Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Near-Optimal Algorithms for the Assortment Planning Problem Under Dynamic Substitution and Stochastic Demand

Vineet Goyal, Retsef Levi, Danny Segev

OPERATIONS RESEARCH
Vol. 64, No. 1, January–February 2016, pp. 219–235
ISSN 0030-364X (print) | ISSN 1526-5463 (online)

http://dx.doi.org/10.1287/opre.2015.1450

Near-Optimal Algorithms for the Assortment Planning Problem Under Dynamic Substitution and Stochastic Demand

Vineet Goyal
Department of Industrial Engineering and Operations Research, Columbia University, New York, New York 10027, vgoyal@ieor.columbia.edu

Retsef Levi
Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, retsef@mit.edu

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

Assortment planning of substitutable products is a major operational issue that arises in many industries such as retailing, airlines, and consumer electronics. We consider a single-period joint assortment and inventory planning problem under dynamic substitution with stochastic demands, and provide complexity and algorithmic results as well as insightful structural characterizations of near-optimal solutions for important variants of the problem. First, we show that the assortment planning problem is NP-hard even for a very simple consumer choice model, where each consumer is willing to buy only two products. In fact, we show that the problem is hard to approximate within a factor better than $1 - 1/e$. Second, we show that for several interesting and practical customer choice models, one can devise a *polynomial-time approximation scheme* (PTAS), i.e., the problem can be solved efficiently to within any level of accuracy. To the best of our knowledge, this is the first efficient algorithm with provably near-optimal performance guarantees for assortment planning problems under dynamic substitution. Quite surprisingly, the algorithm we propose stocks only a *constant* number of different product types; this constant depends only on the desired accuracy level. This provides an important managerial insight that assortments with a relatively small number of product types can obtain almost all of the potential revenue. Furthermore, we show that our algorithm can be easily adapted for more general choice models, and we present numerical experiments to show that it performs significantly better than other known approaches.

Keywords: assortment planning; dynamic substitution; polynomial time approximation schemes.
Subject classifications: analysis of algorithms; inventory/production: approximations/heuristics.
Area of review: Optimization.
History: Received June 2009; revisions received October 2012, August 2014, May 2015; accepted August 2015. Published online in *Articles in Advance* January 6, 2016.

1. Introduction

Assortment planning is a major operational issue that arises in many industries such as retailing, airlines, and consumer electronics. Given a set of products that are differentiated by price, quality, and possibly other attributes, one has to decide on the product assortment and the respective quantities that will be stocked and offered to customers. Such decisions become particularly important when different products are *substitutable* and customers exhibit a *substitution behavior*. For example, customers may a priori prefer product A to product B but may still be willing to buy product B if product A is not offered or not available anymore. The substitution behavior can be *assortment-based* (or *static*), i.e., unaffected by the availability of products and depends only on the specific assortment of products, or it can be *stock-out-based* (or *dynamic*), i.e., driven by stock-out events and availability of products. When customers exhibit substitution behavior (static or dynamic), the demands for different product types are correlated, and accounting for product substitutability can lead to significantly higher revenues and profits. However, this requires joint multiproduct assortment and inventory decisions, which usually give rise to complex optimization models that are computationally challenging. Assortment planning under substitution forms one of the core problem domains in revenue management, and many variants of these problems have been studied extensively in the literature.

We consider a single-period joint assortment and inventory planning problem with stochastic demand and dynamic substitution. Specifically, we study a single period model with finite number of product types (or item types), each with a per-unit selling price and potentially other attributes that differentiate between different product types (e.g., quality, size, color). At the beginning of the period, one has to decide jointly on the assortment and the inventory levels, i.e., which product types to offer and how many units to stock from each offered product, subject to a *capacity constraint* on the total number of units that can be stocked. After the assortment and inventory decisions are made, a stochastic number of customers arrive one after the other, each with a *random preference* on the product types. A preference is an ordered list or permutation of product types that reflects the order in which the customer prefers different product types. Note that the no-purchase alternative can appear at any position in the preference. No-purchase alternative at any position denotes that buying nothing is more preferable to buying any product below this position. We assume that the preference of each customer is independent and identically distributed according to a known distribution over all potential preferences. Upon arrival, each customer purchases the first available product type in her preference list. If no product on the customer's list is available, the customer leaves without purchasing any product. The goal is to find the assortment and inventory levels that maximize the expected revenue obtained from the units purchased by customers. Note that both the number of customers and their respective preferences are stochastic in this model, and the customers arrive in a sequential manner.

1.1. Our Results

Complexity and approximability. The model described above is in general computationally intractable. Specifically, we show that the above-mentioned problem is *NP-hard* even for the special case when there is only one customer and all preference lists consist of only two product types, i.e., there is no efficient algorithm for solving the problem to optimality, unless P = NP. Therefore, it is only natural to seek for approximation algorithms that compute near-optimal solutions to the problem. However, even the model with a single customer (but with general preference lists) can be shown to be hard to approximate within some fixed constant. In particular, there is no polynomial-time algorithm that is guaranteed to recover at least $1-1/e$ fraction of the optimal expected revenue for all possible instances of the problem unless P = NP. Therefore, the worst-case approximation factor for any efficient algorithm cannot be better than $1-1/e$, unless P = NP. We present the hardness results in Section EC.2 of the online appendix (available as supplemental material at http://dx.doi.org/10.1287/opre.2015.1450).

.

Polynomial time approximation scheme. In view of these hardness of approximation results, we study this problem with two additional assumptions that still capture important practical situations. We first focus attention on *nested* preference lists. Here, the product types are ordered by increasing per-unit selling price. Customers always prefer the cheapest product type available upon arrival. Each preference list corresponds to a different price threshold, and a customer with that preference list is willing to buy all product types with per-unit price lower than the threshold. Nested preference lists arise in situations where the quality of different product types is similar, and customers differentiate only by price. (Similar choice models have been studied by Talluri and van Ryzin 2004.) The second assumption is that the number of customers follows an increasing failure rate (IFR) distribution. (For a definition of an IFR distribution, see Shaked and Shanthikumar 1994 and Assumption 2 in §3.) This is a well-known class of distributions that includes many of the traditional distributions used in the operations research and operations management literature including, among many others, the normal, uniform, exponential, geometric, Poisson, and even beta distributions for certain parameters.

Interestingly, by imposing the above-mentioned assumptions, one can design a *polynomial time approximation scheme* (PTAS) for the problem, i.e., for any accuracy level $0 < \epsilon < 0.5$, one can compute a solution with expected revenue at least $(1-\epsilon)$ of the optimal expected revenue, in time that is polynomial in the input size for any fixed ϵ. Practically speaking, this result implies that the problem can be solved efficiently to within any degree of accuracy. This stands in contrast to the inapproximability result, stating that there is no approximation algorithm for the general model with worst-case performance guarantee better than $1-1/e$ unless P = NP. To the best of our knowledge, this is the first efficient algorithm with provably good performance guarantees for assortment planning problems under dynamic substitution. Moreover, our algorithm stocks only $O((1/\epsilon)\cdot\log(1/\epsilon))$ different product types. Note that this constant depends only on the desired accuracy level ϵ, and not on any other parameter of the problem, including the overall number of product types or the capacity. (It is worth pointing out that these parameters affect the actual product types being offered and their respective quantities but not how many different product types are picked.) This provides an important managerial insight that assortments with a relatively small number of product types can obtain almost all of the potential revenue.

There have been relatively few approximation schemes for stochastic optimization models (for example, see the recent results in Halman et al. 2009, 2008). Most of these results are based on formulating the respective problem as a dynamic program that can be solved in pseudo-polynomial time and then employ that to devise a *fully polynomial approximation algorithm* (FPTAS). (The running time of an FPTAS depends polynomially on $1/\epsilon$, compared to a PTAS where it can depend on an arbitrary function of $1/\epsilon$.) In contrast, the model studied in this paper does not seem to admit a tractable dynamic program. Instead, we use several structural properties of a near-optimal solution to identify a subset of product types of constant size, which in turn leads to a PTAS. This concept has previously been applied to other combinatorial optimization problems (see, for instance, de la Vega and Lueker 1981, Har-Peled 2011), but, to the best of our knowledge, not to stochastic optimization problems. We believe that the new ideas introduced in this paper may also be applicable in other substitution and revenue management models and, more generally, stochastic optimization models.

We also show that if the distribution of the number of customers is not IFR, small subsets of product types cannot guarantee near-optimal performance. In particular, we construct an example, in which any solution that stocks a constant number of product types performs arbitrarily bad compared to the optimal policy.

Extensions and computational experiments. By employing dynamic programming techniques, we show how to leverage our PTAS to a more general choice model, in which customer choices are affected not only by price but also by quality. Specifically, each customer is willing to buy products within a certain *quality category*, and within that category, product types are differentiated only by price and admit a nested form. The underlying assumption is that the prices of different quality categories are separated to different customer segments. This choice model captures several important practical settings.

In addition, we describe an extension of our algorithmic approach to choice models different than the ones we obtained a worst-case analysis. Furthermore, we conducted extensive computational experiments using the well-known multinomial logit (MNL) choice model to examine how the resulting algorithms perform compared to other algorithmic approaches previously proposed. The experiments and the results are described in §5, and indicate that our approach performs generally better than previous approach.

Maximizing revenue vs. maximizing profit. We emphasize that we consider a model with a capacity constraint on the total number of units being stocked and assume that there are no per-unit purchasing costs. This is in contrast to several dynamic substitution models considered in the literature where each product has a per-unit revenue and a per-unit purchasing cost and the goal is to maximize the expected profit (for instance, see Mahajan and van Ryzin 2001 and Netessine and Rudi 2003). Capacity constraints arise in many retailing settings, for example, when there is limited shelf space. Ignoring purchasing prices is an appropriate assumption when the cost of buying or producing the products is a sunk cost (e.g., seats in an airplane), or when the costs are identical (e.g., fashion industry), or when unsold products can be fully or almost fully salvaged. Therefore, it is important to study models with capacity constraints instead of modeling purchasing costs. It is worth pointing out that the objective of maximizing revenue (instead of profit) has also been studied in a recent paper by Fisher and Vaidyanathan (2007) that describes a practical application of assortment optimization in the retail world.

Whereas the above works provide the practical motivation to our assumption, we also show how our method can be extended to compute near-optimal solutions for models with purchasing costs. This result is obtained by essentially reducing such models to a multicapacitated setting, where carefully picked subsets of products are given separate capacity constraints. More specifically, under the technical assumptions used to design the above-mentioned approximation scheme, suppose in addition that we are given a budget constraint of B on the total purchasing cost, under which the optimal inventory levels guarantee an expected revenue of R^*. Then, by following the approximation algorithm, we sketch in Section EC.1.2 of the online appendix, one can compute an inventory vector in which the expected revenue is at least $(1-\epsilon)R^*$, without exceeding the total purchasing cost. That being said, the downside of this extension is that the resulting algorithm is no longer a PTAS but rather a quasi-PTAS (where the running time exponent also involves factors that are polylogarithmic in the input size; see, for instance, Bansal et al. 2006, Remy and Steger 2009, Chan and Elbassioni 2011).

1.2. Literature Review

Joint assortment planning and inventory management problems with substitution have been extensively studied; we refer the reader to directly related papers, surveys, and books (Kok et al. 2006, Lancaster 1990, Ho and Tang 1998, Ramdas 2003) for a comprehensive review of the recent literature. Pentico (1974) was one of the earliest to consider an assortment planning problem with *downward substitution* and a deterministic sequence of customer arrivals, showing that under several conditions, a certain planning-horizon type policy is optimal. Van Ryzin and Mahajan (1999) consider a static substitution model with MNL demand distributions. They show that in this model the optimal solution consists of the most popular product. Cachon et al. (2005) generalize this static substitution model to incorporate search costs. Hopp and Xu (2005) consider the problem of integrating assortment decisions with pricing decisions, again with MNL demands. Anupindi et al. (2006) consider a *probit* demand model and include a penalty for customer's disutility in substituting to less preferred product type.

The above-mentioned papers are primarily focused on static substitution models, where customer preferences and purchasing decisions depend only on the assortment being offered but not on the specific inventory levels observed at the time of purchase. Specifically, customers do not substitute other product types just because more preferred types are stocked out. Thus, the demand for each product type is independent of the actual inventory levels of other products and only depends on the subset of product types being offered. However, in many practical applications such as airlines and retailing, customers do not exhibit static substitution behavior but rather a dynamic substitution. In particular, customers readily substitute when a more preferred product type is stocked out. Models with dynamic substitution are generally more complex and challenging. The demand for a specific product type is affected not only by the assortment being offered but also by the respective inventory levels that change dynamically over time as customers arrive and consume products. As a result, assortment and inventory decisions must be made simultaneously.

Whereas the static substitution model has been extensively studied, there is relatively little work on dynamic substitution models. Parlar and Goyal (1984) were the first to study a dynamic substitution model. They consider a probabilistic substitution model and show that the profit function is concave for a wide range of problem parameters. Smith and Agrawal (2000) consider a dynamic substitution model specified by first-choice probabilities and a substitution matrix, and show that static substitution yields bounds on the demand for each product in the dynamic substitution case. Mahajan and van Ryzin (2001) study a joint assortment and inventory planning problem with stochastic demands and general preferences where each product type has per-unit revenue and cost and the goal is to maximize the expected profit. Assuming that customer sequences can be sampled, they propose a sample path gradient-based algorithm and show that under fairly general conditions it converges to a local maximum. However, they do not provide any performance bounds for the expected profit of a local maximum as compared to the optimal expected profit.

Netessine and Rudi (2003) consider a substitution model where each customer preference consists of only two products (a first-choice as well as a second-choice product), in the assortment of an arbitrary number of products, and obtain analytically tractable solutions for the assortment planning problem in both centralized inventory management and competition. It is worth pointing out that we prove the assortment planning problem (in our model) to be NP-hard even when each customer preference consists of only two products. However, our model is different from the model studied by Netessine and Rudi (2003); they consider an uncapacitated problem with a per-unit purchasing cost for each product type and stocking a fractional number of any product type is allowed, whereas we consider a model with a capacity constraints on the total number of units that can be stocked across all product types. As mentioned earlier, it is more useful to consider a model with capacity constraints instead of purchasing costs for some applications (for instance, airline seats and fashion industry). Moreover, we also show that a model with purchasing costs can be approximately reduced to a problem with only capacity constraints (and no purchasing costs).

Kok and Fisher (2007) assume an MNL demand model within a Bayesian framework, propose an algorithm to estimate the model parameters, and also solve the assortment and planning problem with one-level stock-out based substitution. Gaur and Honhon (2006) give a heuristic for the problem under a location choice model based on the solution of the static substitution case. Honhon et al. (2010) consider a general customer choice model with a stochastic demand, but the sequence of customer preferences satisfies the following property: if a certain preference list occurs with probability p, then for every sequence of $1/p$ customers there will be one customer with this preference list. Therefore, the customer choices are not completely random in their model. The authors provide a novel characterization of the local maxima and propose a dynamic programming based algorithm to solve the problem. However, the running time of this algorithm is exponential in the number of product types, implying that it is practical only in cases where the number of product types is small. Nagarajan and Rajagopalan (2008) consider a dynamic substitution model where the individual demands for different products are negatively correlated, and they show that a *partially decoupled* policy is optimal under fairly general conditions. Chen and Bassok (2008) study the problem under a general customer choice model where all product types have identical prices and costs. Assuming sample path based allocation (i.e., one sees all customers first and decides how to allocate the inventories), they show that if the number of customers is fixed, then with high probability all the demand can be satisfied by stocking a total of N units of the products even when the customer preferences are random. However, the authors assume that the allocation of products to customers is simultaneous instead of sequential. That is, the products can be allocated by the retailer after all customers have arrived and their preferences become known to the retailer, as long as the product appears in the preference list of the customer. Using normal approximations, the authors also argue that, for a sufficiently large number of customers, the sample path based allocation model is a good approximation for the sequential allocation model, in which customers arrive one after the other and pick the most preferred product type among the available ones.

1.3. Outline

The rest of this paper is organized as follows. Section 2 provides a mathematical formulation of the model. In §3, we describe the PTAS and establish its performance analysis. In §4, we discuss the extension to the more general model of quality categories. In Section EC.3 of the online appendix, we also show that when the number of customers follows a non-IFR distribution, structural results similar to those in §3 need not hold. In Section 5, we present extensive computational experiments, where the performance of our approach is tested when applied to more general settings.

2. Model Formulation

Consider n product types with per-unit selling prices $p_1 \leqslant \cdots \leqslant p_n$, respectively, and a capacity bound C on the total number of units (across all product types) that can be stocked. A random number of customers, say M, arrive one after the other, where M is random variable with a known distribution. Each customer $j = 1, \ldots, M$ has a random preference list L_j, which specifies a subcollection of products in decreasing order of preference. This list comes from a known distribution and is independent and identically distributed among customers and also independent of the number of customers M. Upon arrival, each customer purchases the first available product in her list, assuming that at least one unit of such products exists at that time; otherwise, the customer leaves without purchasing at all. For an inventory vector $(y_1, \ldots, y_n)$, where y_i specifies the number of units stocked from product type i, let $R_m(y_1, \ldots, y_n)$ denote the revenue attained if m customers arrive. Observe that this revenue is still random caused by stochasticity in the preference lists of customers. The objective is to determine the inventory level of each product type, subject to the capacity constraint on the total number of units being stocked, so that the expected revenue is maximized, i.e.,

$$\max_{(y_1,\ldots,y_n)\in\mathbb{Z}_+^n}\left\{\mathrm{E}[R_M(y_1,\ldots,y_n)]\colon \sum_{i=1}^n y_i \leqslant C\right\}.$$

Note that the expectation is taken with respect to the number of customers M and their stochastic preference lists. We will use $(y_1^*,\ldots,y_n^*)$ to denote the optimal solution and OPT to denote the optimal expected revenue.

As outlined in §1.2 above, the model studied in this paper has received quite extensive attention in the operations management and the operations research literature, since it seems to capture some fundamental trade-offs that are central to management decisions in practice. The definition of an item in our model will depend on the assumption regarding the choice model. Specifically, it refers to the granular unit based on which customers choose. In many cases this could be a single SKU and in others a family of products that have the same price and common characteristics. In other words, the definition of item will depend on how refined the choice model is. The single period assumption captures many settings, in which it is impractical to assume frequent replenishment of the inventories, either because no back store inventory is being kept or because operationally the replenishment cannot occur frequently (say more than once a day, at the end of the day). The definition of the "period" would be determined by the specific setting being modeled. Clearly, there are situations in which multiperiod models are more appropriate, but these models tend to be extremely hard to solve. As a result, many practical approaches to model and devise policies for multiperiod settings rely on solving single period problems repeatedly, in which case variants of the model studied in this paper serve as a building block in the design of practical solutions to more complex models.

3. Polynomial Time Approximation Scheme

We show that the general model described in §2 is hard to approximate beyond a certain degree of accuracy (see Section EC.2 of the online appendix for details). In what follows, we consider a special case of the model obtained by imposing additional assumptions and give a polynomial time approximation scheme (PTAS) for computing near-optimal solutions with arbitrary level of accuracy. For any fixed $0<\epsilon<0.5$, our algorithm computes an inventory vector that obtains a fraction of at least $(1-\epsilon)$ of the optimal expected revenue, in time polynomial in the input size. We proceed by listing the additional assumptions.

Assumption 1: *Nested preference lists.* We assume that the common distribution from which customers pick their preference lists consists of only lists of the form $L=(1,\ldots,\ell)$ for some $\ell\leqslant n$. Therefore, there are $n+1$ possible preferences lists, $()$, (1), $(1,2)$, $\ldots$, $(1,2,\ldots,n)$, but their respective probabilities can be arbitrary (here, $()$ denotes the empty preference list for customers who prefer the no-purchase alternative to any product type). We refer to such preference lists as *nested lists*. Recall that the products are indexed such that $p_1\leqslant\cdots\leqslant p_n$. (Note that, with the nested lists assumption in place, the latter assumption is without loss of generality: if, for some $i_1<i_2$ we have $p_{i_1}>p_{i_2}$, there is no motivation to stock i_2 at all, and this product type can therefore be eliminated.) Let α_i denote the probability that the preference list picked from this distribution contains the product i, i.e.,

$$\alpha_i=\sum_{\ell=i}^n \Pr[L=(1,\ldots,\ell)].$$

It is easy to verify that $\alpha_1\geqslant\cdots\geqslant\alpha_n$. In §4, we show that our results extend to a more general setting, in which customer preferences are affected by both quality and price. In particular, the product types are partitioned into quality categories. Customer preferences are characterized by a given quality category, whereas within a given category the preference lists are nested.

Assumption 2: *IFR.* We assume that the distribution of the number of customers, M, has an increasing failure rate (IFR). An integer-valued random variable X is said to be IFR if $\Pr[X=k]/\Pr[X\geqslant k]$ is nondecreasing over the integer domain. It can be proven (see, for instance, Chapter 1 of Shaked and Shanthikumar 1994) that this definition is equivalent to requiring that the sequence of random variables $[X-k\mid X\geqslant k]_{k\in\mathbb{Z}}$ is stochastically nonincreasing in k. For definitions of stochastic order and stochastic monotonicity, see Shaked and Shanthikumar (1994) and Definition EC.1 in Section EC.1.1. IFR distributions capture many of the commonly studied distributions in the operations management and operations research literature. They include among many others the normal, uniform, exponential, geometric, Poisson, and even beta distributions for certain parameters.

Assumption 3: *Revenue evaluation.* Given an inventory vector $(y_1,\ldots,y_n)$, there is a polynomial-time procedure for computing $\mathrm{E}[R_M(y_1,\ldots,y_n)]$, possibly up to a multiplicative error of $(1-\epsilon)$.

In Section EC.1.3 of the online appendix, we show that when the distribution of the number of customers M has finite support, say on $\{0,\ldots,T\}$, then the expected revenue of any given vector $(y_1,\ldots,y_n)$ can be computed (exactly) by means of dynamic programming in time $O(CT^2)$. We also explain, in the general case, how to efficiently make the support of M finite, at the cost of ϵ-approximating $\mathrm{E}[R_M(y_1,\ldots,y_n)]$ instead of exactly computing this quantity.

3.1. An Overview of the Analysis

For ease of exposition, we first provide a high-level overview of the PTAS and its worst-case analysis. We show that, for any accuracy level $0<\epsilon\leqslant 0.5$, there exists a solution that stocks only a subset of product types of size $O((1/\epsilon)\cdot\log(1/\epsilon))$, obtaining at least $(1-\epsilon)$ fraction of the optimal revenue. Note that the size of this product set depends on the accuracy level ϵ but not on any other parameter of the problem. In particular, we establish the following theorem.

Theorem 1. *For any accuracy level $0 < \epsilon \leqslant 0.5$, there is an inventory vector $(y_1, \ldots, y_n)$ that satisfies the following properties*:

1. *The total capacity is at most C, that is, $\sum_{i=1}^{n} y_i \leqslant C$.*
2. *The number of nonzero coordinates in $(y_1, \ldots, y_n)$ is only $O((1/\epsilon) \cdot \log(1/\epsilon))$.*
3. $\mathrm{E}[R_M(y_1, \ldots, y_n)] \geqslant (1-\epsilon)\mathrm{OPT}$.

Furthermore, our proof gives a constructive method to efficiently identify the corresponding subset of $O((1/\epsilon) \cdot \log(1/\epsilon))$ product types. As a result, one can naively enumerate all possible solutions consisting of these product types and take the best among those solutions (as mentioned in Assumption 3, the expected revenue of any given solution can be evaluated efficiently). The overall number of such solutions is $O(C^{O((1/\epsilon)\cdot\log(1/\epsilon))})$ since there are a total of $C+1$ possible choices for the number of units to stock for each of the $O((1/\epsilon) \cdot \log(1/\epsilon))$ product types. However, this is only pseudo polynomial in the size of the input. As explained in §3.5, by employing suitable discretization, this can be improved to enumerating only $O((\log C)^{O((1/\epsilon)\cdot\log(1/\epsilon))})$, which is indeed polynomial in the input size.

To obtain a high-revenue subset of product types of size $O((1/\epsilon) \cdot \log(1/\epsilon))$, as described in Theorem 1, we partition the product types into *frequent* and *rare* product types. For any product type i, let X_i denote the random number of customers whose preference list contains product type i.

Definition 1. A product type i is called *frequent* if the expected number of units sold when C units of product type i are stocked is at least $\epsilon^2 C$, i.e.,

$$\mathrm{E}[\min(X_i, C)] \geqslant \epsilon^2 C.$$

Otherwise, the product type i is referred to as *rare*.

Consider the following instance to illustrate Definition 1. Suppose the number of product types, $n = 2$ with $p_1 = 1$, $p_2 = 50$, the capacity $C = 5$. There are two possible preference lists $L_1 = (1)$, $L_2 = (1, 2)$ with probabilities $P(L = L_1) = 0.9$ and $P(L = L_2) = 0.01$. Let the number of customers be deterministic with $M = 10$. Also, let the accuracy level $\epsilon = 0.2$. Therefore,

$$\mathrm{E}[\min(X_1, C)] = 5 \geqslant \epsilon^2 C = 0.2, \quad \text{and}$$
$$\mathrm{E}[\min(X_2, C)] \simeq 0.099 < \epsilon^2 C.$$

Therefore, product type 1 is frequent, and product type 2 is rare. The optimal assortment contains four units of product type 1 and one unit of product type 2, i.e., $y_1^* = 4$ and $y_2^* = 1$, and

$$\mathrm{E}[R_M(y_1^*, y_2^*)] = 6.85.$$

We next describe the main ideas for our PTAS.

Main idea 1: $O((1/\epsilon) \cdot \log(1/\epsilon))$ *frequent product types are sufficient.* Since $\alpha_i \geqslant \alpha_{i'}$ whenever $i < i'$, it follows that the frequent product types must be $1, \ldots, F$, for some $0 \leqslant F \leqslant n$, whereas the rare products are $F+1, \ldots, n$. In Lemma 1 we show that, by applying appropriate truncation and discretization, it is possible to efficiently identify a small subset of frequent product types of size $O((1/\epsilon) \cdot \log(1/\epsilon))$ that obtains a fraction of at least $(1-\epsilon)^3$ of the expected revenue obtained by frequent product types in the optimal solution. (In fact, this would be true for any solution.) The proof of Lemma 1 relies only on the fact that preferences are nested (see Assumption 1).

Main idea 2: *One rare product type is sufficient.* The analysis is completed by showing that a single rare product type can be used to attain at least $(1-\epsilon)$ of the expected revenue obtained by rare product types in the optimal solution. This part of the analysis relies on several central ideas. The first idea is to establish an upper bound on the total achievable expected revenue. This upper bound is derived by considering an *uncapacitated* variant of the problem where there is no capacity constraint and one is allowed to stock any number of units. The uncapacitated variant is clearly a relaxation of the original model and provides an upper bound on the optimal revenue. In Theorem 2 we show that, in the uncapacitated variant, it is optimal to stock only one product type. Specifically, it is optimal to stock M units (the number of customers is known in advance) of the *maximal product type*, which is the product type i that maximizes the expected marginal revenue $\alpha_i p_i$ from a single customer. The uncapacitated variant is discussed in §3.2.

Observe that if one stocks C units of a given rare product type, the expected number of units sold is much smaller than the capacity, specifically, less than $\epsilon^2 C$. Intuitively, this observation implies that if one only considers rare product types, the resulting problem is "almost" uncapacitated, and hence stocking only the maximal product type among rare product types should be near optimal. However, it turns out that this intuition is incorrect in general, unless the distribution of the number of customers M satisfies certain properties. In particular, we prove in Lemmas 2, 3, and 4 that this intuition is indeed valid when the distribution of M is IFR (see Assumption 2). This implies that there exists a solution that stocks only $O((1/\epsilon) \cdot \log(1/\epsilon))$ product types and obtains a fraction of at least $(1-\epsilon)^4$ of the optimal revenue. On the other hand, we demonstrate that the latter property does not hold for general distributions of M (see Section EC.3 in the online appendix for details).

The extension of the PTAS to the more general model, with preference lists that capture both price and quality, is discussed in §4. In this case, the central idea is to employ the PTAS for nested preference lists as an auxiliary subroutine within a dynamic programming approach.

3.2. The Uncapacitated Deterministic-*M* Problem

In this section, we consider the uncapacitated variant, in which the number of customers is known in advance (i.e., $M = m$ deterministically) and there is no constraint on the number of units to be stocked. We show that, for this variant, it is optimal to stock only the product type that maximizes the marginal expected revenue, which is the expected revenue from a single customer if only one unit of the product type is stocked and only one customer arrives. The marginal expected revenue of each product type is exactly the probability that a customer is willing to purchase that product type times the respective per-unit price, i.e., $\alpha_i p_i$. We call this product type the maximal product type and denote it by i^*. That is, $i^* = \arg\max_i \alpha_i p_i$.

Theorem 2. *For any number of customers, $m \in \mathbb{Z}_+$, it is optimal to stock m units of the maximal product type i^*. The resulting optimal expected revenue is $m\alpha_{i^*} p_{i^*}$.*

Proof. First observe that since the preference lists are nested, there is no benefit in stocking more than m units. Consider now any solution $y = (y_1, \ldots, y_n)$. For each customer $j = 1, \ldots, m$, let $V_j = V_j(y)$ be the revenue generated by that customer, and let $I_j = I_j(y)$ be the cheapest product type available upon the arrival of customer j; note that the distributions V_j and I_j are solution dependent. Since there are m customers and at least m units, there will always be some product type available upon the customer arrival. Observe that $\mathrm{E}[V_j] = \mathrm{E}[\mathrm{E}[V_j \mid I_j]] = \mathrm{E}[\alpha_{I_j} p_{I_j}] \leqslant \alpha_{i^*} p_{i^*}$. It follows that the total revenue of any solution is upper bounded by $m\alpha_{i^*} p_{i^*}$. However, the solution that stocks m units of product type i^* has expected revenue $m\alpha_{i^*} p_{i^*}$, so it must be optimal. $\square$

Since the number of customers is known in advance, stocking m units of the maximal product type i^* is optimal for the uncapacitated variant, and the optimal expected revenue is $m\alpha_{i^*} p_{i^*}$. Since the uncapacitated variant is a relaxation of the capacitated problem, we conclude that the optimal expected revenue of the original capacitated model (with random M) is upper bounded by $\mathrm{E}[M]\alpha_{i^*} p_{i^*}$. Moreover, the following is an immediate interesting corollary.

Corollary 1. *Consider the capacitated model described in §2, with Assumption 1. Suppose that the capacity C is larger or equal to the maximal possible value that the number of customers M can attain. Then, it is optimal to stock C units of the maximal product type i^*.*

3.3. Frequent Product Types

Recall that a product type i is called *frequent*, if the expected number of units sold, assuming only C units of this product type are stocked, is at least $\epsilon^2 C$. That is, $\mathrm{E}[\min\{X_i, C\}] \geqslant \epsilon^2 C$, where $X_i \sim B(M, \alpha_i)$ is the number of customers willing to buy product type i. Note that, conditioning on $[M = m]$, the random variable X_i follows a binomial distribution with parameters (m, α_i). Because of the nested preferences assumption, frequent product types can be numbered as $1, \ldots, F$, and rare product types by $F + 1, \ldots, n$, for some $0 \leqslant F \leqslant n$. In the remainder of this section, we consider an optimal inventory vector,

$$(\underbrace{y_1^*, \ldots\ldots, y_F^*}_{\text{frequent}}, \underbrace{y_{F+1}^*, \ldots\ldots, y_n^*}_{\text{rare}}).$$

Lemma 1 shows that there exists a subset of frequent product types of size $O((1/\epsilon) \cdot \log(1/\epsilon))$, which obtains a fraction of at least $(1 - 3\epsilon)^3$ of the expected revenue obtained by frequent product types in the optimal solution. Furthermore, if the optimal solution stocks more than one unit of rare product types, one can ensure that a capacity of at least ϵC is allocated to rare product types (i.e., the total number of units of frequent product types does not exceed $(1 - \epsilon)C$). Intuitively, we wish to make sure that the number of units of rare product types that are being sold in expectation (at most $\epsilon^2 C$) is small relative to the capacity allocated to them (at least ϵC). This property will enable us to use the uncapacitated bound derived in §3.2.

Lemma 1. *Let $(y_1^*, \ldots, y_n^*)$ be an optimal inventory solution. For any $0 < \epsilon < 0.5$, there exists a feasible inventory vector $(y_1, \ldots, y_F)$ of frequent product types with only $O((1/\epsilon) \cdot \log(1/\epsilon))$ nonzero coordinates, such that*

1. $\mathrm{E}[R_M(y_1, \ldots, y_F, y_{F+1}^*, \ldots, y_n^*)] \geqslant (1 - 3\epsilon)^3 \cdot \mathrm{E}[R_M(y_1^*, \ldots, y_n^*)] = (1 - 3\epsilon)^3 \cdot \mathrm{OPT}$.
2. *If* $\sum_{i=F+1}^n y_i^* \geqslant 2$ *then* $\sum_{i=1}^F y_i \leqslant (1 - \epsilon)C$.

Proof. We present a constructive proof that shows how to efficiently identify the corresponding subset of frequent product types.

Phase 1: *Eliminating cheap products.* Consider the frequent product types $1, \ldots, F$, and recall that p_F is the price of the most expensive frequent product type. We begin by arguing that it is possible to discard cheap product types from $(y_1^*, \ldots, y_F^*)$, without losing too much revenue in expectation. A product type i is called *cheap* if $p_i \leqslant \epsilon^3 p_F$.

We argue that if we do not stock any cheap product type, the expected revenue reduces by at most $\epsilon\mathrm{OPT}$. To understand this claim, note that the total expected revenue from cheap product types is upper bounded by $\epsilon^3 p_F C$, since at most C units of such product types could be sold, at a price of at most $\epsilon^3 p_F$ each. On the other hand, we argue that $\mathrm{OPT} \geqslant \epsilon^2 C p_F$. Consider the solution where we stock C units of product type F. The expected revenue of this solution is given by

$$\mathrm{E}[\min\{X_F, C\}] \cdot p_F \geqslant \epsilon^2 C p_F,$$

where the inequality follows, since product type F is frequent. Therefore, $\mathrm{OPT} \geqslant \epsilon^2 C p_F$ and by not stocking cheap product types, one may lose up to $\epsilon^3 p_F C \leqslant \epsilon\mathrm{OPT}$.

Phase 2: *Picking $O((1/\epsilon) \cdot \log(1/\epsilon))$ frequent product types.* Phase 1 ensures that all remaining frequent products have per-unit price in $[\epsilon^3 p_F, p_F]$. We proceed by geometrically partitioning this interval by powers of $1 + \epsilon$ to obtain $O((1/\epsilon) \cdot \log(1/\epsilon))$ subintervals: $[\epsilon^3 p_F, (1 + \epsilon)\epsilon^3 p_F)$,

$[(1+\epsilon)\epsilon^3 p_F, (1+\epsilon)^2\epsilon^3 p_F)$, etc. We modify the optimal solution by considering each of these intervals, and the respective product types with prices within the interval, and reallocating every unit purchased to the cheapest product type that falls within that interval. For instance, if there are 10 product types, say $t, \ldots, t+9$, in the interval $[\epsilon^3 p_F, (1+\epsilon)\epsilon^3 p_F)$, with respecting inventory levels $y_t^*, \ldots, y_{t+9}^*$, then we will stock $y_t^* + \cdots + y_{t+9}^*$ units of type t (which is the cheapest one) and no units whatsoever of types $t+1, \ldots, t+9$. Because of the nested preference lists, it follows that each unit in the modified solution is now consumed with probability at least as high as before. Moreover, if and when it is indeed consumed, the resulting revenue is at least $1/(1+\epsilon) \geqslant 1-\epsilon$ times its revenue prior to this transformation, since the endpoints of each price interval differ by a factor of $1+\epsilon$. The solution after this modification has only $O((1/\epsilon)\cdot\log(1/\epsilon))$ frequent product types and obtains at least $(1-\epsilon)^2$ fraction of the optimal revenue.

Phase 3: *Transferring some capacity to rare product types.* Suppose now that after Phase 2, the capacity allocated to frequent product types is larger than $(1-\epsilon)C$ and $\sum_{i=F+1}^n y_i^* \geqslant 2$. Consider the current solution $\bar{y} = (y_1, \ldots, y_n)$ obtained from the optimal solution after the modifications in Phase 1 and 2. For each unit stocked in $\bar{y}$, compute the expected contribution to the overall revenue, that is, the probability of this unit to be consumed times its per-unit price. We then discard $\sum_{i=1}^F y_i - \lfloor (1-\epsilon)C \rfloor$ units, choosing the ones with the smallest contribution. Since the expected revenue from the remaining units can only increase after removing these units, it follows that we lose a fraction of the expected revenue that is upper bounded by

$$\frac{\sum_{i=1}^F y_i - \lceil (1-\epsilon)C \rceil + 1}{\sum_{i=1}^F y_i} \leqslant \frac{\epsilon C + 1}{(1-\epsilon)C} \leqslant \frac{\epsilon C + \epsilon C/2}{(1-\epsilon)C} = \frac{3\epsilon}{2(1-\epsilon)} \leqslant 3\epsilon.$$

The first inequality follows from $\lceil (1-\epsilon)C \rceil \leqslant \sum_{i=1}^F y_i \leqslant C$, the second inequality holds since $\epsilon C \geqslant \sum_{1=F+1}^N y_i \geqslant 2$, and the last inequality since $\epsilon < 0.5$. This concludes the proof of the lemma. $\square$

3.4. Rare Product Types

Lemma 1 implies that there exists an inventory vector $(y_1, \ldots, y_F, y_{F+1}^*, \ldots, y_n^*)$ achieving a $(1-3\epsilon)^3$ fraction of the optimal revenue, and that its nonzero components among $y_1, \ldots, y_F$ are contained in an $O((1/\epsilon)\cdot\log(1/\epsilon))$-sized subset of $1, \ldots, F$. The next issue is how to complete each such combination of frequent product types by augmenting it with rare product types. Clearly, if one considers a combination such that $\sum_{i=1}^F y_i = C-1$, it is straightforward to compute the single rare product type that should be stocked to maximize the overall expected revenue simply by enumerating all possibilities. When $\sum_{i=1}^F y_i < C-1$, by Lemma 1 we can assume that $\sum_{i=1}^F y_i \leqslant (1-\epsilon)C$. In other words, there is a capacity of at least ϵC to stock units of rare product types. In the next section, we show that in fact one can use the entire residual capacity to stock only one rare product type, and obtain a $(1-\epsilon)$ fraction of the optimal expected revenue. This result is stated in the following lemma.

Lemma 2. *Consider any inventory vector $(y_1, \ldots, y_F, y_{F+1}^*, \ldots, y_n^*)$ satisfying $\sum_{i=1}^F y_i \leqslant (1-\epsilon)C$. Let i^* be the product type that maximizes $\alpha_i p_i$ among all rare product types, i.e., $i^* = \arg\max_{i\geqslant F+1} \alpha_i p_i$. Then,*

$$\mathrm{E}\Bigg[R_M\Bigg(y_1, \ldots, y_F, \underbrace{0, \ldots, 0, C - \sum_{i=1}^F y_i, 0, \ldots, 0}_{only\ i^*}\Bigg)\Bigg] \geqslant (1-\epsilon)\cdot \mathrm{E}[R_M(y_1, \ldots, y_F, y_{F+1}^*, \ldots, y_n^*)].$$

In fact, we will show that a fraction of at least $(1-\epsilon)$ of the total expected revenue from the rare product types, in any solution in which frequent product types are allocated a total capacity of at most $(1-\epsilon)C$, can be obtained by stocking only product type i^*. We start by introducing some notation:

- Let R_{I} and R_{II} be the random revenues obtained from rare product types in the inventory vectors $(y_1, \ldots, y_F, y_{F+1}^*, \ldots, y_n^*)$ and $(y_1, \ldots, y_F, 0, \ldots, 0, C - \sum_{i=1}^F y_i, 0, \ldots, 0)$, respectively.
- Let Z_{i^*} be the random number of customers with a preference list containing product type i^*, arriving after all units of frequent product types have been consumed; let $Z_{i^*} = 0$ if the frequent product types were not fully consumed.
- Let $\mathcal{A}$ denote the event "all units of frequent product are consumed."

Clearly, $\mathrm{E}[R_{\mathrm{I}}] = \Pr[\mathcal{A}]\cdot \mathrm{E}[R_{\mathrm{I}} \mid \mathcal{A}]$ and $\mathrm{E}[R_{\mathrm{II}}] = \Pr[\mathcal{A}]\cdot \mathrm{E}[R_{\mathrm{II}} \mid \mathcal{A}]$. Therefore, to prove Lemma 2, it is sufficient to prove that $\mathrm{E}[R_{\mathrm{II}} \mid \mathcal{A}] \geqslant (1-\epsilon)\mathrm{E}[R_{\mathrm{I}} \mid \mathcal{A}]$. The proof relies on two properties of the random variable Z_{i^*}. Specifically, we show that two properties of the random variable M, the total number of customers, are preserved in $[Z_{i^*}|\mathcal{A}]$. In Lemma 3 it is shown that even conditioning on the event that all units of frequent product types are consumed (i.e., the event $\mathcal{A}$), then product type i^* can still be considered as a rare product type. In Lemma 4, it is proven that, like M, the distribution of $[Z_{i^*} \mid \mathcal{A}]$ is also IFR. We proceed by stating Lemmas 3 and 4; the corresponding proofs are deferred, and we first show that these lemmas can be used to establish Lemma 2.

Lemma 3. *Let us assume that Z_{i^*} is defined as above. Then $\mathrm{E}[\min\{Z_{i^*}, C - \sum_{i=1}^F y_i\} \mid \mathcal{A}] \leqslant \epsilon^2 C$.*

Lemma 4. *The distribution of $[Z_{i^*}|\mathcal{A}]$ is IFR. In particular, $\mathrm{E}[Z_{i^*} - k \mid Z_{i^*} \geqslant k, \mathcal{A}] \leqslant \mathrm{E}[Z_{i^*} \mid \mathcal{A}]$ for every integer $k \geqslant 0$.*

Proof of Lemma 2. We show that Lemmas 3 and 4 imply that $\mathrm{E}[R_{\mathrm{II}} \mid \mathcal{A}] \geqslant (1-\epsilon)\mathrm{E}[R_{\mathrm{I}} \mid \mathcal{A}]$, from which the proof of Lemma 2 follows immediately. Note that

$$\begin{aligned}
&\mathrm{E}[Z_{i^*} \mid \mathcal{A}] \\
&= \mathrm{E}\left[\min\left\{Z_{i^*}, C - \sum_{i=1}^{F} y_i\right\} + \left[Z_{i^*} - \left(C - \sum_{i=1}^{F} y_i\right)\right]^+ \,\middle|\, \mathcal{A}\right] \qquad (1)\\
&= \mathrm{E}\left[\min\left\{Z_{i^*}, C - \sum_{i=1}^{F} y_i\right\} \,\middle|\, \mathcal{A}\right] \\
&\qquad + \Pr\left[Z_{i^*} \geqslant C - \sum_{i=1}^{F} y_i \,\middle|\, \mathcal{A}\right] \\
&\qquad \cdot \mathrm{E}\left[Z_{i^*} - \left(C - \sum_{i=1}^{F} y_i\right) \,\middle|\, Z_{i^*} \geqslant C - \sum_{i=1}^{F} y_i, \mathcal{A}\right] \\
&\leqslant \mathrm{E}\left[\min\left\{Z_{i^*}, C - \sum_{i=1}^{F} y_i\right\} \,\middle|\, \mathcal{A}\right] + \epsilon \cdot \mathrm{E}[Z_{i^*} \mid \mathcal{A}]
\end{aligned}$$

The first inequality holds, since by Lemma 4 we have $\mathrm{E}[Z_{i^*} - (C - \sum_{i=1}^{F} y_i) \mid Z_{i^*} \geqslant C - \sum_{i=1}^{F} y_i, \mathcal{A}] \leqslant \mathrm{E}[Z_{i^*} \mid \mathcal{A}]$ and by Lemma 3 we have $\Pr[Z_{i^*} \geqslant C - \sum_{i=1}^{F} y_i \mid \mathcal{A}] \leqslant \epsilon$, since

$$\begin{aligned}
\epsilon^2 C &\geqslant \mathrm{E}\left[\min\left\{Z_{i^*}, C - \sum_{i=1}^{F} y_i\right\} \,\middle|\, \mathcal{A}\right] \\
&\geqslant \Pr\left[Z_{i^*} \geqslant C - \sum_{i=1}^{F} y_i \,\middle|\, \mathcal{A}\right] \cdot \left(C - \sum_{i=1}^{F} y_i\right) \\
&\geqslant \epsilon C \cdot \Pr\left[Z_{i^*} \geqslant C - \sum_{i=1}^{F} y_i \,\middle|\, \mathcal{A}\right].
\end{aligned}$$

The last inequality follows from the assumption $\sum_{i=1}^{F} y_i \leqslant (1-\epsilon)C$. By rearranging Inequality (1), it follows that $\mathrm{E}[\min\{Z_{i^*}, C - \sum_{i=1}^{F} y_i\} \mid \mathcal{A}] \geqslant (1-\epsilon) \cdot \mathrm{E}[Z_{i^*} \mid \mathcal{A}]$. We conclude the proof by noting that

$$\begin{aligned}
\mathrm{E}[R_{\mathrm{II}} \mid \mathcal{A}] &= \mathrm{E}\left[\min\left\{Z_{i^*}, C - \sum_{i=1}^{F} y_i\right\} \,\middle|\, \mathcal{A}\right] \cdot p_{i^*} \\
&\geqslant (1-\epsilon) \cdot \mathrm{E}[Z_{i^*} \mid \mathcal{A}] \cdot p_{i^*} \geqslant (1-\epsilon) \cdot \mathrm{E}[R_{\mathrm{I}} \mid \mathcal{A}],
\end{aligned}$$

where the last inequality follows from the upper bound derived by the uncapacitated variant discussed in §3.2. □

3.5. Algorithm Running Time

We have provided a constructive proof of Theorem 1. In particular, one can enumerate $O(C^{O((1/\epsilon)\cdot\log(1/\epsilon))})$ solutions to obtain $(1-3\epsilon)^4$ fraction of the optimal expected revenue. However, this is only pseudo-polynomial in the size of the input. In fact, one can improve the running time to $O((\log C)^{O((1/\epsilon)\cdot\log(1/\epsilon))})$, which is polynomial in the input size. For this purpose, let $S = \{i_1, \ldots, i_K\}$ be the subset of $O((1/\epsilon)\cdot\log(1/\epsilon))$ product types identified by our algorithm, and let $n_k^* \in \{0, \ldots, C\}$ be the number of units of product type i_k in the best solution that stocks only $i_1, \ldots, i_K$. For every $1 \leqslant k \leqslant K$, we consider the following $O(\log C)$ choices for the number of units to stock from product type i_k:

$$\{0\} \cup \{\lfloor (1+\epsilon)^{\ell} \rfloor : 0 \leqslant \ell \leqslant \lfloor \log_{(1+\epsilon)} C \rfloor\}.$$

Clearly, when $n_k^* > 0$, this value is located between two consecutive exponents of $1+\epsilon$, say $(1+\epsilon)^{\ell_k} \leqslant n_k^* < (1+\epsilon)^{\ell_k+1}$. If we stock $\lfloor (1+\epsilon)^{\ell_k} \rfloor$ units of product type i_k instead of n_k^*, we obtain a fraction of at least $1/(1+\epsilon) \geqslant 1-\epsilon$ from the total expected revenue of each product type. It follows that with running time polynomial is the size of the input, one can obtain at least $(1-3\epsilon)^5$ of the optimal expected revenue.

Note that in Theorem 1, we claim that for any $0 \leqslant \epsilon \leqslant 0.5$, there is a sparse assortment with only $O((1/\epsilon)\cdot\log(1/\epsilon))$ product types that obtains at least $(1-\epsilon)$ fraction of the optimal expected revenue. However, we show instead that the revenue of the approximate sparse assortment computed by our PTAS is at least $(1-3\epsilon)^5$ of the optimal expected revenue. To reconcile this and obtain a solution that guarantees at least $(1-\epsilon)$ of the optimal expected revenue, we can run the algorithm with accuracy level

$$\hat{\epsilon} = \frac{1}{3}(1-(1-\epsilon)^{1/5}) \approx \frac{\epsilon}{15}.$$

4. Extension to Quality Categories

In this section, we extend the PTAS presented above to a model in which customers differentiate between product types not only based on price but also based on quality. This model is richer and captures additional important practical settings. The details of the models are as follows.

Model description. As before, we assume that product types are numbered such that the per-unit selling price is monotone nondecreasing, that is, $p_1 \leqslant \cdots \leqslant p_n$. In addition, product types are partitioned into *quality categories*. Specifically, let $1 = i_1 < i_2 < \cdots < i_k < i_{k+1} = n+1$, define k quality categories $[i_l, i_{l+1})$, $l = 1, \ldots, k$, where the quality increases in l (i.e., in the price range). For each category $[i_l, i_{l+1})$, the product types $i_l, i_l+1, \ldots, i_{l+1}-1$ are of similar quality and are differentiated only by price. Each preference list of an arriving customer consists of some quality category, say $[i_l, i_{l+1})$, and a price threshold $p < p_{i_{l+1}}$. Thus, such customer is willing to buy product types within $[i_l, i_{l+1})$, with preference to the cheapest available product type available within this category up to price threshold p. The assumption is that the price ranges of different quality categories are well separated.

We have the following theorem.

Theorem 3. *There is a PTAS for the model where the preference lists are based on quality categories and each preference list is nested within exactly one quality category.*

If we know the capacity constraint for each quality category, then we can compute a near-optimal allocation within each quality category separately using the PTAS discussed in §3. However, we do not know the capacity allocation for each quality category separately. Instead, we have a constraint of C on the total number of units across product types of all quality categories. We show that one can find near-optimal capacity allocation to different quality categories using the following dynamic program: Let $f_k(t)$ be the optimal expected revenue if $t \in \{1, \ldots, C\}$ units are stocked in quality categories $\{k, k+1, \ldots, l\}$ and let $\mu_k(t')$ be the optimal expected revenue from units in quality category k if a total of t' units are allocated to product types within the quality category $[i_k, i_{k+1})$. Note that we can compute a $(1-\epsilon)$-approximation for $\mu_k(t)$ for all $k \in \{1, 2, \ldots, l\}$, $t \in \{1, \ldots, C\}$ using the PTAS described earlier. Now for any $k < l$,

$$f_k(t) = \max_{0 \leqslant t' \leqslant t} (\mu_k(t') + f_{k+1}(t - t')).$$

Using the above dynamic program, we can obtain a PTAS for $f_1(C)$. Note that the above dynamic program is only pseudo-polynomial because of its linear dependency on C. However, we can improve the running time to polynomial in the input size by discretizing the state space of the dynamic program. We sketch the details in Section EC.1.4 of the online appendix.

5. Computational Experiments

In this section, we present a computational study that is meant to evaluate how the approximation scheme proposed in §3 performs in practice. For this purpose, we focused on two main questions: (1) How does the suggested method behave for the original problem studied? (2) How does this method behave in more general choice models, and how does it compare against existing algorithms?

Initially, we study the performance of our PTAS for the nested preference list model as a function of accuracy level and problem size, and in particular, we consider the dynamic capacitated assortment optimization problem with nested preference lists. Here, performance is measured in terms of both objective value and running times, as we vary the accuracy level and size of the problem, including number of products and number of customers. These results are also compared with respect to a sample path based upper bound.

In addition, we study the performance of our algorithm for more general choice models that do not necessarily satisfy the nested preference list assumption. Even though the analysis of our PTAS makes use of this assumption, we show that the algorithm itself can be easily adapted to more general choice models. We consider the case where customer preferences follow a MNL choice model, and we compare the performance of our algorithm with known heuristics for the dynamic assortment optimization problem, including a variant of the stochastic gradient based algorithm of Mahajan and van Ryzin (2001). Whereas we require the nested preference list assumption for the theoretical performance guarantee for our algorithm, the computational results below show that this algorithm performs better than other known heuristics for the MNL choice model.

5.1. Performance for Nested Preference Lists

In what follows, we study the performance of our PTAS as a function of accuracy level ϵ and problem size, including the number of products and capacity bound.

Experimental setup. Let us first describe the setup and parameter choices for our computational experiments. We use the following values for the base utilities, U_j: $U_j = 2^{\rho(n-j+1)}$ for $j = 1, \ldots, n$ where $\rho = 0.15$. For each customer, U_0 is random distributed uniformly in $[0, 2^{\rho(n+1)}]$. We can compute α_j, $j = 1, \ldots, n$ as follows.

$$\alpha_j = \Pr[U_j \geqslant U_0] = \frac{1}{2^{\rho \cdot j}}. \tag{2}$$

Each customer i samples the random utility, $U_{i0} \in [0, 2^{\rho(n+1)}]$, and buys the highest utility product (with utility at least U_{i0}) that is available. If none of the products with utility at least U_{i0} is available, this customer leaves without purchasing. We use the following randomly generated values for the product prices: $p_0 = 0$, $p_1 = c$, and for any $j > 1$,

$$p_j = \begin{cases} p_{j-1} + \Delta & \text{w.p. } 1-\rho \\ 2p_{j-1} & \text{w.p. } \rho \end{cases}$$

where Δ is a constant. We use $c = 10$ and $\Delta = 1$ in our computational experiments. We assume that the number of customers, $M = m$, is deterministic for the experiments and varies between 40 and 200. The number of product types, n, varies in $\{10, 12, 15, 20\}$, and the capacity bound C on the total number of units of all product types varies between 15 and 100.

We test the performance of our PTAS both with respect to the objective value and running time for various combinations of the accuracy level and problem size. By Theorem 1 and its proof, we know that for any $\epsilon > 0$, the assortment obtained by our PTAS has an expected revenue of at least $(1-3\epsilon)^5 \cdot$ OPT. We conduct experiments with $\epsilon \in \{0.05, 0.1, 0.2\}$ and compare the objective values with a sample path based upper bound (see below) as well as the running times. We note that the objective value of the solution computed by our PTAS for each value of ϵ is computed by averaging the profit of the given solution over a set of i.i.d. sample paths. Therefore, the computed objective value is an unbiased estimator of the expected revenue of the given solution.

Sample path based upper bound. Since for sufficiently large instances the value of OPT cannot be computed in reasonable time, for any instance with number of product types, n, number of customers, m, and capacity bound, C (with $C < m$), we compute a sample path based upper bound as follows. We sample for each customer $1 \leqslant i \leqslant m$ the utility of product type 0, U_{i0}, as an i.i.d. uniform in $[0, 2^{\rho(n+1)}]$. We compute the maximum possible profit on the sample path using the following dynamic programming algorithm. Let $F(i, j, k)$ be the maximum profit that can be obtained from customer k onwards by stocking at most j

units of product types $\{i, i+1, \ldots, n\}$. For any customer $k = 1, \ldots, m$, let

$$\tau_k = \arg\max\{\ell \mid U_{k\ell} > U_{k0}\},$$

i.e., τ_k is the most expensive product type that customer k prefers to not purchasing at all. Therefore, for all $i \in [n]$ and $k \in [m]$, $F(i, 0, k) = 0$. For all $i \in [n]$, $j \in [C]$,

$$F(i, j, m) = \begin{cases} r_{\tau_m} & \text{if } \tau_m \geqslant i \\ 0 & \text{otherwise.} \end{cases}$$

For any $k < m$, $i \in [n]$ and $j \in [C]$,

$$F(i,j,k) = \begin{cases} F(i,j,k+1) & \text{if } \tau_k < i \\ \max\left(F(\tau_k+1, j, k+1), \max\limits_{\ell=i}^{\tau_k}(r_\ell + F(\ell, j, k))\right) & \\ & \text{otherwise.} \end{cases}$$

If $\tau_k < i$, then customer k does not buy any product out of $i, \ldots, n$, and the maximum profit possible is $F(i, j, k+1)$. Otherwise, we pick the maximum between the two options of not selling any product to customer k or selling some product $\ell \in \{i, \ldots, \tau_k\}$. The maximum profit on a given sample path is $F(1, C, 1)$. We obtain an upper bound by averaging the maximum profit obtained over the same set of sample paths that are used to compute the objective value of the solutions computed by our PTAS. Note that in computing the upper bound, we assume full information before making the assortment and inventory decisions.

Table 1 summarizes our experimental results. For each instance and each value of $\epsilon \in \{0.2, 0.1, 0.05\}$, we report the objective value of the solution obtained, the running time in seconds, and the number of product types stocked in the solution. We use a time limit of 7,200 seconds; if the algorithm does not terminate before that, we report the best solution found within the time limit. We also report the sample path based upper bound in the last column.

These results show that, for all instances in our experiments and for all values of ϵ, the objective value computed by our PTAS is at least 70% of the sample path based upper bound. Moreover, for many instances, the objective value is at least 90% of the upper bound. This shows that our PTAS performs significantly better than the theoretical performance bound on the test instances. Recall that, for any choice of $\epsilon > 0$, we obtain an expected revenue of at least $(1-3\epsilon)^5 \cdot \text{OPT}$. For $\epsilon = 0.2$, 0.1, and 0.05, $(1-3\epsilon)^5 = 0.01$, 0.17, and 0.44, respectively. Therefore, the performance demonstrated in the numerical experiments is significantly better than the theoretical guarantees. We note that we only compare to an upper bound. The true performance is possibly even better if we compare to the true optimal. However, computing the optimal solution is intractable even for small instances. We are able to compute the optimal solution for only three of the above instances (namely ($n = 10$, $C = 15$), ($n = 10$, $C = 20$), and ($n = 15$, $C = 15$)) within a time limit of 24 hours for each instance. For these instances, the optimal value is about 0.9 of the upper bound on average.

As expected, the objective value of the solution obtained by our PTAS improves when we decrease the value of ϵ. For all instances except one, $\epsilon = 0.05$ corresponds to the best solution found. However, the improvement in objective value from $\epsilon = 0.2$ to $\epsilon = 0.05$ is relatively small; of the order of 1%. For the last instance with $n = 20$, $m = 50$, $C = 25$, the solution for $\epsilon = 0.05$ is worse than those for $\epsilon = 0.2, 0.1$. In this case, the algorithm with $\epsilon = 0.05$ reaches the time limit of 7,200 seconds and therefore is not able to complete the enumeration over all solutions for the frequent product types. We note that although the performance usually improves as we decrease the accuracy level, ϵ, it is not always necessary even if we complete the enumeration step for the smaller value of ϵ. This phenomenon occurs since the collection of products remaining after eliminating cheap ones (see Definition 1 and proof of Lemma 1) depends heavily on ϵ. In particular, for different values of ϵ, the subsets of frequent products picked could be very different. Consequently, for specific instances, we might actually obtain greater expected revenue with ϵ_1 than with some $\epsilon_2 < \epsilon_1$. That being said, the worst-case performance guarantee of our algorithm is at

Table 1. Performance of PTAS (objective value, running time, and number of products stocked) for different values of ϵ.

			$\epsilon = 0.2$			$\epsilon = 0.1$			$\epsilon = 0.05$			
n	m	C	Obj.	Time (s)	#Prod	Obj.	Time (s)	#Prod	Obj.	Time (s)	#Prod	UB
10	40	15	235.2	0.3	2	237.3	0.4	3	**238.2**	18.2	4	270.8
10	40	20	280.3	0.3	2	281.5	0.6	4	**283.4**	121.3	5	324.5
10	40	25	317.6	0.3	3	319.5	1.1	2	**319.6**	570.3	2	365.1
10	50	30	387.8	0.3	3	390.3	2	3	**391.3**	2,184.2	4	442.8
10	100	40	630.1	0.4	2	632.8	5.3	3	**635.9**	7,200*	5	693.5
10	100	50	719.5	0.4	2	722.8	14.5	3	**725.3**	7,200*	6	787.8
10	150	80	1,130.6	0.7	2	1,131.9	111.8	3	**1,135.4**	7,200*	4	1,209.8
10	200	100	1,456.1	1	2	1,470.7	291.6	4	**1,471.3**	7,200*	5	1,551.1
12	100	50	721.2	1.5	3	723.3	169.6	4	**725.2**	7,200*	6	791.9
15	40	15	236.9	0.3	3	237.8	2.6	4	**238.4**	322.61	5	347.3
15	40	20	280.3	0.4	2	281.5	10.6	4	**283.4**	3,681.7	5	380.2
20	50	25	375.7	1.4	3	**375.8**	171.4	4	362.9	7,200*	4	471.9

Note. Here, * in the running time denotes that the algorithm reached the time limit.

least $(1-O(\epsilon))$OPT, which clearly improves as the value of ϵ decreases.

The running time of our PTAS increases significantly as we decrease the value of ϵ from 0.2 to 0.05. For $\epsilon = 0.2$, all the instances complete in less than 1.5 seconds, whereas for $\epsilon = 0.05$, six out of 12 instances reach the time limit of 7,200 seconds. The number of product types in the solution also increases typically as we decrease the value of ϵ. For $\epsilon = 0.2$, the number of product types is at most three for all instances, whereas for $\epsilon = 0.05$, there are several instances where the solution stocks six product types. This can be explained by noting that the number of frequent product types increases (by definition), and consequently the algorithm has to enumerate over a significantly larger number of solutions. Therefore, the running time also increases significantly as we decrease the value of ϵ.

5.2. Performance for General Choice Models

In this section, we consider the dynamic assortment planning problem for general random utility based choice models. The class of models arising from a random utility model is quite general and includes the multinomial logit (MNL), probit, nested logit (NL), and mixture of MNL (MMNL) models. In the random utility based choice model, the utility of product j for each customer i, $U_{ij} = U_j + \epsilon_{ij}$ where U_j depends on the attributes of product j and ϵ_{ij} is a random idiosyncratic component of the utility. The preference list for a customer is given by the decreasing order of utilities of products. The nested preference list model can be thought of as the following random utility based model: suppose $\epsilon_{ij} = 0$ for all $j = 1, \ldots, n$, ϵ_{i0} is random and the base utilities of products are $U_1 \geqslant U_2 \geqslant \cdots \geqslant U_n$. This leads to nested preference lists of the form $(1, \ldots, j)$, $j = 0, \ldots, n$. For the MNL model, the random idiosyncratic components ϵ_{ij} are i.i.d. distributed according to a standard Weibull distribution.

Adapting the algorithm. Even though the approximation scheme in §3 was originally presented for nested preference lists, we proceed by showing how to easily adapt the algorithm for general choice models. Consider the following model, where for any product j and customer i, the utility is $U_{ij} = U_j + \epsilon_{ij}$, where ϵ_{ij} is distributed according to a known distribution f_j. Note that ϵ_{ij} is i.i.d. for each customer i. We can adapt our algorithm for the general choice model as follows. For any product type j, let α_j denote the probability that a customer is willing to buy product type j. In the nested preference list model, α_j is equal to the probability that the preference list contains product type j. For a general choice model, we can compute α_j as the probability that the utility of product type j is more than the utility of product 0, i.e.,

$$\alpha_j = \Pr[U_j + \epsilon_{ij} \geqslant U_0 + \epsilon_{i0}], \tag{3}$$

which can be computed when $f_0, \ldots, f_n$ are known. Now, for any product type $j = 1, \ldots, n$, let X_j denote the number of customers whose preference list contains j before 0. Recall that a product type is frequent if $\mathrm{E}[\min\{X_j, C\}] \geqslant \epsilon^2 C$ and rare otherwise. Also, a product type j is cheap if $p_j \leqslant \epsilon^3 p_F$, where p_F is the price of the most expensive frequent product. We can classify the products as frequent or rare for any general choice model. Algorithm 1 describes the procedure for general choice models, which proceeds along the lines similar to the proof of Lemma 1. We note that the theoretical performance bound given in Theorem 1 does not hold for the adapted algorithm in the case of general choice models.

Algorithm 1 (Adapted algorithm for general choice models)
1: For each product type j, compute
   $\alpha_j = \Pr[U_j + \epsilon_{ij} \geqslant \epsilon_{i0}]$.
2: Eliminate cheap product types with price smaller than
   $\epsilon^3 p_F$.
3: Allocate ϵC capacity to the product $\arg\max_j \alpha_j p_j$.
4: Allocate remaining capacity as follows.
   (a) Geometrically partition the range $[\epsilon^3 p_F, p_F]$.
   (b) Consider $O((1/\epsilon)\cdot\log(1/\epsilon))$ product types at the boundary of the geometric partition.
   (c) Select best solution from all possible inventory levels for $O((1/\epsilon)\cdot\log(1/\epsilon))$ types, such that the total capacity is at most $(1-\epsilon)C$.

Multinomial logit (MNL) choice model. We test the performance of our algorithm for the case of multinomial logit model. Therefore, we assume that ϵ_{ij}, for all $j = 0, \ldots, n$, are distributed according to the standard Weibull distribution. We use the following values for the base utilities, U_j: $U_0 = 1$ and $U_j = n - j + 1$ for $j = 1, \ldots, n$, and the following values for profit, p_j: $p_0 = 0$ and for any $j > 1$,

$$p_j = \begin{cases} p_{j-1} + \Delta & \text{w.p. } 1-\rho \\ 2 \cdot p_{j-1} & \text{w.p. } \rho. \end{cases}$$

We use $c = 10$, $\Delta = 1$ and $\rho = 0.15$ as before in our computational experiments. Given that the random utility is generated from a standard Weibull distribution, we can compute the values of α_j, $j = 1, \ldots, n$ as follows:

$$\alpha_j = \frac{1}{1 + e^{U_0 - U_j}}. \tag{4}$$

We compare the performance of our adapted Algorithm 1 with several other heuristics described below. As before, we assume that the number of customers, $M = m$, is deterministic for the experiments and varies between 40 and 200. The number of product types, $n \in \{10, 12, 15, 20\}$ and the capacity bound, C, on the total number of units of all product types varies between 15 and 100. We set a time limit of 7,200 seconds and report the best solution found within that time frame for all the heuristics including our adapted PTAS.

5.3. Other Heuristics

We compare the performance of our adapted PTAS with the following heuristics for the dynamic assortment planning problem: (i) local search; (ii) projected gradient descent; and (iii) static MNL based heuristic. We describe the corresponding algorithms below.

Local search. We consider the following local search heuristic, where in each step the algorithm either increases the inventory of some product type if the total capacity is less than C or finds a pair of product types, such that increasing the inventory of one and decreasing the inventory of the other improves the expected revenue. The algorithm terminates with a local maximum solution when no such local swap is possible. The specifics of this algorithm are given in Algorithm 2.

Algorithm 2 (Local search algorithm (LS))
1: Initial feasible solution: $y_j = C$ if product type j maximizes $\alpha_j p_j$ and 0 otherwise.
2: While $(y_1, \ldots, y_n)$ is not local optimum:
  (a) Find product types i, j such that moving one unit from i to j increases the expected revenue.
  (b) The new solution is obtained by $y_i \leftarrow y_i - 1$ and $y_j \leftarrow y_j + 1$.

We denote the expected revenue computed by the above algorithm as LS. Although we used a time limit of 7,200 seconds, this algorithm terminated in less than 1 minute, and converged to a local maximum for all instances in our experiments.

Projected gradient descent. We consider an adaptation of the stochastic gradient descent algorithm of Mahajan and van Ryzin (2001) for the dynamic assortment planning problem. Originally, their paper considered a per-unit selling price and per-unit cost for each product type, and the goal was to maximize the expected profit. There were no constraints on the inventory in their model. In our model, there is a capacity constraint on the total number of the units that can be stocked, and the goal is to maximize the expected revenue. Moreover, the revenue function is defined only for integer inventory vectors.

To implement a gradient descent algorithm for our problem, we define a continuous extension of the revenue function following the Lovász extension of a discrete function. For the revenue function $f: \mathbb{Z}_+^n \to \mathbb{R}_+$, we define the continuous extension $\hat{f}: \mathbb{R}_+^n \to \mathbb{R}_+$ as follows.

$$\hat{f}(y_1, \ldots, y_n) = f(\lfloor y_1 \rfloor, \ldots, \lfloor y_n \rfloor) + \sum_{i=1}^{n} \left[f\left(\sum_{j=1}^{i} 1_{\pi(j)}\right) - f\left(\sum_{j=1}^{i-1} 1_{\pi(j)}\right) \right] \cdot (\lceil y_{\pi(i)} \rceil - \lfloor y_{\pi(i)} \rfloor),$$

where $\pi \in \Pi(\{1, \ldots, n\})$ is a permutation such that

$$y_{\pi(1)} \geqslant y_{\pi(2)} \geqslant \cdots \geqslant y_{\pi(n)}.$$

We implement the gradient descent algorithm for the continuous extension $\hat{f}$. The gradient of $\hat{f}$ can be expressed as follows.

$$\frac{\partial}{\partial y_{\pi(i)}} \hat{f} = f\left(\sum_{j=1}^{i} 1_{\pi(j)}\right) - f\left(\sum_{j=1}^{i-1} 1_{\pi(j)}\right), \quad i = 1, \ldots, n,$$

which can be computed via sampling. Since there is a capacity constraint on the total number of items stocked, after each solution update we need to project the solution to the feasible set. Therefore, in any iteration k, we update the solution as follows.

$$y^k = \frac{C \cdot (y^{k-1} + a_k \nabla \hat{f}(y^{k-1}))}{e^T (y^{k-1} + a_k \nabla \hat{f}(y^{k-1}))},$$

where e is the vector of all ones and a_k is the step size in iteration k. The algorithm terminates when we find a stationary point, i.e., $y^k = y^{k-1}$. This essentially translates to the condition

$$\nabla \hat{f}(y^k) = \tau y^k,$$

for some $\tau \in \mathbb{R}$. The complete algorithm is described in Algorithm 3.

Algorithm 3 (Projected gradient descent algorithm (PGD))
1: Initial feasible solution: $y^1 = 0$, $k = 1$.
2: Let step size $a_i = 1/i$ in iteration i.
3: While $(\|y^k - y^{k-1}\|_\infty \geqslant \epsilon)$
  (a) Compute $\nabla \hat{f}(y^k)$.
  (b) Update.
$$y^{k+1} = \frac{C \cdot (y^k + a_k \nabla \hat{f}(y^k))}{e^T (y^k + a_k \nabla \hat{f}(y^k))}.$$
  (c) Update $k \leftarrow k + 1$.
4: Return the continuous solution y^k.

Note that the solution y computed by the above gradient descent is not integral, and thus, infeasible. Also, it is not clear how to efficiently compute an integral solution whose expected revenue is close to $\hat{f}(y)$. The solution y_u where

$$y_j^u = \lceil y_j \rceil, \quad \forall\, j = 1, \ldots, n,$$

violates the capacity constraint; and solution y^l, where

$$y_j^l = \lfloor y_j \rfloor, \quad \forall\, j = 1, \ldots, n,$$

can have significantly lower expected revenue as compared to $\hat{f}(y)$. Therefore, it is possible that $\hat{f}(y)$ is higher than the optimal expected revenue for some instances.

Static MNL based heuristic. We also consider a heuristic based on solving a static assortment optimization for the MNL model, which is given in Algorithm 4. We first compute the optimal assortment, $S^* \subseteq [n]$, that maximizes the expected revenue from a single customer for the given MNL model. If we order the product types such that their profits satisfy $p_1 \geqslant p_2 \geqslant \cdots \geqslant p_n$, then the optimal assortment S^* has to be $\{1, 2, \ldots, k\}$ for some $k \in [n]$ (see Talluri and van Ryzin 2004, Gallego et al. 2004). Now for each product type $j \in S^*$, we stock units proportional to U_j. In particular, we compute the number of units, y_j, for all $j \in S^*$ as

$$y_j = \lambda_j = \min\left(\left\lceil \frac{U_j}{\sum_{i \in S^*} U_i} \cdot C \right\rceil, C - \sum_{i=1}^{j} y_i \right).$$

Algorithm 4 (Static MNL assortment based heuristic)

1: Compute the optimal static assortment, S^* for the MNL model as

$$\max_{S\subseteq[n]} \sum_{j\in S} p_j \cdot \frac{U_j}{U_0+\sum_{i\in S} U_i}.$$

We know that $S^* = \{1, \ldots, k\}$ for some $k \in [n]$ where products are ordered such that $p_1 \geqslant \cdots \geqslant p_n$.

2: For each $j = 1, \ldots, k$ $(j \in S^*)$

$$y_j = \lambda_j = \min\left(\left\lceil \frac{U_j}{\sum_{i\in S^*} U_i} \cdot C \right\rceil, C - \sum_{i=1}^{j} y_i\right).$$

3: Return $(y_1, \ldots, y_n)$.

Sample path based upper bound. For each instance, we compute the sample path based upper bound as follows. We sample utilities, U_{ij} for each customer $i = 1, \ldots, m$ and product type $j = 0, \ldots, n$. Note $U_{ij} = U_j + \epsilon_j$ where ϵ_j is i.i.d. according to the standard Weibull distribution. For each customer i, let

$$r_i = \max\{p_j \mid U_{ij} \geqslant U_{i0},\ j = 1, \ldots, n\},$$

denote the maximum profit that can be obtained from the customer. Therefore, the maximum profit on this sample path for any feasible assortment is the sum of the highest C values of $r_1, \ldots, r_m$. We obtain an upper bound by averaging the maximum profit obtained over the same set of sample paths used to compute the objective value for other heuristic solutions. Note that in computing the upper bound, we disregard the sequence of arrivals of the customers and the fact that the inventory decisions are fixed at the beginning of the planning horizon. This upper bound is more conservative than the one for the case of nested preference lists, where we use a dynamic program to compute the optimal profit under full information for each sample path. For the MNL choice model, the preference lists are not nested and a similar dynamic programming solution does not work.

5.4. Results

We describe the computational performance of our adapted Algorithm 1 and the three other heuristics. Similar to the case of nested preference lists, we first compare the performance of our adapted algorithm as a function of the accuracy level ϵ and the problem size, including number of product types, n, number of customers, m, and the capacity bound, C. We test the algorithm for three values of $\epsilon \in \{0.05, 0.1, 0.2\}$. The values of n are $\{10, 15, 20\}$, m ranges from 40 to 200, and C ranges from 15 to 100. It is important to emphasize again that for each value of $\epsilon \in \{0.05, 0.1, 0.2\}$, we discretize the profits in powers of $(1+\epsilon)$ and use ϵ to classify the product types as frequent or rare.

Table 2 summarizes the performance of our adapted Algorithm 1 for the case of MNL choice model. For each instance and each value of ϵ, we report the objective value of the solution computed by our algorithm, the running time in seconds, and the number of product types stocked in the solution. As before, we use a time limit of 7,200 seconds for each time instance and report the best solution found within that time limit. Also, as in the previous experiment, the objective value for a given solution is computed by averaging the profit over a set of i.i.d. sample paths.

Our computational results show that the running time increases significantly as we decrease the value of ϵ. For instance, when $\epsilon = 0.2$, our PTAS computes the best solution in at most 20 minutes in all cases (and significantly faster for most instances). On the other hand, when $\epsilon = 0.05$, the algorithm hits the time limit of 7,200 seconds on all instances. However, as we decrease ϵ, the performance in terms of objective value typically improves. For instance, our PTAS obtains significantly better solutions for all instances for $\epsilon = 0.1$ as compared to $\epsilon = 0.2$. However, we note that the objective value is not a monotone function of the accuracy level. For example, in some instances, our PTAS obtains an inferior solution with $\epsilon = 0.05$ as compared to $\epsilon = 0.1$. This happens since we truncate our algorithm using a time limit and do not complete the enumeration over frequent product types.

Table 2. Performance of our adapted Algorithm 1 as a function of the accuracy level and problem size.

			$\epsilon = 0.2$			$\epsilon = 0.1$			$\epsilon = 0.05$		
n	m	C	Obj.	Time (s)	#Prod	Obj.	Time (s)	#Prod	Obj.	Time (s)	#Prod
10	40	15	214.7	2	2	**240.7**	62	2	237	7,200*	3
10	40	20	277.8	4	2	**311.4**	171	2	310.2	7,200*	3
10	40	25	338	6	2	372.4	375	3	**377.2**	7,200*	4
10	50	30	407.8	10	2	454.1	913	3	**457.1**	7,200*	4
10	100	40	578.5	34	2	**648.1**	5,406	2	634.7	7,200*	3
10	100	50	701.2	53	2	**797.2**	7,200*	2	786	7,200*	3
10	150	80	1,119	197	2	**1,269.4**	7,200*	2	1,257	7,200*	3
10	200	100	1,411.2	404.5	2	**1,602.7**	7,200*	2	1,580.7	7,200*	2
12	100	50	791.1	63	2	**841**	7,200*	2	836.5	7,200*	2
15	40	15	268	20	2	292.4	372	1	**300**	7,200*	2
15	40	20	343.9	42	2	375.7	1,261	3	**391.3**	7,200*	3
20	50	25	1,060	1,040	1	1,060	7,200*	1	**1,070**	7,200*	2

Notes. The best objective value for each instance is marked in bold. Here, * denotes that the time limit is reached.

Table 3. Comparison of different heuristics with our algorithm, setting $\epsilon = 0.05$.

			PTAS, $\epsilon = 0.05$		Local search		Grad-descent		Static MNL		
n	m	C	Obj.	#Prod	Obj.	#Prod	Obj.	#Prod	Obj.	#Prod	Static UB
10	40	15	**237**	3	189.5	4	143.8	10	227.5	6	259.9
10	40	20	**310.2**	3	218.1	6	234.8	10	298	6	340.8
10	40	25	**377.2**	4	256.4	5	300.8	10	373.3	6	418.8
10	50	30	**457.1**	4	305.7	8	376.7	10	442.6	6	504.4
10	100	40	**634.7**	3	484.1	6	530.3	10	592.8	6	691.9
10	100	50	**786**	3	553.2	7	675.3	10	742.4	6	852.9
10	150	80	**1,257**	3	830.9	10	1,097.7	10	1,178.4	6	1,358.3
10	200	100	**1,580.7**	2	1,061.4	10	1,420.6	10	1,472.8	6	1,706.6
12	100	50	**836.5**	2	583.8	11	716.5	12	798.4	7	933
15	40	15	**300**	2	308.3	7	196.2	15	273.2	7	417.5
15	40	20	**391.3**	3	358	9	291.7	15	371.5	7	518.4
20	50	25	1,070	2	740	6	589.8	20	**1,089.4**	5	1,136.2

Notes. The last column presents the sample path based upper bound. The best objective value is bold for each instance.

We also emphasize that our PTAS computes sparse solutions for all instances and all values of ϵ, where we only stock a small number of product types (typically two or three in most test instances, and four in only two of the instances). This property follows from the design of our algorithm, where only solutions stocking a small number of product types are being enumerated. For the case of nested preference lists, we prove that such an algorithm guarantees a near-optimal solution. For general choice models, we are unable to provide theoretical performance bounds, but the computational experiments show that our adapted algorithm performs well as compared to other known heuristics for general choice models.

Comparison with other heuristics. In Table 3, we present the performance comparison with the heuristics described earlier: local search, projected gradient based algorithm, and static MNL assortment based algorithm. For each test instance, we report the objective value of the solution computed by each heuristic. The running time for all heuristics over all instances is at most five minutes, which is significantly lower than the running time of our adapted PTAS. Therefore, we do not report the specific running times for each instance. We also report a sample path based upper bound for each instance.

These experiments show that our adapted PTAS performs significantly better than other heuristics on all but one instance. Among the three heuristics, the static MNL assortment based heuristic performs the best. In fact, for the instance with $n = 20$, $m = 50$, $C = 25$, it performs even better than our adapted PTAS, although the difference is less than 2%. However, for all other instances, our adapted PTAS performs significantly better. It is important to note that the static MNL based heuristic is the only heuristic (among the approaches we compare) that utilizes the MNL structure of the choice model. All other methods, including our adapted PTAS, can be extended to general choice models. Therefore, it is not surprising that the static MNL assortment based heuristic performs better than local search and projected gradient descent methods.

For the local search heuristic, we start with a solution that stocks C units of the product type that maximizes the expected revenue from a single customer. We experimented with several initial solutions, but this choice always gives the best local optimum. The solution computed by the local search algorithm stocks a large number of product types and in some cases stocks all the available product types. This is in contrast to the solution computed by our adapted PTAS that stocks only a small number of product types. However, the local search heuristic performs significantly worse than our algorithm and even worse than the static MNL assortment based heuristic.

It is worth mentioning that the projected gradient based method only computes a fractional solution, i.e., the inventory levels at a local optimum are not necessarily integral. Therefore, the solution is possibly infeasible, and it is not entirely clear how to obtain a feasible solution with objective value close to the objective value at the local optimum, or if that is even possible at all.

6. Concluding Remarks

Approximability of general model. Our main results indicate that the model under consideration is indeed NP-hard for general choice models and can be efficiently approximated within any degree of accuracy for certain classes of choice models. However, even though we were able to establish a lower bound of $1 - 1/e$ on the approximability of an extremely simple model (with a single customer), it is quite possible that stronger complexity results could be obtained for the model in its utmost generality (i.e., arbitrary number of customers with arbitrary preferences). For this reason, it would be interesting to investigate such complexity issues as part of future research and possibly complement them by corresponding upper bounds through algorithmic methods.

Hardness of nested preferences. Under the technical restrictions listed in §3, where customer preferences are assumed to be nested and where the number of arriving customers is assumed to be drawn from an IFR distribution, we were able to devise a polynomial-time approximation scheme (PTAS). However, we do not know if this variant is NP-hard or can be solved in polynomial time. We pose the task of fully characterizing the hardness of nested preferences model as an interesting direction for future research. We note that even if this model can be optimally solved in polynomial time, the managerial insights from the structure of a near-optimal solution (that a constant number of product types are sufficient) obtained by our algorithm are useful. Moreover, our algorithm provides the following:

1. an immediate way to recognize the products involved;
2. an efficient enumeration method to spread the capacity C between these products. Here, only $O((\log C)^{O((1/\epsilon)\cdot\log(1/\epsilon))})$ inventory vectors need to be evaluated.

Furthermore, the computational experiments show that our PTAS performs well for significantly general choice models derived from random utility models, and thus can be used as a practical method for assortment optimization problems.

Improved running time. As demonstrated in §5, the empirical performance of our PTAS is significantly better than its worst-case approximation guarantees. Moreover, the underlying approach of distributing the overall capacity between a small number of carefully picked product types performs in practice better than several other heuristics, including local search, gradient descent, and static MNL approximation. That said, even though we enumerate over very few product types, for sufficiently small accuracy levels our algorithm suffers from impractical running times when applied to large-scale instances. It is an interesting open question to come up with provably good approximation methods that admit fast implementations at the same time.

In this context, a particularly interesting approach has recently been proposed by Segev (2015), who showed how to approximate the nested preference list model within any degree of accuracy in quasi-polynomial time without the IFR assumption. In terms of running time, this result still appears to be theoretical in nature. However, if one is willing to lose a logarithmic factor in optimality in the worst-case (but the performance could potentially be much better on average), the suggested approach can be dramatically simplified, producing an approximate inventory vector in milliseconds. We believe that further improvements along these lines, possibly resulting in constant-factor performance guarantees, could be possible.

Supplemental Material

Supplemental material to this paper is available at http://dx.doi.org/10.1287/opre.2015.1450.

Acknowledgments

The first author's research is partially supported by National Science Foundatioin [Grants CMMI-1201116 and CMMI-1351838 (CAREER Award)]. The second author's research is partially supported by National Science Foundation [Grants DMS-0732175 and CMMI-0846554 (CAREER Award)], Air Force Office of Scientific Research [Award FA9550-08-1-0369], an SMA grant, and the Solomon Buchsbaum Research Fund of MIT.

References

Anupindi R, Gupta S, Venkataramanan MA (2006) Managing variety on the retail space: Using household scanner panel data to rationalize assortments. Technical report, University of Michigan, Ann Arbor.

Bansal N, Chakrabarti A, Epstein A, Schieber B (2006) A quasi-PTAS for unsplittable flow on line graphs. *Proc. 38th Ann. ACM Sympos. Theory Comput., STOC '06* (ACM, New York), 721–729.

Cachon G, Terwiesch C, Xu Y (2005) Retail assortment planning in the presence of consumer search. *Manufacturing Service Oper. Management* 7(4):330–346.

Chan T-HH, Elbassioni KM (2011) A QPTAS for TSP with fat weakly disjoint neighborhoods in doubling metrics. *Discrete Comput. Geometry* 46(4):704–723.

Chen F, Bassok Y (2008) Substitution and variety. Working paper, Marshall School of Business, University of Southern California, Los Angeles.

de la Vega WF, Lueker GS (1981) Bin packing can be solved within 1 + epsilon in linear time. *Combinatorica* 1(4):349–355.

Fisher ML, Vaidyanathan R (2007) An algorithm and demand estimation procedure for retail assortment optimization. Manuscript.

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

Gaur V, Honhon D (2006) Assortment planning and inventory decisions under a locational choice model. *Management Sci.* 52(10):1528–1543.

Halman N, Klabjan D, Li CL, Orlin J, Simchi-Levi D (2008) Fully polynomial time approximation schemes for stochastic dynamic programs. *Proc. Nineteenth Ann. ACM-SIAM Sympos. Discrete Algorithms* (SIAM, Philadelphia), 700–709.

Halman N, Klabjan D, Mostagir M, Orlin JB, Simchi-Levi D (2009) A fully polynomial-time approximation scheme for single-item stochastic inventory control with discrete demand. *Math. Oper. Res.* 34(3): 674–685.

Har-Peled S (2011) *Geometric Approximation Algorithms*, Mathematical Surveys and Monograph, Vol. 173, chapter 23 (American Mathematical Society, Providence, RI).

Ho TH, Tang CS (1998) *Product Variety Management: Research Advances* (Springer, New York).

Honhon D, Gaur V, Seshadri S (2010) Assortment planning and inventory decisions under stockout-based substitution. *Oper. Res.* 58(5): 1364–1379.

Hopp WJ, Xu X (2005) Product line selection and pricing with modularity in design. *Manufacturing Service Oper. Management* 7(3):172–187.

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

Kok AG, Fisher ML, Vaidyanathan R (2006) Assortment planning: Review of literature and industry practice. Agrawal N, Smith SA, eds. *Retail Supply Chain Management* (Kluwer, New York), 99–154.

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

Mahajan S, van Ryzin G (2001) Stocking retail assortments under dynamic consumer substitution. *Oper. Res.* 49(3):334–351.

Nagarajan M, Rajagopalan S (2008) Inventory models for substitutable products: Optimal policies and heuristics. *Management Sci.* 54(8): 1453–1466.

Netessine S, Rudi N (2003) Centralized and competitive inventory models with demand substitution. *Oper. Res.* 51(2):329–335.

Parlar M, Goyal SK (1984) Optimal ordering decisions for two substitutable products with stochastic demands. *Oper. Res.* 21(1):1–15.

Pentico D (1974) The assortment problem with probabilistic demands. *Management Sci.* 21(3):286–290.

Ramdas K (2003) Managing product variety: An integrative review and research directions. *Production Oper. Management* 12(1):79–101.

Remy J, Steger A (2009) A quasi-polynomial time approximation scheme for minimum weight triangulation. *J. ACM* 56(3):Article No. 15.

Segev D (2015) Assortment planning with nested preferences: Dynamic programming with distributions as states? Working paper, University of Haifa, Haifa, Israel.

Shaked M, Shanthikumar JG (1994) *Stochastic Orders and Their Applications* (Academic Press, San Diego, CA).

Smith S, Agrawal N (2000) Management of multi-item retail inventory systems with demand substitution. *Oper. Res.* 48(1):50–64.

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

Van Ryzin G, Mahajan S (1999) On the relationship between inventory costs and variety benefits in retail assortments. *Management Sci.* 45(11):1496–1509.

Vineet Goyal is an associate professor in the Industrial Engineering and Operations Research Department at Columbia University. He is interested in the design of efficient and robust data-driven algorithms for large-scale dynamic optimization problems with applications in revenue management and smart grid problems. He received the NSF Faculty Early Career Development Award in 2014, Google Faculty Research Award in 2013, and the IBM Faculty Award in 2014.

Retsef Levi is the J. Spencer Standish (1945) Professor of Operations Management at the MIT Sloan School of Management. He is a member of the Operations Management Group at MIT Sloan and is affiliated with the Operations Research Center and the Computational for Design and Optimization Program. His current research is focused on the design of analytical data-driven decision support models and tools addressing complex business and system design decisions under uncertainty in areas such as health and healthcare management, supply chain, procurement and inventory management, revenue management, pricing optimization, and logistics. He is interested in the theory underlying these models and algorithms, as well as their computational and organizational applicability in practical settings. He received the NSF Faculty Early Career Development award, the 2008 INFORMS Optimization Prize for Young Researchers, and the 2013 Daniel H. Wagner Prize.

Danny Segev is a faculty member in the Department of Statistics at the University of Haifa, Israel. His current research broadly focuses on the design of algorithmic ideas and analytical methods whose performance guarantees are provably good, with an emphasis on real-life problems that have practical importance, yet leave plenty of room for theoretical investigations. These include optimization problems and computational issues in inventory management, bioinformatics, statistical theory, scheduling, and graph theory.