathcal{D}} \widetilde{\Pr}\left[E^i(j, \kappa_{i,j}, \mathcal{K})\right] \cdot r_j \\ &\leq 4\epsilon \cdot \mathcal{R}(S^*).\end{aligned}$$

□

6. GENERALIZATION TO MIXTURE OF RANDOM SWAP BASED DISTRIBUTIONS

In this section, we consider an extension of our model to a mixture over random swap based distributions where there is a collection of prototype lists and a corresponding probability distribution of starting from any of them to generate a random permutation. More precisely, let $\mathcal{L}_0$ be a set of prototype lists, where we assume that for all $L_0 \in \mathcal{L}_0$, the set of desirable items and undesirable items are identical. This mixture over our basic model allows to capture heterogenous consumers that may differ significantly in their ranking over the same set of desirable items. This is a significant generalization over the basic model with only one prototype list. For any pair of items i and j,

$$\Pr[i \succ j] = \sum_{L_0 \in \mathcal{L}_0} \Pr[L_0] \cdot \Pr[i \succ j | L_0].$$

The above expression suggests that the probabilistic claims of Section 2 continue to hold in the mixture model. Note however that these claims assume a certain ordering in the prototype list that now has to hold for all $L_0 \in \mathcal{L}_0$. We show that the analysis of our PTAS (Section 4) can be adapted to the case of mixture model and we have the following result.

THEOREM 6.1. *There exists a PTAS for the assortment optimization problem under a mixture of random swap based distribution over permutations model (for both arbitrary and adjacent swaps).*

We can also obtain the EPTAS for the case of arbitrary swap model. However, the running time exponential in $|\mathcal{L}_0|$. In order to extend the dynamic program to obtain an EPTAS for the mixture model, we need to introduce a variable κ for each $L_0 \in \mathcal{L}_0$. This implies that that the size of the dynamic program, and therefore the running time of the resulting algorithm, are exponential in the number of prototype lists. We therefore obtain the following result.

THEOREM 6.2. *There exists an EPTAS for the assortment optimization problem under a mixture of a fixed number of random arbitrary swap based distribution over permutations model.*

We defer the detailed proofs to the full version of the paper.

7. CARDINALITY CONSTRAINED ASSORTMENT OPTIMIZATION

In the cardinality constrained assortment optimization problem, we assume that there is an upper bound of C on the number of items to be picked. The problem can therefore be formulated as follows

$$\max_{S \subseteq [n], |S| \leq C} \mathbb{E}[\mathcal{R}(L, S)].$$

Note that the PTAS presented in Section 4 allows to handle a constraint of this nature. The modification needed is to set $K = \min\{C, 1/\epsilon\}$ and for each candidate assortment of $\mathcal{D}$, only enumerate over feasible nested assortment in $\mathcal{U}$. Since our algorithm is enumerating over small size solutions, it also returns the optimal constrained solution provided that we upper bound K by C. Furthermore, the results hold if the distribution over permutations is given by a mixture of random swap based distributions.

THEOREM 7.1. *There exists a PTAS for the cardinality constrained assortment optimization problem under a mixture of random swap based distribution over permutations model (for both arbitrary and adjacent swaps).*

Note that similar remarks also apply to the EPTAS for the arbitrary swap model, and therefore our EPTAS can also be adapted to handle a cardinality constraint for the mixture of a constant number of random arbitrary swap based distribution over permutations.

7.1. General capacity constrained assortment optimization

We now consider a general capacity constrained assortment problem. Here, each item i has a weight w_i, and there is a total capacity of W. The problem can be formulated as

$$\max_{S\subseteq[n],\sum_{i\in S} w_i\leq W} \mathbb{E}\left[\mathcal{R}(L,S)\right].$$

We show that there exists a $(1-\epsilon)$-optimal solution to the capacity constrained assortment optimization problem with at most $1/\epsilon$ items from $\mathcal{D}$. However, unlike in the unconstrained or cardinality constrained case, Claim 3.1 does not hold anymore, and the optimal solution is not guaranteed to pick a nested solution from the items in $\mathcal{U}$. Note that after having picked a candidate assortment from $\mathcal{U}$, the problem reduces to finding the best possible assortment from items in $\mathcal{D}$ subject to an adjusted capacity constraint (the total capacity is reduced by the weights of items selected from $\mathcal{U}$).

Suppose that we fix some assortment from $\mathcal{U}$ and in addition, the number of items, say C, to be picked from $\mathcal{D}$. The goal is to pick exactly C items from $\mathcal{U}$ to complete the assortment and maximize the expected revenue subject to the capacity constraint. Given that we pick exactly C items from $\mathcal{U}$, any item from $\mathcal{U}$ that we pick has the same probability of being chosen by a random consumer. Consequently, the problem reduces to a knapsack problem with an additional exact cardinality constraint. For this problem, [Caprara et al. 2000] give a PTAS. Since the size of the optimal assortment in $\mathcal{D}$ has cardinality at most $|\mathcal{D}|$, we only have to run this PTAS at most $|\mathcal{D}|$ times to complete any assortment from $\mathcal{U}$ into a near-optimal assortment with items from $\mathcal{D}$ subject to a capacity constraint. Consequently, we have the following result.

THEOREM 7.2. *There exists a PTAS for the capacity constrained assortment optimization problem under a mixture of random swap based distribution over permutation model (for both arbitrary and adjacent swaps).*

We defer a detailed discussion to the full version of the paper.

REFERENCES

Ali Aouad, Vivek Farias, Retsef Levi, and Danny Segev. 2015. The approximability of assortment optimization under ranking preferences. *Working paper* (2015).

Brenda S. Baker. 1994. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM* 41, 1 (1994), 153–180.

M. Ben-Akiva and S. Lerman. 1985. *Discrete choice analysis: theory and application to travel demand*. Vol. 9. MIT press.

Alberto Caprara, Hans Kellerer, Ulrich Pferschy, and David Pisinger. 2000. Approximation algorithms for knapsack problems with cardinality constraints. *European Journal of Operational Research* 123, 2 (2000), 333–345.

James Davis, Guillermo Gallego, and Huseyin Topaloglu. 2013. Assortment planning under the multinomial logit model with totally unimodular constraint structures. *Department of IEOR, Columbia University. Available at http://www. columbia. edu/gmg2/logit_const. pdf* (2013).

Jean-Paul Doignon, Aleksandar Pekeč, and Michel Regenwetter. 2004. The repeated insertion model for rankings: Missing link between two subset choice models. *Psychometrika* 69, 1 (2004), 33–54.

Leah Epstein and Asaf Levin. 2014. An efficient polynomial time approximation scheme for load balancing on uniformly related machines. *Mathematical Programming* 147, 1-2 (2014), 1–23.

Fedor V. Fomin, Daniel Lokshtanov, Venkatesh Raman, and Saket Saurabh. 2011. Bidimensionality and EPTAS. In *Proceedings of the 22nd annual ACM-SIAM symposium on Discrete Algorithms*. 748–759.

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

Vineet Goyal, Retsef Levi, and Danny Segev. 2016. Near-optimal algorithms for the assortment planning problem under dynamic substitution and stochastic demand. *Operations Research* 64, 1 (2016), 219– 235.

Dorothee Honhon, Sreelata Jonnalagedda, and Xiajun Amy Pan. 2012. Optimal algorithms for assortment selection under ranking-based consumer choice models. *Manufacturing & Service Operations Management* 14, 2 (2012), 279– 289.

Klaus Jansen. 2010. An EPTAS for scheduling jobs on uniform processors: using an MILP relaxation with a constant number of integral variables. *SIAM Journal on Discrete Mathematics* 24, 2 (2010), 457–485.

A. Gürhan Kök, Marshall L. Fisher, and Ramnath Vaidyanathan. 2015. Assortment planning: Review of literature and industry practice. In *Retail supply chain management*. Springer, 175–236.

Kelvin Lancaster. 1990. The economics of product variety: A survey. *Marketing Science* 9, 3 (1990), 189–206.

Tyler Lu and Craig Boutilier. 2011. Learning Mallows models with pairwise preferences. In *Proceedings of the 28th International Conference on Machine Learning*. 145–152.

R.D. Luce. 1959. *Individual choice behavior: A theoretical analysis*. Wiley.

Colin L. Mallows. 1957. Non-null ranking models. I. *Biometrika* (1957), 114–130.

D. McFadden. 1978. Modeling the choice of residential location. *Transportation Research Record* 672 (1978), 72–77.

D. McFadden. 1980. Econometric models for probabilistic choice among products. *Journal of Business* 53, 3 (1980), 13–29.

R.L. Plackett. 1975. The analysis of permutations. *Applied Statistics* 24, 2 (1975), 193–202.

Kamalini Ramdas. 2003. Managing product variety: An integrative review and research directions. *Production and operations management* 12, 1 (2003), 79–101.

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

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

B. Wierenga. 2008. Handbook of Marketing Decision Models, vol. 121 of International Series in Operations Research & Management Science. (2008).

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

A. APPENDIX

A.1. Computing the expected revenue

In this appendix, we show that for a given subset of items S, we can efficiently compute $\mathcal{R}(S)$ by sampling. More precisely, we have the following result.

LEMMA A.1. *For a given subset $S \subseteq [n]$, let $\hat{R}(S)$ be the sample revenue computed using $O(\log \delta \cdot n^3/\epsilon^2)$ samples. With probability at least $1 - 2\delta$,*

$$|\hat{\mathcal{R}}(S) - \mathcal{R}(S)| \leq 4\epsilon \cdot \mathcal{R}(S).$$

PROOF. For any item $i \in S$, let E_i be the event that item i is the first to appear in a random list out of S and $\mathcal{X}$. With this definition,

$$\mathcal{R}(S) = \sum_{i \in S} \Pr[E_i] \cdot r_i.$$

Let i^* be the highest revenue item in S. For all item $i \in S$, let $\hat{\mu}_i$ be the average number of times E_i is observed in $m = \log \delta \cdot n^3/2\epsilon^2$ samples. By the Chernoff-Hoeffding bound, we have

$$\Pr\left[|\hat{\mu}_i - \Pr[E_i]| \geq \frac{\epsilon}{n^3}\right] \leq 2e^{-2m\epsilon^2} = 2\delta.$$

Consequently, with probability at least $1 - 2\delta$,

$$|\hat{\mu}_i - \Pr[E_i]| \leq \frac{\epsilon}{n^3}.$$

We now bound the revenue of any item as a function of $\mathcal{R}(S^*)$. We consider two cases.

Case 1. For every item $j \in \mathcal{D}$, we have $r_j \leq r^* \leq 2\mathcal{R}(S^*)$ where $r^* = \max\{r_i : i \in \mathcal{D}\}$ (see proof of Theorem 4.1).

Case 2. We now look at items j in $\mathcal{U}$. Consider what happens if we only offer the highest revenue item j^* in $\mathcal{U}$. In particular, $r_{j^*} = \max\{r_i : i \in \mathcal{U}\}$. In this case, given that there is at least one swap, $\Pr[j^* \succ_1 \mathcal{X} | X > 0] \geq 1/n^2$. When this happens, if there are additional swaps, $\Pr[j^* \succ_k \mathcal{X} | X > 0, j^* \succ_1 \mathcal{X}] \geq 1/2$ by Claim 2.1 for any $k \geq 1$. Consequently,

$$\mathcal{R}(S^*) \geq \mathcal{R}(\{j^*\}) \geq \Pr[X > 0] \cdot \frac{r_{j^*}}{2n^2},$$

and therefore, $\Pr[X > 0] r_j \leq 2n^2 \cdot \mathcal{R}(S^*)$ for all $j \in \mathcal{U}$. Therefore, for all $j \in \mathcal{U}$,

$$\Pr[E_j] r_j \leq \Pr[E_j | X > 0] \Pr[X > 0] r_j \leq 2n^2 \cdot \mathcal{R}(S^*).$$

We can now put everything together to prove the desired inequality. Let $\tilde{S}$ be the set of items from S such that $\Pr[E_i] \leq \epsilon/n^3$. We have with probability at least $1 - 2\delta$,

$$\begin{aligned}
\left|\hat{R}(S) - \mathcal{R}(S)\right| &\leq \left|\sum_{i \notin \tilde{S}} (\hat{\mu}_i - \Pr[E_i]) r_i\right| + \sum_{i \in \tilde{S}} (\hat{\mu}_i + \Pr[E_i]) \cdot r_i \\
&\leq \sum_{i \notin \tilde{S}} \frac{\epsilon}{n^3} \cdot 2n^2 \cdot \mathcal{R}(S^*) + \sum_{i \in \tilde{S}} \frac{2\epsilon}{n^3} \cdot 2n^2 \cdot \mathcal{R}(S^*) \\
&\leq 4\epsilon \mathcal{R}(S^*).
\end{aligned}$$

□

A.2. Proof of Claim 2.1 for adjacent swaps

Without loss of generality, we assume that $r^0(i) = i$ and $r^0(j) = i + 1$. We condition on the first swap. Suppose, i and j are swapped together. In that case, if there exists two other items ℓ and m, then by coupling the swap involving both i and j with the swap (ℓ, m), we have

$$\Pr\left[i \succ_k j | \text{the first swap is } (i,j) \text{ or } (\ell, m)\right] = \frac{1}{2}$$

Therefore, let $E = \{(i, i+1), (\ell, m)\}$. We have

$$\Pr\left[i \succ_k j\right] = \frac{1}{2} \cdot \Pr\left[E\right] + \Pr\left[i \succ_{k-1} j | i \succ_0 j\right] \cdot (1 - \Pr\left[E\right])$$

The proof is then completed by induction.

A.3. Analysis of the bad example for arbitrary swaps

We first take a look at the possible nested assortments. These are $S_1 = \mathcal{U}$, $S_2 = \{1\} \cup \mathcal{U}$, and $S_3 = \{1, d\} \cup \mathcal{U}$. We then consider $S_4 = \{d\} \cup \mathcal{U}$ which is not a nested assortment, and show that it performs better than the nested assortments.

Assortment S_1. In this assortment, all the items of $\mathcal{U}$ are offered and no item from $\mathcal{D}$. Therefore, if $X = 0$, the revenue obtained is 0. Otherwise, we get a revenue of $2/(1-\alpha)$ if one item of $\mathcal{U}$ gets swapped before $\mathcal{X}$, i.e. if one item of $\mathcal{U}$ together with one item of $\mathcal{D} \cup \mathcal{X}$ are swapped. This happens with probability

$$\frac{n(n+3)}{\binom{2n+3}{2}} = \frac{n(n+3)}{(2n+3)(n+1)} = \frac{1}{2} + O\left(\frac{1}{n}\right).$$

Consequently,

$$\mathcal{R}(S_1) = (1-\alpha) \cdot \left(\frac{1}{2} + O\left(\frac{1}{n}\right)\right) \cdot \frac{2}{1-\alpha} = 1 + O\left(\frac{1}{n}\right).$$

Assortment S_2. In this assortment, we offer item 1 together with all items in $\mathcal{U}$. If $X = 0$, the revenue obtained is $1 + \epsilon$. If $X = 1$, the revenue is not $1 + \epsilon$ if and only if item 1 is swapped with an item from $\mathcal{U}$ or with $\mathcal{X}$. Therefore,

$$\begin{aligned}
\mathcal{R}(S_2) &= (1+\epsilon) \cdot \alpha + (1-\alpha) \cdot \left(\frac{n}{\binom{2n+3}{2}} \cdot \frac{2}{1-\alpha} + \left(1 - \frac{n+1}{\binom{2n+3}{2}}\right) \cdot (1+\epsilon)\right) \\
&= \left(1 - \frac{1-\alpha}{(2n+3)}\right) \cdot (1+\epsilon) + \frac{2n}{(n+1)(2n+3)} \\
&\leq 1 + \epsilon + \frac{2n}{(n+1)(2n+3)} \\
&= 1 + \epsilon + O\left(\frac{1}{n}\right).
\end{aligned}$$

Assortment S_3. In this last nested assortment, we offer items 1 and d together with $\mathcal{U}$. Note that this assortment gives a lower revenue than S_2. Indeed, the only difference is when $X = 1$ and the swap involves item 1 and item d. In that case, the revenue is 1 instead of $1 + \epsilon$. Therefore, $\mathcal{R}(S_3) \leq \mathcal{R}(S_2)$.

Assortment S_4. We perform a similar analysis for $S_4 = \{d\} \cup \mathcal{U}$. If $X = 0$, the revenue is 1. When $X = 1$, consider the event E where item d is not swapped. This happens with probability

$$1 - \frac{2n+2}{\binom{2n+3}{2}} = 1 - \frac{2}{2n+3}.$$

Conditional on that event E, the revenue is $2/(1-\alpha)$ if any item in $\mathcal{U}$ is swapped with an item in $\{1,\ldots,n+1\}$. This happen with probability

$$\frac{n(n+1)}{\binom{2n+2}{2}} = \frac{2n(n+1)}{(2n+2)(2n+1)} = \frac{n}{2n+1}.$$

As a result, we have the following lower bound on the revenue of S_4:

$$\begin{aligned}
\mathcal{R}(S_4) &\geq \alpha + (1-\alpha)\cdot\left(1-\frac{2}{2n+3}\right)\cdot\left(\frac{n}{2n+1}\cdot\frac{2}{1-\alpha}+\left(1-\frac{n}{2n+1}\right)\right)\\
&= \alpha + \left(1-\frac{2}{2n+3}\right)\cdot\left(\frac{2n}{2n+1}+\left(1-\frac{n}{2n+1}\right)\cdot(1-\alpha)\right)\\
&= \alpha + \left(1+O\left(\frac{1}{n}\right)\right)\cdot\left(1-\frac{1}{2n+1}+\frac{n+1}{2n+1}\cdot(1-\alpha)\right)\\
&= \frac{3+\alpha}{2}+O\left(\frac{1}{n}\right).
\end{aligned}$$

Combining all cases together, we have

$$\frac{\mathcal{R}(S_4)}{\max\{\mathcal{R}(S_1),\mathcal{R}(S_2),\mathcal{R}(S_3)\}} = \frac{5}{2}-\epsilon-O\left(\frac{1}{n}\right),$$

since we can pick α arbitrary close to 1.

A.4. Bad example for adjacent swaps

In this section, we exhibit an example for the case of adjacent swaps for which a nested solution is not optimal. Unlike the case of arbitrary swaps, we were not able to construct counter-example that can be rigorously analyzed without a lengthy proof. Instead, while our construction is pretty simple, we estimated the revenue of each solution via numerical simulations (10,000 samples were used).

The construction. Consider the following example with $2n+2$ items and $d = n+2$. The revenues are as followed

$$r_i = \begin{cases} 1.1 & \text{if } i = 1 \\ 0 & \text{if } 2 \leq i \leq n+1 \\ 1 & \text{if } i = n+2 \\ 2 & \text{if } i > n+2 \end{cases}.$$

The initial preference list is ordered as follows (here the revenues are used instead of the item numbering) :

$$L_0 = \left[1.1 \succ \underbrace{0 \succ \cdots \succ 0}_{n \text{ items}} \succ 1 \succ \mathcal{X} \succ \underbrace{2 \succ \cdots \succ 2}_{n \text{ items}}\right].$$

Note that since all the items in $\mathcal{U}$ have the same revenue, the initial list is deterministic. Additionally, we consider the following distribution on the number of swaps:

$$X = \begin{cases} 0 & \text{with probability } 1/2 \\ 3,000 & \text{otherwise} \end{cases}.$$

For $n = 30$, we get the following numerical results. Let $S_1 = \{1\}$, $S_2 = \mathcal{U}$, $S_3 = \{1\}\cup\mathcal{U}$ and $S_4 = \{n+2\}\cup\mathcal{U}$. Note that S_1, S_2, and S_3 are all nested solutions with respect to the set of desired items $\mathcal{D}$. We have

$$\mathcal{R}(S_1) = 1.0879, \qquad \mathcal{R}(S_2) = 0.8360, \qquad \mathcal{R}(S_3) = 1.1153, \qquad \mathcal{R}(S_4) = 1.2920.$$

Therefore, the optimal solution is S_4.