Assortment and Inventory Optimization: From Predictive Choice Models to Near-Optimal Algorithms

by

Ali Aouad

M.S. in Applied Mathematics, Ecole Polytechnique (2013)

Submitted to the Operations Research Center
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2017

Author . . . .
Operations Research Center
July 31, 2017

Certified by . . . .
Vivek Farias
Associate Professor
Thesis Supervisor

Certified by . . . .
Retsef Levi
Professor
Thesis Supervisor

Accepted by . . . .
Patrick Jaillet
Chairman, Department Committee on Graduate Theses

Assortment and Inventory Optimization:
From Predictive Choice Models to Near-Optimal Algorithms

by

Ali Aouad

Submitted to the Operations Research Center
on July 31, 2017, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

Finding optimal product offerings is a fundamental operational issue in modern retailing, exemplified by the development of recommendation systems and decision support tools. The challenge is that designing an accurate predictive choice model generally comes at the detriment of efficient algorithms, which can prescribe near-optimal decisions. This thesis attempts to resolve this disconnect in the context of assortment and inventory optimization, through theoretical and empirical investigation.

First, we tightly characterize the complexity of general nonparametric assortment optimization problems. We reveal connections to maximum independent set and combinatorial pricing problems, allowing to derive strong inapproximability bounds. We devise simple algorithms that achieve essentially best-possible factors with respect to the price ratio, size of customers' consideration sets, etc.

Second, we develop a novel tractable approach to choice modeling, in the vein of nonparametric models, by leveraging documented assumptions on the customers' *consider-then-choose* behavior. We show that the assortment optimization problem can be cast as a dynamic program, that exploits the properties of a bi-partite graph representation to perform a state space collapse. Surprisingly, this exact algorithm is provably and practically efficient under common consider-then-choose assumptions. On the estimation front, we show that a critical step of standard nonparametric estimation methods (rank aggregation) can be solved in polynomial time in settings of interest, contrary to general nonparametric models. Predictive experiments on a large purchase panel dataset show significant improvements against common benchmarks.

Third, we turn our attention to joint assortment optimization and inventory management problems under dynamic customer choice substitution. Prior to our work, little was known about these optimization models, which are intractable using modern discrete optimization solvers. Using probabilistic analysis, we unravel hidden structural properties, such as weak notions of submodularity. Building on these findings, we develop efficient and yet conceptually-simple approximation algorithms for common parametric and nonparametric choice models. Among notable results, we provide best-possible approximations under general nonparametric choice models (up to lower-order terms), and develop the first constant-factor approximation under the popular Multinomial Logit model. In synthetic experiments vis-a-vis existing heuristics, our approach is an order of magnitude faster in several cases and increases revenue by 6% to 16%.

Thesis Supervisor: Vivek Farias
Title: Associate Professor

Thesis Supervisor: Retsef Levi
Title: Professor

Acknowledgments

My PhD experience is a story of all the people that I met: scholars, mentors and friends. I would like to thank my advisors for their strong dedication and unflinching support. Their concern and accessibility helped me through the slings and arrows of PhD life. Working alongside Prof. Retsef Levi has been a humbling and inspiring experience. I greatly admire Retsef's style as a scholar, advancing theory, exhilarating students in the classroom, training and mentoring researchers, and imparting operational insights to large organizations, beyond the academic sphere. He taught me rational optimism when encountering difficult obstacles in research, and patiently ingrained the idea of favoring concision and elegance over formalism in a proof. This thesis work would have been impossible without his guidance, in particular the simplified analysis developed in Chapter 5. This thesis also owes to the groundbreaking work of Prof. Vivek Farias on nonparametric choice models. My intellectual interactions with Vivek have shaped my aspirations as a scholar, in particular, on finding balance between theory and practice. I enjoyed our creative and enthusiastic discussions on new research directions. I am particularly thankful he introduced me to the areas of statistical learning and reinforcement learning; I hope to pursue this interest in the future and gain more familiarity with these fields. I am immensely grateful to Prof. Danny Segev for his dedication, technical guidance and friendship. Danny stands as a remarkable problem-solver; I learnt a great deal from his expertise in combinatorial optimization and approximation algorithms. He also stirred my interest toward location theory, which resulted in an exciting research paper on ordered median models. The little structure and clarity I could reach in my technical writing is indebted to his acute comments and constructive criticisms – always enlivened by an unmatched sense of humor.

I would like to sincerely thank Profs. Georgia Perakis and Huseyin Topaloglu for being on my thesis committee and providing very helpful comments throughout the job search. I am greatly appreciative of the support of Infoscout Inc, a data provider and market analysis company, in particular the CEO, Jared Schrieber, and the Head

of Insights, Prabhath Nanisetty. Applying our models to real-world choice data has been an important step in our research investigation in Chapter 3.

Finally, this thesis work has been favored by numerous discussions with labmates and friends in the Operations Research Center and MIT more broadly, including Arthur Flajolet, Rajan Udwani, Lennart Baardman, Will Ma, Rim Harriss and Charles Thraves. On many occasions, my discussions with Prof. Patrick Jaillet helped me gain perspective on my research path. Prof. Kris Ferreira gave me judicious advice during this critical year, and I enjoyed our refreshing research conversations on the interface between operations and marketing. Miles Lubin and Sebastien Martin shared instrumental advice to efficiently implement the algorithms described in this thesis. Collaborating with Prof. Yaron Shaposhnik to improve the patient flow at Brigham Women's Hospital was one of the most instructive research project. We have also learned together the vicissitudes of real-world data collection. My friend Andrew Li stimulated my interest for matrix completion problems during our research trip to French-Polynesia; our friendship will be a lasting memory from my PhD time.

I am grateful that this journey was made possible in such a vibrant and stimulating environment. Life in Cambridge would not have been the same without my roommates and friends, Michael Gharbi and Remi Lam. Most of all, I would like to express my deepest appreciation to my parents, who made sacrifices over the years to provide me the best opportunities. Their open-mindedness and generosity inspires me in every step I take.

To my love, Carole

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, consumer goods markets have grown in complexity, with significant proliferation of products in most categories. Finding an optimal product offering is a fundamental operational issue in modern retailing and online advertising, exemplified by the development of recommendation systems and decision support tools (Kök et al. 2009, Fisher 2011, Sinha et al. 2013). The typical problem is to determine what subset of items, out of all possible alternatives, should be offered (or displayed) to incoming customers. In addition, firms generally have rationed supply, and the inventory resources should be allocated strategically between the products offered.

This thesis work is more specifically centered around two core operational problems: *assortment optimization* and *inventory management.* Assortment optimization seeks to determine an optimal subset of items to offer the customers in order to maximize a given metric, such as expected revenue, click-through-rate, social welfare, etc., in the presence of various constraints. Inventory management is concerned with allocating inventory capacity across products, by finding optimal stocking levels. In practice, these two decision levers are inherently connected: the mismatch between inventory supply and uncertain demand creates stock-outs, which effectively alter the assortment offered to the incoming customers.

The advent of big data creates the opportunity to design sophisticated demand forecasting models to inform these operational decisions. Firms now collect large amounts of *choice data*, describing purchase decisions made by customers in various environments (e.g., through store-level aggregates, loyalty programs, digitalized receipts, online purchase history, etc.). In addition, e-commerce platforms can easily experiment new assortments or recommendation sets in face of incoming customers, to glean information about their preferences.

Modeling power and computational tractability. In this context, turning choice data into operational decisions presents several fundamental challenges. The typical workflow involves *modeling* customers' choice preferences, through what is known as a probabilistic choice model, *estimating* the corresponding model parameters from historical data, and finally *deciding* on the optimal operational policy. As such, an ideal choice model possesses the ability to: (i) explain choice data accurately, both in-sample and out-of-sample; (ii) admit statistically and computationally efficient estimation methods; (iii) yield tractable decision models.

Unfortunately, as further explained in Sections 1.1 and 1.3, the state of affairs in revenue management and combinatorial optimization literature suggests that these objectives are conflicting, across most families of choice models. On the one hand, the increasing availability of data incites to use general choice models. However, as choice models become more detailed, both their estimation from data, and the resulting optimization problems become computationally intractable. As illustrated by experiments conducted in this thesis, these complexity barriers are not merely theoretical. Despite the advances of discrete optimization solvers and integer programming, choice-based models are computationally prohibitive, even at small scale, notably in the context of inventory management. This disconnect between predictive choice models and tractable algorithms can somewhat be explained in that the traditional approach in revenue management regards *modeling*, *estimation* and *optimization* as distinct activities, that span across different research communities[1].

Overall contributions. The overarching objective of this thesis is to develop novel models and algorithms to inform assortment and inventory decisions, that strike a

[1] In simplified terms, *modeling* is the affair of empirical research, while *optimization* falls in the realm of algorithmic research.

good balance between modeling power and computational tractability. We primarily pursue theoretical investigation, through worst-case performance analysis. We also conduct extensive experiments, on both synthetic and real datasets, to showcase the merits of the newly proposed methods.

The first part of this thesis (Chapters 2 and 3) studies the standard assortment optimization problem. While this model takes a simplified view on the interaction between a firm and the incoming customers (e.g., overlooking inventory limitations), this formulation serves as a central building-block to various optimization problems in revenue management (Liu and Van Ryzin 2008, Sauré and Zeevi 2013, Golrezaei et al. 2014, Aouad and Segev 2015, Gallego et al. 2016), including network revenue management, display optimization, personalized assortment optimization, learn-and-earn models, etc. Despite the healthy history of assortment optimization research and growing literature in this area, this thesis makes contributions on three fronts:

1. We tightly characterize the approximability[2] of assortment optimization problems under the most general model specification, known as *nonparametric.*

2. We identify specialized settings, in the vein of nonparametric choice models, that admit exact polynomial-time algorithms, by exploiting novel graph-based properties.

3. We prove that this additional structure is also beneficial to estimate the model parameters efficiently, and shows significantly better prediction accuracy against common benchmarks, on both synthetic and real choice data.

The second part of the thesis (Chapters 4 and 5) investigates a more general class of optimization models, where a firm allocates its limited inventory across assortment products, at the beginning the selling-period, before observing a sequence of arriving customers with random preferences. While such joint assortment optimization and inventory management models have received significant attention since the seminal

[2] We reveal connections to maximum-independent set and combinatorial pricing, complemented by the design of algorithms essentially attaining the best-possible approximation ratios, up to lower-order terms.

paper of Mahajan and van Ryzin (2001), the antecedent literature bears only few positive results, under restrictive model specifications, and little is known about the computational aspects of these optimization models under commonly used choice models, like the Multinomial Logit model. Indeed, the computational challenges stem from the stochastic nature of customers' preferences, which vary dynamically upon inventory stock-outs. This thesis develops the first provably-good algorithms, and shows, quite surprisingly, that greedy-like procedures achieve worst-case performance guarantees under a broad class of choice models:

1. We devise the best-possible approximation under general nonparametric models with respect to the price parameters, up lower-order terms.

2. By unraveling submodular-like and decomposition properties, we show that this ratio can be beaten in specialized settings, including the popular Multinomial Logit choice model, to obtain constant-factor approximations.

3. In synthetic experiments, we demonstrate that the newly proposed algorithms outperform existing heuristics in terms of performance and speed.

All together, the analytical and empirical results we obtain underpin the relevance of problem-specific methodology vis-a-vis general-purpose heuristics and off-the-shelf solvers, which turn out to be inoperative in several empirical settings we consider. Furthermore, we develop a number of new concepts and techniques in combinatorial optimization, such as a weaker notion of submodularity, decomposition schemes and probabilistic couplings, possibly applicable in broader settings.

Conceptually speaking, our work illustrates the role of combinatorial structures in choice-based revenue management problems. In hindsight, this role is twofold. In the context of assortment optimization problems, we employ a graph-based structure as an *instrument* that enables to identify "well-behaved" classes of choice models. Specifically, we develop a graph-based dynamic program that allows a direct connection between modeling assumptions on the customers' choice behavior and the computational complexity of the resulting formulations. Conversely, in our study of joint assortment optimization and inventory management problems, we unravel *latent* structure (in the form of submodular-like properties) in order to work around the combinatorial nature of the problem.

Organization. In what follows, we pursue in Section 1.1 a qualitative discussion around practical aspects and challenges of choice modeling. Next, we provide a detailed description of our contributions in Section 1.2. Finally, we provide a bird's-eye view on related literature in Section 1.3.

1.1 Choice Modeling: Practical Aspects

Choice data and predictive task. Practical applications of choice modeling, such as the assortment optimization problems studied here, begin with transactional choice data. In our experiments, we had access to consumer-level receipt information from a panel population of hundreds of thousands of households. Specifically, our choice data can be viewed a sequence of purchase decisions recorded in different stores. The side information available in this dataset is very rich; it contains various customer-level covariates: age, income level, ethnicity, etc. However, the operational problems discussed in this thesis take the perspective of a central agent (e.g., retailer) whose decision levers (e.g., assortment or stocking levels) apply indifferently to all incoming customers. Thus, throughout this thesis, we consider a more common and limited observational setting, referred to as *assortment data*, where firms observe the relative sales volumes of products in given assortments[3].

Ideally, this data allows to learn the heterogeneous preferences of the customer population, through an estimation procedure. To formalize this estimation (and prediction) problem, suppose we are given n item alternatives, over which individuals make idiosyncratic choice decisions. Given the historical assortment data, we would like to learn the choice probability $\Pr[i|S]$ of picking item i in *any* assortment $S \subseteq [n]$.

[3] A concrete example in brick-and-mortar retailing is given by point-of-sale aggregates. Our methodology can also be leveraged to compute personalized assortments, by clustering beforehand the population into customer segments, using customer-level covariates.

This probability measure is known as the *choice model*; it can be viewed as a complex demand function over discrete alternatives[4].

Structure of choice models. At face value, choice models lie in a high dimensional functional space; it is unclear how to encode such functions as reasonable inputs to computation problems. The crux of choice modeling is precisely to posit a suitable structure in order to express these choice probabilities.

On the one extreme, an almost generic assumption placed on choice models is *rationality*. This property is rooted in utility theory: it states that preferences can be expressed as a distribution over ranked preference lists, which reflect how agents compare the utilities provided by the different items. In general, this model requires prescribing a probability parameter to each of $\Omega(n!)$ preference lists, which is clearly impractical for computational tasks. These caveats notwithstanding, this general model becomes practically relevant under the assumption of *sparsity*. That is, the *nonparametric* choice models are based on the assumption that the support of the distribution is "small" relative to the $\Omega(n!)$ possible preference lists.

On the other extreme, *separable* demand functions are the simplest possible choice models used in operations management. Here, each item has a fixed choice probability, independent of what other alternatives offered to the customers, i.e., the only preference lists that occur with positive probability are singletons. When demand is separable, the assortment optimization and inventory management problems take the form of a multi-item newsvendor model. Optimal stocking levels can be found easily, for instance using greedy algorithms that exploit the property of diminishing marginal returns. However, this traditional demand model overlooks the *substitution effects* across products, which are observed in differentiated markets. To wit, the choice probability for a given item often depends on what other items are simultaneously offered to the customers. Our experiments (along with previous literature) show that the separable demand model has rather limited explanatory power, explaining

[4] An important caveat to the estimation of a choice model from historical data is that retailers choose their assortment in response to their prior knowledge of the demand. However, since we focus on prediction accuracy (as opposed to causal inference), we can overlook these endogeneity effects.

less than 20% of the choice probabilities (R^2 score) in several product categories.

Hence, recent revenue management studies have operated a paradigm shift in demand modeling, from separable models to choice-based models. By capturing the substitution effects, the resulting assortment optimization and inventory management problems are considerably more challenging from a computational standpoint.

Modeling power and computational tractability. As stated earlier, most families of choice models are subject to a limiting tradeoff between modeling power and computational tractability.

A concrete example is the Multinomial Logit choice model, widely studied in the literature (including in Chapter 5 of the present thesis) and very popular among practitioners. This model assigns a preference weight parameter $w_i \in \mathbb{R}^+$ to each item $i \in [n]$, and assumes that choice probabilities are proportional to the preference weights in every assortment, i.e., for any $S \subseteq [n]$ and $i \in S$, $\Pr[i|S] \propto w_i$. Following important research efforts, this parametric model is now well-understood and admits simple and efficient algorithms for data-driven estimation and various assortment optimization problems. Nevertheless, this model has a limited explanatory power in regard to customer heterogeneity. Indeed, practical applications usually require assuming that preferences arise from a *mixture* of Multinomial Logit instances, each corresponding to a distinct *customer class*. As shown by the predictive experiments of Chapter 2, capturing heterogeneous customer classes indeed leads to dramatic improvements of the model accuracy, even when restricting attention to a specific segment of the population. However, a well-known negative result in assortment optimization literature shows that the computational tractability of the MNL model does not carry over to the finite mixture model, even with two customer classes (Bront et al. 2009, Rusmevichientong et al. 2014). The state of the art algorithm available for assortment optimization has exponential dependency with respect to the number of mixture components (Désir and Goyal 2014), so that its applicability beyond 3-4 customer classes is unclear. Similar limitations exist on the estimation front: the mixture model is not generally identifiable, as can be shown in simple examples with two products and two customer classes.

A similar tradeoff pertains to nonparametric choice models. Here, the sparsity of the distribution over preference lists controls the desired level of heterogeneity in a given instance. For example, nonparametric models enable to vary the sparsity of the distribution depending on the amount of data available (Farias et al. 2013). Unfortunately, this view of sparsity, in and of itself, is not sufficient to yield tractable estimation and optimization problems. Indeed, estimating sparse distributions generally requires to search over the space of permutations, which is itself a notoriously hard computational problem (Ailon et al. 2008, van Ryzin and Vulcano 2014). Furthermore, the complexity of state-of-the-art algorithms for assortment optimization and inventory management problems grows exponentially with the sparsity of the distribution (Honhon et al. 2010). This thesis work attempts to work around the limitations of nonparametric models by pursuing two types of approaches. First, these optimization models are studied, in their utmost generality, through the lens of approximation algorithms. Outside of theory, these approximations are shown to be efficient in a practical sense. Second, we investigate more parsimonious models by leveraging documented assumptions on customers' preferences, to find a middle ground between parametric models and general nonparametric models. Specifically, in Chapter 3, our work identifies a class of distributions over preference lists with exponentially many degrees of freedom[5], and yet at the same time, assortment optimization and estimation problems can be solved efficiently[6].

1.2 Our Results

Our main results can be summarized as follows.

We provide the best-possible approximability bounds for nonparametric assortment optimization models in Chapter 2. Since nonparametric models, in their utmost generality, are subject to strong computational limitations, we develop a novel mod-

[5] In the space of distributions over permutations.

[6] The running time complexity scales polynomially in the sparsity of the corresponding distribution, in contrast to the state of affairs for general nonparametric models.

eling approach that leverages documented assumptions on the customers' *consider-then-choose* behavior. This approach builds upon a bi-partite graph representation of nonparametric assortment optimization models, which also describes the states of a natural dynamic programming formulation. By studying the properties of this graph, we identify various common assumptions that lead to polynomial-time formulations (including for constrained variants of the problem). We also show that a critical step of the standard nonparametric estimation methodology (rank aggregation) can be solved in polynomial time in settings of interest, contrary to general nonparametric models. We conduct a number of predictive experiments on synthetic and real-world data that show strong merits vis-a-vis common parametric choice models.

Next, we turn our attention to joint assortment optimization and inventory management problems under dynamic choice substitution. While these models were generally viewed as intractable, using various probabilistic arguments, we are able to unravel hidden structural properties, such as weak notions of submodularity and decomposition schemes. Building on these findings, we develop the first provably-good algorithms in various nonparametric and parametric settings. Among notable results in Chapters 4 and 5, we devise the best-possible approximation essentially attainable under general nonparametric choice models. We also show that this ratio can be beaten in more specialized settings, including the widely-used Multinomial Logit choice model, to obtain a constant-factor approximation. Interestingly, the newly proposed algorithms are conceptually simple and rely primarily on greedy procedures. In extensive synthetic experiments, we demonstrate the practical relevance of our algorithms against heuristics proposed in prior literature.

Below, each chapter is discussed in turn, with a more detailed description of our contributions. Note that we provide an implementation for most algorithms described in this thesis, and make our code available for other researchers and practitioners as a suite of packages.

1.2.1 Chapter 2 – Complexity of non-parametric assortment optimization models

We provide the best-possible approximability bounds for assortment optimization under a general choice model, where customer choices are modeled through a distribution over preference lists of their preferred products. This nonparametric model subsumes most choice models of interest in revenue management applications. From a technical perspective, we show how to relate this optimization problem to the computational task of detecting large independent sets in graphs, allowing us to argue that general nonparametric models are extremely hard to approximate with respect to various problem parameters, even under sparse distributions. These findings are complemented by a number of approximation algorithms that attain essentially best-possible factors, proving that our hardness results are tight up to lower-order terms. Our results imply that a simple and widely studied policy, known as revenue-ordered assortments, achieves the best possible performance guarantee with respect to the price parameters.

1.2.2 Chapter 3 – Consider-then-choose choice models

Empirical literature in marketing and psychology shows that that customers choose among alternatives in two phases, by first screening products to decide which alternatives to consider, before then ranking them. We develop a dynamic programming framework to study the computational aspects of assortment optimization under consider-then-choose premises, that impose structural restrictions on distributions over preference lists. Although nonparametric choice models generally lead to computationally intractable assortment optimization problems, we are able to show that for many practical and empirically vetted assumptions on how customers consider then choose, our resulting dynamic program is efficient. Our approach unifies and subsumes several specialized settings analyzed in previous literature. In synthetic experiments, our algorithms lead to practical computation schemes that outperform a state-of-the-art integer programming solver in terms of running time, in several parameter regimes of interest.

Furthermore, we argue that the proposed consider-then-choose structure is also beneficial from an estimation standpoint. Indeed, a critical step of nonparametric estimation methods is *rank aggregation*, that consists in optimizing a linear objective over the space of permutations. While this problem is generally NP-hard, we provide a polynomial time dynamic programming formulation under the proposed consider-then-choose model. Empirically, we demonstrate the predictive power of our modeling approach on a combination of synthetic and real industry datasets, where prediction errors are significantly reduced against common parametric choice models (namely, the Mixture of Multinomial Logits with up to 3 customer segments).

Conceptually-speaking, our approach attempts to find a middle ground between parametric and general nonparametric models. Our approach to "adding structure" yields tractable models, both on the decision-making front and on the estimation front. In particular, we identify classes of distributions with exponentially many degrees of freedom (in the space of permutations), and yet at the same time, assortment optimization and rank aggregation are polynomially solvable in these settings.

1.2.3 Chapter 4 – Joint assortment and inventory decisions: Nonparametric models

Motivated by applications in highly differentiated markets, such as retailing, online advertising and airlines, we consider the single-period joint assortment optimization and inventory management problem, where stock-out events elicit *dynamic substitution* effects. This class of problems is known to be notoriously hard to deal with from a computational standpoint, since the seminal paper by (Mahajan and van Ryzin 2001). In fact, prior to our work, only a handful of modeling approaches were shown to admit provably-good algorithms, at the cost of strong restrictions on customers' choice outcomes. Our main contribution is to provide the first efficient algorithms with provable performance guarantees under a broad class of choice models. Under general nonparametric choice models, our approximation algorithm is best-possible with respect to the price parameters, up to lower-order terms. In particular, we obtain a constant-factor approximation under horizontal differentiation, where product prices are uniform. In more structured settings, where the customers' ranking behavior is motivated by price and quality cues, we derive improved guarantees through tailor-made algorithms. In extensive computational experiments, our approach dominates existing heuristics in terms of revenue performance, as well as in terms of speed, given the myopic nature of our methods. From a technical perspective, we introduce a number of novel algorithmic ideas of independent interest, and unravel hidden relations to submodular maximization.

1.2.4 Chapter 5 – Joint assortment and inventory decisions: Multinomial Logit model

We study the joint assortment optimization and inventory management problem, where stock-out events elicit dynamic substitution effects, described by the Multinomial Logit (MNL) choice model. The MNL has gained widespread popularity among revenue management practitioners, since it can be estimated efficiently, even with limited data (Ford 1957, McFadden 1973), and it leads to to tractable assortment optimization formulations (Talluri and van Ryzin 2004, Rusmevichientong et al. 2010, 2014). Nevertheless, the dynamic problem formulation in question is not known to admit efficient algorithms with analytical performance guarantees prior to this work, and most of its computational aspects are still wide open.

In this setting, we devise the first provably-good approximation algorithm, attaining a constant-factor guarantee for a broad class of demand distributions, that satisfy the *increasing failure rate* property. Our algorithm relies on a combination of greedy procedures, where inventory decisions are restricted to specific classes of products and the objective function takes modified forms. We demonstrate that our approach substantially outperforms state-of-the-art heuristic methods in terms of performance and speed, leading to an average revenue gain of 6% to 16% on synthetic instances. In the course of establishing our main result, we develop new algorithmic ideas that may be of independent interest. These include weaker notions of submodularity and monotonicity, shown sufficient to obtain constant-factor worst-case guarantees, despite using noisy estimates of the objective function.

1.3 Further Readings

Our research effort is positioned in a growing literature about assortment optimization and choice modeling techniques. Here, we delineate our contributions in this literature. A more comprehensive survey is provided through the different chapters of this thesis.

Parametric choice models. This line of research focuses on the standard assortment optimization problem (with known distribution and unlimited inventory), where positive results were obtained under variants the Multinomial Logit (MNL). Despite known limitations from a predictive standpoint, this choice model is the most popular approach among practitioners in light of its tractability. In particular, the MNL model yields tractable assortment optimization problems (Talluri and van Ryzin 2004) and can handle various extensions, including more general attraction-based models (Li et al. 2015, Davis et al. 2014), robust formulations (Rusmevichientong and Topaloglu 2012), totally-unimodular constraints (Davis et al. 2013), and exploration-exploitation models that tradeoff learning with revenue generation (Rusmevichientong et al. 2010, Sauré and Zeevi 2013). A more detailed review of these results is discussed in Chapter 3. On the other hand, there has been relatively little progress around joint assortment optimization and inventory management models, since the seminal work of Mahajan and van Ryzin (2001). Our contributions in Chapter 5 complete this positive picture, by deriving the first analytical results for joint assortment optimization and inventory management models under the Multinomial Logit choice model.

In contrast, assortment optimization in the face of a Mixture of Multinomial Logits, describing heterogeneous customer classes, is a notoriously hard problem (Bront et al. 2009, Rusmevichientong et al. 2014). The best known algorithms under mixture distributions have exponential dependency, so that their applicability beyond 3-4 customer classes is unclear (Désir and Goyal 2014).

The recent research efforts around the Markov chain model are closest in spirit to the approach developed in this thesis. This model, introduced by Blanchet et al. (2016), parametrizes the customers' rankings through the transitions of a Markov chain, thus generalizing the Multinomial Logit choice model. This model can be viewed as an alternative to capturing complex and heterogeneous substitution patterns, while yielding tractable decision models. The focus of this line of research has been around tractable methodologies for decision-making problems, specifically constrained assortment optimization (Blanchet et al. 2016, Désir et al. 2015), as well as pricing and network revenue management problems (Feldman and Topaloglu 2014).

Nonparametric choice models. The major part of our results fall under the nonparametric modeling framework, where preferences arise from a sparse distribution over ranked preference lists. This model was introduced by Rusmevichientong et al. (2006) and Farias et al. (2013), who provided robust revenue estimation methodologies and derived identifiability conditions for very sparse distributions. While other estimation methods were developed (Bertsimas and Mišic 2015), including under censored demand observations (van Ryzin and Vulcano 2014), the key computational challenge is to search over the space of permutations. In the context of assortment optimization problems, the general approach has been to devise heuristic methods and integer programming formulations (Jagabathula 2014, Bertsimas and Mišic 2015). In contrast with previous literature, this thesis investigates the tractability of nonparametric decision problems through the lens of approximation algorithms, and develop more parsimonious choice structures that allow for exact polynomial-time algorithms.

Joint assortment and inventory decisions. As mentioned earlier, in contrast to the standard assortment optimization problems, little is known about choice-based inventory management models. In this problem formulation, since firms have limited inventory capacity, the demand is explicitly described by a stochastic sequence of arriving customers with random preferences (rather than a single representative agent). As will be further explained in Chapters 4 and 4, under multiple stochastic arrivals, the problem we study becomes considerably more challenging than its standard counterpart, due to the *dynamic substitution*, elicited by stock-out events. In this setting the objective function is not well-behaved. For instance, under a general model of choice, for a continuous relaxation of the dynamic assortment problem, Mahajan and van Ryzin (2001) observed that the revenue function is not even quasiconcave, while various counter-examples provided in Chapter 4 show that this function is not submodular, even for very simple choice modeling approaches.

As a result, most of the work we are aware of in the context of dynamic substitution model develops heuristics based on continuous and deterministic relaxations, or study different probabilistic settings (Smith and Agrawal 2000, Mahajan and van Ryzin 2001, Gaur and Honhon 2006, Nagarajan and Rajagopalan 2008, Honhon et al. 2010, Honhon and Seshadri 2013, Segev 2015, Goyal et al. 2016). For the problem we consider, these approaches either give rise to exponential-time algorithms (Honhon et al. 2010), or converge to local optima, such as the gradient-descent method proposed by Mahajan and van Ryzin (2001), or apply to rather stylized choice models (Segev 2015, Goyal et al. 2016).

Chapter 2

Complexity of Nonparametric Assortment Optimization Models

2.1 Introduction

What selection of products should an e-retailer display for each search query? How does a brick and mortar retailer determine the product assortment in each store? The challenge of finding a selection of products that maximizes revenue or customer satisfaction, in the face of heterogeneous customer segments, who have different preferences across products, has been recognized in several industries as a strategic and operational driver of success. Specifically, assortment optimization is paramount to revenue management in highly differentiated markets, such as offline and online retail. The typical computational problem in this context is that of identifying a selection of products that maximizes revenue (assuming no stock-out events) based on previously-estimated random and heterogeneous customer preferences over the underlying set of products. The extensive literature in economics, marketing, and operation management proposes numerous approaches to modeling customer choice preferences, which are then used for predicting the variations in market shares in response to how the product mix changes.

This chapter is focused on studying the computational complexity of a very general, nonparametric, problem formulation, where customers choices are modeled through an arbitrary distribution over ranked preference lists. The incorporation of this choice model into an operational decision-making problem was first proposed by Rusmevichientong et al. (2006), while tractable estimation methodologies were investigated by Farias et al. (2013). This modeling approach, whose specifics are given in Section 2.1.3, subsumes most models of practical interest as special cases, being equivalent to a general random utility model, in which a representative agent maximizes his random utility function over a set of alternatives to derive his preferences.

Assortment optimization was shown to be computationally tractable under specific choice preference structures proposed in the revenue management literature. Probably the most well-known settings that still admit polynomial-time solution methods are the widespread multinomial-logit (MNL) model, and variants of the nested-logit (NL) model. In the specific context of nonparametric choice models, the work of Honhon et al. (2012) identifies classes of simple combinatorial structures enabling polynomial-time algorithms. A comprehensive review of these results is provided in Chapter 3. Additionally, we refer the reader to the work of Blanchet et al. (2016), Davis et al. (2014) and Li et al. (2015); the references therein provide an exhaustive overview of tractable approaches in assortment optimization.

Even though there has been an ever-growing stream of positive results for specific classes of instances, where various structural and probabilistic assumptions are made, assortment optimization is generally known to initiate computationally-hard problems. This was formally corroborated by several intractability results, such as that of Davis et al. (2014) and Gallego and Topaloglu (2014), who demonstrated that natural extensions of the NL model are NP-hard. Under mixtures of logits, this problem is known to be strongly NP-hard even for two customer classes, as shown by Bront et al. (2009) and Rusmevichientong et al. (2014). As a result, beyond attraction-based models with a single customer class, the family of tractable choice models remains quite limited.

It is worth noting that the above-mentioned results merely state that the problems in question cannot be solved to optimality in polynomial-time (unless P = NP), and in fact, very little is known about hardness of approximation in this context. To our knowledge, the only result in this spirit was given by Goyal et al. (2016), showing that under ranking preferences, the capacitated variant of assortment optimization is NP-hard to approximate within factor better than $1 - 1/e$.

2.1.1 Our results

The main contribution of this chapter is to provide best-possible inapproximability bounds for assortment optimization under nonparametric ranking preferences. From a technical perspective, we show how to relate this model to the computational task of detecting large independent sets in graphs, allowing us to argue that general nonparametric models are extremely hard to approximate with respect to various problem parameters. These findings are complemented by a number of approximation algorithms that attain essentially best-possible performance guarantees with respect to various parameters, such as the ratio between extremal prices and the maximum length of any preference list. Our results provide a tight characterization (up to lower-order terms) of the approximability of assortment optimization under a general model specification, as we briefly summarize next.

Hardness of approximation. By proposing a reduction from the maximum independent set problem, we prove that assortment optimization under nonparametric ranking preferences is NP-hard to approximate within factor $O(n^{1-\epsilon})$ for any fixed $\epsilon > 0$, where n stands for the number of products. It is worth noting that this bound holds even when all preference lists are derived from a common permutation over the set of products, meaning that all customers rank their alternatives consistently according to a unique order. Moreover, our reduction also gives an inapproximability bound of $O(\log^{1-\epsilon}(P_{\max}/P_{\min}))$, where $P_{\min}$ and $P_{\max}$ designate the minimal and maximal prices, respectively. Finally, through a reduction from the Min-Buying pricing problem, we establish APX-hardness even when there are only two distinct prices, with uniform probability of customer arrivals. The specifics of these results are given in Section 2.2.

Approximation algorithms. On the positive side, we devise approximation algorithms showing that the above-mentioned inapproximability bounds are best possible. By examining revenue-ordered assortments, we propose an efficient algorithm that attains performance guarantees of $O(\lceil \log(P_{\max}/P_{\min}) \rceil)$ and $O(\lceil \log(1/\tilde{\lambda}) \rceil)$, where $\tilde{\lambda}$ denotes the combined arrival probability of all customers who have the highest price item on their list. In particular, when all customer arrival probabilities are polynomially bounded away from 0, this bound translates to a logarithmic approximation (for example, under a uniform distribution). Finally, we devise a tight approximation algorithm in terms of the maximum length of any preference list. We prove that an $e\Delta$-approximation can be obtained via randomly generated assortments under a well-chosen distribution, where Δ denotes the maximal size of any preference list. Consequently, an immediate implication is that, when all preference lists are comprised of $O(1)$ products, we can approximate the optimal revenue within a constant factor. By derandomization, the resulting algorithm asymptotically matches the $O(\Delta^{1-\epsilon})$ inapproximability bound hiding within our reduction from the independent set problem. Additional details on these algorithms are provided in Section 2.3.

2.1.2 Subsequent work

The techniques developed in this chapter have spurred new complexity results for related assortment optimization problems. In particular, after communicating our reduction from the maximum independent set problem to Antoine Désir, Vineet Goyal, and Jiawei Zhang, they observed that ideas in this spirit provide tight inapproximability bounds for the mixture-of-MNL model (Désir and Goyal 2014). In addition, shortly after our work appeared online (Aouad et al. 2015), a working paper of Berbeglia and Joret (2015) focused on the performance analysis of revenue-ordered assortments. In comparison to the $O(\lceil \log(P_{\max}/P_{\min}) \rceil)$ approximation we provide in Section 2.3.1, they were able to improve on the constant hiding within the $O(\cdot)$-notation. However, unlike our tight inapproximability bound (see Section 2.2.1), they prove only constant-factor hardness, similar to the previously known result by Goyal et al. (2016). Their algorithmic results hold in a broader setting that generalizes the class of random utility choice models. Indeed, the only technical assumption required is the regularity axiom, stating that the probability of choosing a specific product does not increase when the assortment is enlarged. It is worth noting that the latter observation also holds for the analysis we develop in Section 2.3.1.

2.1.3 The nonparametric ranking preferences model

We are given a collection of n items (or products), where the per-unit selling price of item i is denoted by P_i. In addition, we model a population consisting of K customer-types, one of which arrives at random according to the probability distribution $(\lambda_1, \ldots, \lambda_K)$. Each customer-type is defined by a ranked preference list over the underlying set of products, according to which purchasing decisions are made. For any customer-type j, a preference list is formed by a subset of the products $C_j \subseteq [n]$, referred to as the *consideration set*, along with a linear order (or permutation) σ_j over the products. Without loss of generality, we can assume that the latter permutation is defined over all product alternatives, i.e., σ_j belongs to the symmetric group $\mathcal{S}_n$, although customer-type j only considers products in his consideration set C_j.

We define an assortment as a selection of products that is made available to customers. When faced with the assortment $S \subseteq [n]$, a customer-type purchases the most preferred item in his list that is made available by S. If none of these products is available, i.e., $S \cap C_j = \emptyset$, he leaves without purchasing any item. Under this decision mechanism, we use $R_j(S)$ to denote the revenue obtained should customer-type j arrive, for the assortment S. Conditional on the arrival of customer-type j, the resulting revenue is equal to the price of the product purchased according to the preference list (C_j, σ_j), or to 0 when none of these products has been made available. The objective is to compute an assortment of products whose expected revenue is maximized, i.e., to identify a subset $S \subseteq [n]$ that maximizes

$$\mathcal{R}(S) = \sum_{j=1}^{K} \lambda_j \cdot R_j(S) \ .$$

2.2 Hardness Results

2.2.1 Relation to maximum independent set

Our main inapproximability result proceeds from unraveling a well-hidden connection between assortment optimization and the maximum independent set problem (henceforth, Max-IS). To this end, we begin by recalling how the latter problem is defined, and state known hardness of approximation results due to Håstad (1996).

An instance of Max-IS is defined by an undirected graph $G = (V, E)$, where V is a set of n vertices, and E is the set of edges. A subset of vertices $U \subseteq V$ is said to be independent if no pair of vertices in U is connected by an edge. The objective is to compute an independent set of maximal cardinality. The most useful inapproximability result for our purposes is that of Håstad (1996), who proved that for any fixed $\epsilon > 0$, Max-IS cannot be approximated in polynomial time within factor $O(n^{1-\epsilon})$ unless P = NP.

Theorem 2.2.1. *Assortment optimization under ranking preferences is NP-hard to approximate within $O(n^{1-\epsilon})$, for any fixed $\epsilon > 0$.*

Proof. In what follows, we describe an approximation-preserving reduction Φ that maps any instance $\mathcal{I}$ of Max-IS, defined on an n-vertex graph, to an assortment optimization instance $\Phi(\mathcal{I})$, consisting of n products and n customers.

We begin by introducing some notation. Given a Max-IS instance $\mathcal{I}$ defined on an undirected graph $G = (V, E)$, let $V = \{v_1, \ldots, v_n\}$, each vertex being designated by an arbitrary label v_i. For each vertex $v_i \in V$, we use $N^-(i)$ to designate the indices of v_i's neighbors that are smaller than i, namely,

$$N^-(i) = \{j \in [n] : (v_i, v_j) \in E \text{ and } j < i\} \enspace .$$

The assortment optimization instance $\Phi(\mathcal{I})$ is defined as follows:

- For each vertex $v_i \in V$, we introduce a product indexed by i, with price $P_i = n^{2i}/\alpha$, where $\alpha = 1/\sum_{i=1}^{n} n^{-2i}$.

- Also, for each vertex $v_i \in V$, there is a corresponding customer-type whose preference list is defined as follows. The consideration set consists of the products $C_i = N^-(i) \cup \{i\}$, and the preference order σ_i is set such that i is the least preferable product. Any order between the remaining products $N^-(i)$ works for our purposes, but to have a concrete definition, we assume that σ_i orders these products by increasing indices (or equivalently, by increasing price).

- The probability (or arrival rate) of customer-type i is $\lambda_i = \alpha/n^{2i}$. Note that, by definition of α, these probabilities indeed sum to 1.

Based on the above-mentioned hardness results of Håstad (1996), in order to establish our inapproximability bound, it is sufficient to prove that Φ satisfies two properties:

1. For any independent set $U \subseteq V$ in $\mathcal{I}$ there exists a corresponding assortment S_U in $\Phi(\mathcal{I})$ with $\mathcal{R}(S_U) \geq |U|$.

2. Reciprocally, given any assortment S in $\Phi(\mathcal{I})$, we can efficiently construct a corresponding independent set $U_S \subseteq V$ in $\mathcal{I}$ of size at least $\lfloor \mathcal{R}(S) \rfloor$.

Claim 2.2.2. *For any independent set $U \subseteq V$, the assortment defined by $S_U = \{i : v_i \in U\}$ guarantees that $\mathcal{R}(S_U) \geq |U|$ for the assortment optimization instance $\Phi(\mathcal{I})$.*

Proof. We begin by observing that for any vertex $v_i \in U$, the only item made available by S_U within the consideration set C_i is product i. To see this, note that C_i consists of the products $N^-(i) \cup \{i\}$, and since U is an independent set, none of v_i's neighbors belongs in U, meaning in particular that $N^-(i) \cap S_U = \emptyset$. Therefore, conditional on the arrival of the customer-type i, the revenue obtained by the assortment S_U is exactly P_i. Thus, we can lower bound the expected revenue due to S_U by

$$\mathcal{R}(S_U) = \sum_{i=1}^{n} \lambda_i \cdot R_i(S_U) \geq \sum_{i \in S_U} \lambda_i \cdot P_i = \sum_{i \in S_U} \frac{\alpha}{n^{2i}} \cdot \frac{n^{2i}}{\alpha} = |U| \ .$$

□

Claim 2.2.3. *For any assortment $S \subseteq [n]$, we can compute in polynomial time an independent set $U_S \subseteq V$ whose cardinality is at least $\lfloor \mathcal{R}(S) \rfloor$.*

Proof. When faced with assortment S, the collection of customers can be partitioned into two groups: Those who purchase their most expensive product, and those who do not. We let $U_S \subseteq [n]$ denote the former subset. By definition, for all $i \in U_S$, customer i purchases product i, which is the most expensive one in C_i. The contribution of this purchase to the expected revenue is therefore $\lambda_i P_i = 1$. On the other hand, the contribution of each customer $i \in [n] \setminus U_S$ to the expected revenue is at most

$$\lambda_i \cdot \max_{j \in N^-(i)} P_j \leq \lambda_i \cdot P_{i-1} = \frac{\alpha}{n^{2i}} \cdot \frac{n^{2(i-1)}}{\alpha} = \frac{1}{n^2} \ .$$

Consequently, the total contribution of the latter customers (of which there are at most n) to the expected revenue is upper bounded by $1/n$. This means that precisely $\lfloor \mathcal{R}(S) \rfloor$ customers generate an expected revenue of 1, and therefore, $|U_S| = \lfloor \mathcal{R}(S) \rfloor$.

We now claim that the vertex set $\{v_i : i \in U_S\}$ forms an independent set in G. Indeed, if $i < j$ are both in U_S and $(v_i, v_j) \in E$, then $i \in N^-(j)$ and v_i is preferred over v_j by customer-type j, given the preference order σ_j. As a consequence, the contribution of customer j to the expected revenue is strictly less than 1, contradicting the fact that $j \in U_S$. □

□

Additional observations. It is worth noting that the maximum and minimum prices in our reduction, denoted $P_{\max}$ and $P_{\min}$ respectively, satisfy

$$\log \left(\frac{P_{\max}}{P_{\min}} \right) = \log \left(\frac{n^{2n}/\alpha}{n^2/\alpha} \right) = O(n \log n) \ .$$

Therefore, as an immediate corollary, we also obtain an inapproximability bound in terms of $P_{\max}$ and $P_{\min}$.

Corollary 2.2.4. *Assortment optimization under ranking preferences is NP-hard to approximate within $O(\log^{1-\epsilon}(P_{\max}/P_{\min}))$ for any fixed $\epsilon > 0$.*

Finally, as pointed out during the construction of σ_i, our reduction does not require a specific order within each preference list, as long as the most expensive product is the least desirable one. As a result, the inapproximability bounds we have just established hold even when all preference lists are derived from a common permutation over the set of products. That is, customer-types rank their alternatives consistently with respect to a single permutation.

2.2.2 Relation to the Min-Buying problem

In the previous reduction, we used distinct selling prices for products, as well as distinct arrival probabilities for customer-types. In fact, we constructed assortment optimization instances wherein both of these parameters have very large variability. Thus, motivated by practical choice specifications, an interesting question is whether the problem is rendered tractable under a small number of distinct prices, possibly with uniform arrival probabilities.

We resolve this question by proving that, for some constant $\alpha > 0$, assortment optimization is NP-hard to approximate within factor better than $1 + \alpha$ even when there are only two distinct selling prices, and preference lists occur according to a uniform distribution. It is worth mentioning that, when all products have identical prices, the problem becomes trivial. Specifically, by selecting all products in the assortment, we ensure that each preference list picks its maximal price item.

Our proof relies on a hardness result obtained by Aggarwal et al. (2004) in the context of multi-product pricing under the Min-Buying choice mode. We begin by formally introducing the latter problem.

An instance of the (uniform) Min-Buying pricing problem can be described as follows. Given a collection of n items, we assume there are K customer-types, each of which arrives at random with probability $1/K$. For all $j \in [K]$, customer-type j is characterized by a subset of products $S_j \subseteq [n]$ she is willing to purchase and by a budget B_j. She buys the least expensive item in S_j that meets her budget constraint. The objective is to determine a pricing vector $p \in \mathbb{R}_+^n$ to maximize the expected revenue under a random customer arrival, i.e.,

$$\max_{p \in \mathbb{R}_+^n} \frac{1}{K} \sum_{j=1}^{K} \min \left\{ p_i : i \in S_j \text{ and } p_i \leq B_j \right\} \ .$$

Aggarwal et al. (2004) proved that the Min-Buying problem is APX-hard even for instances with only two distinct budget values. Thus, the two-budget case of Min-Buying is NP-hard to approximate within $1 + \alpha$, for some constant $\alpha > 0$.

Theorem 2.2.5. *Assortment optimization under ranking preferences is NP-hard to approximate within $1 + \alpha$, for some constant $\alpha > 0$, even with two distinct selling prices and with uniform customer arrival probabilities.*

Proof. In what follows, we construct an efficiently-computable mapping Φ of each instance $\mathcal{I}$ of the Min-Buying problem to an instance $\Phi(\mathcal{I})$ of the assortment optimization problem satisfying the next two claims:

1. $\text{OPT}(\Phi(\mathcal{I})) \geq \text{OPT}(\mathcal{I})$.

2. Given any assortment for $\Phi(\mathcal{I})$, we can compute in polynomial time a pricing vector for $\mathcal{I}$ whose expected revenue is at least as good.

These properties jointly imply that our reduction translates the APX-hardness result of Aggarwal et al. (2004) to the assortment optimization problem, thus proving the desired claim.

We begin by noting that, without any loss in the expected revenue, any pricing vector of the Min-Buying problem can be transformed into another vector such that the price of each product is identical to the budget of at least one customer-type. In other terms, we can restrict the feasible pricing vectors to reside within $\mathcal{B}^\setminus$, where $\mathcal{B} = \{\mathcal{B}_\infty, \dots, \mathcal{B}_\mathcal{K}\}$.

Given an instance $\mathcal{I}$ of the Min-Buying problem, we define a corresponding assortment optimization instance $\Phi(\mathcal{I})$ as follows:

- The collection of products in $\Phi(\mathcal{I})$ is $[n] \times \mathcal{B}$, meaning that each combination of product $i \in [n]$ and price $B \in \mathcal{B}$ is represented by a distinct 'copy' product in $\Phi(\mathcal{I})$.

- There are K customer-types with uniform arrival probabilities.
- For every customer j, the preference list is derived from S_j by considering all copies of products in S_j that meet the budget constraint B_j, namely,

$$C_j = \left\{ (i,B) \in [n] \times \mathcal{B} : \rangle \in \mathcal{S}_| \text{ and } \mathcal{B} \leq \mathcal{B}_| \right\} \ .$$

Here, the preference order in σ_j is based on decreasing prices. That is, a less expensive product is always preferred over a more expensive one; when there are ties (equal prices), the relative ranking of products is set arbitrarily.

Proof. Proof of Claim 1: Let $p \in \mathcal{B}^n$ be a pricing vector in $\mathcal{I}$. We build an assortment that generates as much revenue in $\Phi(\mathcal{I})$ as the price vector p in $\mathcal{I}$. The idea is to determine an assortment where each customer buys the same combination of price and product as in $\Phi(\mathcal{I})$. Specifically, for each product $i \in [n]$, we select in the assortment product (i, p_i), i.e., the copy of i with price p_i, which is possible since $p_i \in \mathcal{B}$.

We now claim that this assortment generates as much revenue as the pricing vector p in the Min-Buying instance. Indeed, in this assortment, each customer-type $j \in [K]$ chooses the least expensive product that intersects his consideration set C_j, noting that ties between products do not have any impact since such products generate identical revenues. By construction of C_j, the purchase price of customer j is thus equal to that of the least expensive product in S_j under pricing p, assuming that the budget constraint is satisfied. We therefore get $\text{OPT}(\Phi(\mathcal{I})) \geq \text{OPT}(\mathcal{I})$. □

Proof. Proof of Claim 2: Reciprocally, let S be an assortment of the instance $\Phi(\mathcal{I})$. We prove that S can be translated in polynomial time into a pricing vector whose revenue in the Min-Buying instance is at least $\mathcal{R}(S)$. First, let us remark that although several of copies of the same product $i \in [n]$, with different prices, have been selected in S, all customers would only buy the least expensive copy. Indeed, if product i belongs to C_j then any cheaper copy belongs to C_j as well, and customer-type j only picks the cheapest. Therefore, we can eliminate from S all redundant copies that are not picked by any customer, and keep only one copy per product. By considering the remaining items, the assortment defines a partial assignment of prices to products: If the copy (i, B), of item i with price B, has been selected – we assign B as the price in $\mathcal{I}$, i.e., set $p_i = B$.

On the other hand, for any product of which no copy has been selected, we set its price to $\max(\mathcal{B})$. We observe that any customer-type j in $\mathcal{I}$, under pricing p, would purchase a product whose price is larger than that of the product she purchases in $\Phi(\mathcal{I})$, when faced with the assortment S. Indeed, if she purchases a product of price B in $\mathcal{I}$, then, either there exists $(i, B) \in C_j \cap S$ and customer j purchases a product of price lower than B in $\Phi(\mathcal{I})$, or $B = \max(\mathcal{B})$ and this customer generates a lower revenue in $\Phi(\mathcal{I})$. This yields the desired result. □

□

2.3 Approximation Algorithms

2.3.1 Approximation in terms of price ratio

In this section, we show that a natural algorithm, often used by practitioners and proposed in related literature for various models, attains the best-possible approximation ratio up to lower order terms under our general choice model. A revenue-ordered assortment consists in selecting all products whose price is greater or equal to a given threshold (Talluri and van Ryzin 2004, Rusmevichientong et al. 2014). In what follows, we use S_p to designate the revenue-ordered assortment corresponding to a minimum price of p, i.e., $S_p = \{i \in [n] : P_i \geq p\}$. As the next theorem shows, by limiting attention to such assortments and selecting the one with largest expected revenue, we are able to match the inapproximability bound established in Corollary 2.2.4.

Theorem 2.3.1. *The optimal revenue-ordered assortment approximates the optimal expected revenue within factor $O(\lceil \ln(P_{\max}/P_{\min}) \rceil)$.*

Proof. Without loss of generality, we may assume that empty preference lists have been discarded, and that the remaining arrival probabilities sum up to 1. Indeed, this can be achieved by renormalizing the distribution, which results in multiplying the expected revenue of any assortment by the same constant.

Let OPT designate the expected revenue obtained by the optimal assortment. For each customer $j \in [K]$, we define a corresponding budget B_j as the highest price on his list, i.e., $B_j = \max_{i \in C_j} P_i$. Without loss of generality, we can assume that customer indices are arranged so that $B_1 \geq \cdots \geq B_K$. Finally, we define $j^* \in [K]$ to be the customer j for which $B_j \cdot \sum_{r=1}^{j} \lambda_r$ is maximized, picking j^* arbitrarily, when the maximum value is attained by two or more customers.

We proceed by considering the assortment $S_{B_{j^*}}$, formed by all products whose price is greater or equal to B_{j^*}. Since $B_1 \geq \cdots \geq B_K$, any preference list in $[j^*]$ contains at least one product with a per-selling price of at least B_{j^*}. As a result, any such preference list generates a revenue greater or equal to B_{j^*} when faced with the assortment $S_{B_{j^*}}$, and therefore,

$$\mathcal{R}(S_{B_{j^*}}) = \sum_{j=1}^{K} \lambda_j \cdot R_j(S_{B_{j^*}}) \geq B_{j^*} \cdot \sum_{r=1}^{j^*} \lambda_r \ . \qquad (2.1)$$

In order to relate this quantity to OPT, we define

$$u^* = \min \left\{ u \in [K] : \sum_{j=1}^{u} \lambda_j \geq \frac{1}{2} \cdot \frac{P_{\min}}{P_{\max}} \right\} \ ,$$

noting that u^* is well defined, since $\sum_{j=1}^{K} \lambda_j = 1$. By remarking that B_j corresponds to the maximal revenue that can be extracted from each customer-type j, we can upper bound the optimal expected revenue by

$$\text{OPT} \leq \sum_{j=1}^{K} \lambda_j \cdot B_j \leq \sum_{j=1}^{u^*-1} \lambda_j \cdot B_j + \lambda_{u^*} \cdot B_{u^*} + \sum_{j=u^*+1}^{K} \lambda_j \cdot B_j \ . \qquad (2.2)$$

By definition of u^*, the first sum on the right is upper bounded by $B_1 \cdot P_{\min}/(2P_{\max}) \leq P_{\min}/2$. For the middle term, Equation (2.1) implies in particular that $\lambda_{u^*} \cdot B_{u^*} \leq$

$\mathcal{R}(S_{B_{j^*}})$. Finally, we can upper-bound the last sum as follows:

$$\begin{aligned}\sum_{j=u^*+1}^{K} \lambda_j \cdot B_j &= \sum_{j=u^*+1}^{K} \frac{\lambda_j}{\sum_{r=1}^{j} \lambda_r} \cdot \left(B_j \cdot \sum_{r=1}^{j} \lambda_r \right) \\ &\leq \sum_{j=u^*+1}^{K} \frac{\lambda_j}{\sum_{r=1}^{j} \lambda_r} \cdot \left(B_{j^*} \cdot \sum_{r=1}^{j^*} \lambda_r \right) \\ &\leq \sum_{j=u^*+1}^{K} \frac{\lambda_j}{\sum_{r=1}^{j} \lambda_r} \cdot \mathcal{R}(S_{B_{j^*}}) \\ &= \sum_{j=u^*+1}^{K} \left(\int_{\sum_{r=1}^{j-1} \lambda_r}^{\sum_{r=1}^{j} \lambda_r} \frac{1}{\sum_{r=1}^{j} \lambda_r} dx \right) \cdot \mathcal{R}(S_{B_{j^*}}) \ ,\end{aligned}$$

where the first inequality follows from the definition of j^*, and the second inequality is derived from Equation (2.1). By the monotonicity of $x \mapsto \frac{1}{x}$, we obtain:

$$\begin{aligned}\sum_{j=u^*+1}^{K} \lambda_j \cdot B_j &\leq \sum_{j=u^*+1}^{K} \left(\int_{\sum_{r=1}^{j-1} \lambda_r}^{\sum_{r=1}^{j} \lambda_r} \frac{1}{x} dx \right) \cdot \mathcal{R}(S_{B_{j^*}}) \\ &= \left(\int_{\sum_{r=1}^{u^*} \lambda_r}^{1} \frac{1}{x} dx \right) \cdot \mathcal{R}(S_{B_{j^*}}) \\ &\leq \left(\int_{\frac{1}{2} \cdot \frac{P_{\min}}{P_{\max}}}^{1} \frac{1}{x} dx \right) \cdot \mathcal{R}(S_{B_{j^*}}) \\ &= \ln \left(2 \cdot \frac{P_{\max}}{P_{\min}} \right) \cdot \mathcal{R}(S_{B_{j^*}}) \ ,\end{aligned}$$

where the second inequality follow from the definition of u^*.

As a result, we can now infer from inequality (2.2) that the assortment $S_{B_{j^*}}$ indeed approximates the optimal expected revenue within factor $O(\lceil \ln(P_{\max}/P_{\min}) \rceil)$, since

$$\begin{aligned}\text{OPT} &\leq \frac{P_{\min}}{2} + \left(1 + \ln \left(2 \cdot \frac{P_{\max}}{P_{\min}} \right) \right) \cdot \mathcal{R}(S_{B_{j^*}}) \\ &\leq \left(\frac{3}{2} + \ln \left(2 \cdot \frac{P_{\max}}{P_{\min}} \right) \right) \cdot \mathcal{R}(S_{B_{j^*}}) \\ &\leq \frac{5}{2} \cdot \left\lceil \ln \left(\frac{P_{\max}}{P_{\min}} \right) \right\rceil \cdot \mathcal{R}(S_{B_{j^*}}) \ .\end{aligned}$$

Here, the second inequality is obtained by observing that $P_{\min} \leq \mathcal{R}(S_{B_{j^*}})$, since by the choice of j^* and by our initial assumption that all empty lists have been eliminated, we have

$$\mathcal{R}(S_{B_{j^*}}) \geq B_{j^*} \cdot \sum_{j=1}^{j^*} \lambda_j \geq B_K \cdot \sum_{j=1}^{K} \lambda_j \geq P_{\min} \ .$$

□

As a corollary, we prove that revenue-ordered assortment also achieve an approximation ratio of $O(\lceil \log(1/\tilde{\lambda}) \rceil)$, where $\tilde{\lambda}$ denotes the combined arrival probability of all customers who have the highest price item on their list. In particular, when all arrival probabilities are polynomially bounded away from 0, i.e. $\Omega(1/\text{poly}(K))$, this bound translates to an $O(\log K)$ approximation (for example, under a uniform distribution).

Corollary 2.3.2. *The assortment optimization problem under ranking preferences can be approximated within factor $O(\lceil \log(1/\tilde{\lambda}) \rceil)$.*

Proof. We prove that, when all products with price smaller than $(\tilde{\lambda}/2) \cdot P_{\max}$ are eliminated, there is still an assortment that generates an expected revenue of at least OPT/2. This transformation guarantees that all remaining prices are within factor $2/\tilde{\lambda}$ of each other, in which case the upper bound given in Theorem 2.3.1 becomes $O(\lceil \log(1/\tilde{\lambda}) \rceil)$.

Let $\bar{S}$ designate the subset of products that have been eliminated, i.e., $\bar{S} = \{i \in [n] : P_i \leq (\tilde{\lambda}/2) \cdot P_{\max}\}$. When we eliminate products from an assortment, the probability that a customer purchases each of the remaining products (and consequently, the expected revenue from the remainder selection) can only increase. For this reason, it is sufficient to consider the contribution of $\bar{S}$ to the expected revenue of the optimal assortment, which can be upper bounded by

$$\sum_{j=1}^{K} \lambda_j \cdot R_j(\bar{S}) \leq \sum_{j=1}^{K} \lambda_j \cdot \frac{\tilde{\lambda}}{2} \cdot P_{\max} = \frac{\tilde{\lambda}}{2} \cdot P_{\max} \leq \frac{\text{OPT}}{2} \ ,$$

where the last inequality holds since $\text{OPT} \geq \tilde{\lambda} \cdot P_{\max}$. Indeed, this is the expected revenue of the assortment formed by stocking only the highest price product. □

2.3.2 Approximation in terms of list length

A close inspection of our reduction from Max-IS (see Theorem 2.2.1) reveals that the maximal size of any preference list was equivalent to the maximal degree Δ in the original graph. As a consequence, this inapproximability result gives an $O(\Delta^{1-\epsilon})$ hardness for assortment optimization with preference lists of size at most Δ. Since there are numerous algorithms for approximating Max-IS in terms of Δ (Karger et al. 1998, Alon and Kahale 1998, Halperin 2002), it is natural to investigate whether improved approximation guarantees can be obtained in terms of the maximum length of any list. In fact, the underlying assumption that each preference list is comprised of relatively few products finds behavioral and empirical support, and subsumes practical choice modeling specifications (Hauser et al. 2009).

In this setting, we analyze the expected revenue of random assortments arising from an appropriate generative distribution. By derandomization, we obtain a polynomial-time algorithm that is asymptotically tight, as asserted by the following theorem.

Theorem 2.3.3. *The assortment optimization problem under ranking preferences can be approximated within factor $e\Delta$, where Δ is the maximal size of a preference list.*

Proof. For any customer-type j, let $M(j)$ be the item with maximal price within the consideration set C_j. The optimal expected revenue is naturally bounded by

$$\mathrm{OPT} \leq \sum_{j=1}^{K} \lambda_j \cdot P_{M(j)} \ .$$

We construct a random assortment S_X through the following procedure: First, we independently draw values for $X_1, \ldots, X_n$, which are n i.i.d. Bernoulli variables with probability of success $1/\Delta$. Then, we pick each product to the assortment if and only if its corresponding variable is successful, meaning that $S_X = \{i \in [n] : X_i = 1\}$.

The important observation is that, for any preference list, the probability that customer-type j would purchase product $M(j)$ when faced with the assortment S_X

is at least

$$\frac{1}{\Delta} \cdot \left(1 - \frac{1}{\Delta} \right)^{|C_j|-1} \geq \frac{1}{\Delta} \cdot \left(1 - \frac{1}{\Delta} \right)^{\Delta-1} \geq \frac{1}{e\Delta} \ ,$$

where the last inequality holds since the function $[x \mapsto (1-1/x)^{x-1}]$ is monotone-decreasing over $(1,\infty)$, and converges to $1/e$. Indeed, this is precisely the probability that $M(j)$ belongs to S_X, and that all other products in C_j are unavailable. We conclude that the expected revenue of S_X is

$$\mathbb{E}_X \left[\sum_{j=1}^{K} \lambda_j \cdot R_j(S_X) \right] = \sum_{j=1}^{K} \lambda_j \cdot \mathbb{E}_X \left[R_j(S_X) \right] \geq \frac{1}{e\Delta} \cdot \sum_{j=1}^{K} \lambda_j \cdot P_{M(j)} \geq \frac{1}{e\Delta} \cdot \mathrm{OPT} \ .$$

This algorithm can be derandomized through the method of conditional expectations (see, for example, Chapter 16.1 in Alon and Spencer (2004)). Indeed, conditional on any partial assortment, i.e., a sequence of fixed binary values for the variables $X_1, \ldots, X_\ell$, the expected revenue can be computed exactly in polynomial time. Specifically, the independence between the Bernoulli variables allows to compute the probability that each customer-type picks a given product in his list. By applying the method of conditional expectations iteratively over $\ell = 1, \ldots, n$, we retrieve a deterministic assortment that approximates OPT within factor $e\Delta$. □

2.4 Concluding Remarks

Cardinality constraints. From a technical point of view, the approximation algorithms we propose in Section 2.3 make use of the freedom in picking assortments of any possible cardinality. An interesting direction for future research is to investigate whether our algorithms can be extended to the capacitated setting, where at most C distinct products can be stocked. Results in this spirit have previously been attained for several tractable models (see, for instance, Rusmevichientong et al. (2010), Davis et al. (2013)), although the computational difficulties here appear to be of significantly different nature.

Specification of the choice model. A particularly desirable property of revenue-ordered assortments is that an explicit description of the preference list distribution is not required, as long as one has access to an efficient oracle for computing the expected revenue of any given assortment. Therefore, the approximation guarantees we provide in Section 2.3.1 extend to a broader class of random utility choice models, where the distribution over preference lists potentially has a large support, such as Mixture of Multinomial Logits (Bront et al. 2009, Méndez-Díaz et al. 2014, Rusmevichientong et al. 2014, Désir and Goyal 2014, Feldman and Topaloglu 2015).

Uniform distribution. An interesting open question is that of determining the best approximation possible for uniform preference list distributions, i.e., when each customer-type is picked with equal probability. Such models are of practical importance, since in many applications, the distribution probabilities are conditioned by the number of samples used to estimate the model parameters. For this special case, one could try to narrow the gap between our APX-hardness results, given in Theorem 2.2.5, and the $O(\log K)$ approximation that follows from Corollary 2.3.2.

Chapter 3

Consider-then-Choose Choice Models

3.1 Introduction

Choosing an optimal assortment requires to model beforehand the customers preferences to predict accurately how the demand shares of products evolve in response to variations in the offer set, through what is called a choice model. As explained in Section 1, building an effective choice model strikes a delicate balance between several desired attributes. Indeed, as choice models become more detailed, both their estimation from data, and the resulting optimization problems face computational barriers, as made explicit by the hardness result obtained in the previous chapter.

The present chapter demonstrates that a class of nonparametric choice models, referred to as 'consider-then-choose' models, renders assortment optimization tractable under a variety of modeling premises. We present a unique dynamic programming formulation of the nonparametric assortment optimization problem, and show that a state space collapse in this problem yields the aforementioned tractability in several practical cases. Outside of theory, we empirically demonstrate the predictive power of our modeling approach using both synthetic and real industry datasets. We illustrate the computational practicality of our approach through extensive comparisons with state-of-the-art integer programming solvers.

Choice modeling and assortment optimization. Generally speaking, choice models can be divided into *parametric* and *non-parametric* models, the latter of which are effectively general distributions over preference lists of products. Until recently, most of the work related to assortment optimization has focused on parametric choice models, primarily attraction-based models in which customer preferences are modeled through a relatively small number of parameters. The survey by Kök et al. (2009) and book by Talluri and Van Ryzin (2006) present excellent overviews on such topics, and our literature review in Section 3.1.2 will summarize the state of the art here. In a nutshell, the literature presents us with the following dichotomy: on the one hand, for simple parametric models such as the Multinomial Logit (MNL) and variants of the Nested logit (NL) model, we now have efficient algorithms available for assortment optimization. On the other hand, these same models impose structural assumptions on customer preferences that may prove unrealistic in practice (Debreu 1960, Ben-Akiva and Lerman 1985). In attempting to address this latter issue, one may consider further generalized models such as a mixture of MNL models (MMNL), but then assortment optimization is no longer easy with the best known algorithms having a complexity that scales exponentially in the cardinality of the mixture (Bront et al. 2009, Rusmevichientong et al. 2014, Désir and Goyal 2014). In addition this latter class of models is notorious for problems such as over-fitting to data.

In an attempt to construct a parsimonious approach to modeling choice, *non-parametric* choice models, where the choice probability arises from a *sparse* distribution over preference lists (Rusmevichiengtong et al. 2006, Farias et al. 2013, van Ryzin and Vulcano 2014) have also received some attention. Here, each customer-type purchases the highest rank item in his preference list made available, or leaves without making any purchase. In this context, Farias et al. (2013) develop a robust estimation methodology, where the sparsity of the distribution scales with the amount of data available, allowing to attain better prediction accuracy than several common parametric models. On the other hand, there is relatively little known on the computational tractability of assortment optimization under these non-parametric models heretofore, beyond a few special cases of interest (Honhon et al. 2012). In fact, sparsity is generally insufficient to alleviate the computational hardness of assortment optimization, and the problem was shown to be NP-hard even to approximate in Chapter 2.

Consider-then-choose models. The aforementioned parametric and non-parametric models place extremely general conditions on the customer's decision making process, effectively requiring a customer to list *all* her options and then pick her most desirable from that list. In reality, one may naturally expect this process to be different with a customer using a set of simple rules to immediately *disregard* the vast majority of choices, and then rank (and select from) the small number of options left. We refer to such models as *consider-then-choose* models, wherein the *consideration set* is the (small) set of products considered. The history of these consider-then-choose idea originates in the marketing and psychology literature. The idea of whittling down choices into a consideration set was first posited by Campbell (1969) and formulated into a theory of the customer's behavior by Howard and Sheth (1969). In his seminal study, Hauser (1978) observed that the consideration set structure is in itself a significant explanatory factor of choice heterogeneity. We review the evolution of this approach to modeling choice in our literature review in Section 3.1.2. However, our objective with considering such models is twofold:

1. We believe that these models have the ability to model real-world data. This belief is motivated by empirical observations made in the antecedent literature on whether and how consideration sets are formed. Further, our modeling approach is borne out by several experiments on real industry datasets.

2. This consider-then-choose structure can be leveraged to mitigate the complexity of assortment optimization problems. In particular, we show that many empirically-vetted assumptions on how customers consider and choose lead to tractable assortment optimization problems.

3.1.1 Our results

Our main contribution is the development of a unified algorithmic framework to study the computational tractability of assortment problems under a family of preference-list based choice models that has been empirically vetted in the marketing literature, specifically, consider-then-choose models. Moreover, our framework allows a direct connection between modeling assumptions on the customers' choice behavior and the resulting computational complexity. Consequently, we show that several practical assumptions regarding how customers consider and choose lead to tractable assortment optimization models. Our dynamic programming algorithm, based on a divide-and-conquer approach in a specific graph representation, provides computationally efficient heuristics for more general preference list distributions, and outperforms a state-of-the-art integer programming solver (IP) for several class of instances. We demonstrate the predictive power of the proposed consider-then-choose modeling framework against common parametric models, using both synthetic experiments and real-world datasets. Our industry partner, Infoscout Inc, operates the largest purchase panel in the US, which provides longitudinal purchase information across retailers and product categories. In what follows, we provide a more detailed sketch of our contributions.

Dynamic program and graph representation. Motivated by the empirical observation that the structure of the consideration sets largely explains choice heterogeneity, we start by formulating in Section 3.3 a dynamic program for *unique-ranking* distributions, where customers consider arbitrary subsets of products, but their relative ranking preferences are derived from a common permutation. We introduce a *bipartite graph representation* of the problem, which is key to our approach and analysis. Indeed, the connected components of this graph capture a natural decomposition of the instance. Our dynamic program makes use of this decomposition procedure in a divide-and-conquer fashion. In contrast to standard dynamic programming, our algorithm relies upon a careful and exhaustive generation of the computational tree prior to solving the recursive equation. This approach allows for a state space collapse,

Table 3.1: Summary of results: polynomial running time guarantees for consider-then-choose choice models.

Consideration sets	Ranking functions	Running time	Sections
Induced intervals	Neighborhood of a ranking function	$O(n^4K)$	3.4.1, 3.6.1
Laminar properties		$O(n^2K^2)$	3.4.2, 3.6.1
Disjunction on d features		$O(n^{4d-2}K)$	3.4.3, 3.6.1
Intervals	Quasi-convex permutations	$O(n^4K^4)$	3.6.2
Two-feature compensatory	Two-feature compensatory	$O(n^4K^4)$	3.6.3

The parameter n describes the number of product alternatives, K denotes the number of preference lists (sparsity of the distribution), and d is a complexity parameter corresponding to the number of features considered by customers upon forming their consideration set in the disjunctive model. The notion of induced intervals means that there exists some *arbitrary numbering according to which the consideration sets are intervals.*

which substantially reduces the complexity under structural assumptions regarding how customers consider and choose. Specifically, we show that the complexity analysis generally boils down to 'counting' the number of connected subgraphs induced by the graph traversal. We prove that our algorithm runs in polynomial time for very sparse distributions, when the number of preference-lists grows logarithmically in the number of products. Also, we show that even in the worst case, our algorithm dominates the brute force enumerative approach.

The extension to general preference list-distributions requires additional technicalities, which are described in Section 3.5. Also, our results naturally extend to capacitated assortment optimization, with a constraint on the size of the assortment. This result is described separately in Appendix A.6.

Tractable consider-then-choose models. In Section 3.4, we investigate several models of consideration sets that stem from documented assumptions on the customers' purchasing behavior. We derive polynomial running time guarantees for the corresponding dynamic program in the unique-ranking setting. In Section 3.6, we investigate more general classes of distributions that combine heterogeneous consideration sets along with ranking heterogeneity. Our results subsume and extend several models studied in previous literature. Our complexity results, and the corresponding structural assumptions, are summarized in Table 3.1.

Empirical performance. Our numerical experiments on synthetic instances, described in Section 3.7, demonstrate that the algorithm is efficient in practice. We compare its performance against an integer programming formulation implemented using a state-of-the-art commercial solver (GUROBI v6.5). We demonstrate that the IP approach becomes intractable to solve large-scale instances of the quasi-convex model. Even under generic consideration set structures, our approach dominates the IP solver in several regime of parameters.

Finally, we demonstrate in Section 3.8 the versatility of our modeling approach against a benchmark formed by 'small' mixtures of Multinomial Logits (MMNL)[1]. The objective is to predict the relative purchase probabilities of products in various assortments. The predictive power of our approach is demonstrated by the experiments conducted on real-world datasets provided by our industry partner, in three distinct product categories. The errors in out-of-sample predictions of the purchase probabilities are reduced on average by 14% to 25% under various metrics. In synthetic experiments, our consider-then-choose model outperforms the benchmarks in the plurality of cases. Specifically, we use the following ground truth models: a large-mixture MMNL model and a simple consider-then-choose model.

3.1.2 Related literature

Our work relates to two streams of literature, namely the operational literature on choice modeling and assortment optimization, and the marketing literature on consider-then-choose models.

Choice models and assortment optimization. In the last two decades, growing product proliferation and differentiation has motivated a paradigm shift in demand modeling from independent demand models to choice-based models, to capture the substitution effects in a given product category (Mahajan and van Ryzin 2001, Kök

[1] In light of previous literature, assortment optimization is practical only for a mixture over a relatively small number of customer segments (a notion we will make precise in Section 3.8).

and Fisher 2007, Ratliff et al. 2008, Vulcano et al. 2010). In this context, assortment optimization has received a great deal of attention in operations management literature. Most of the focus has been on variants of this problem under the widespread attraction-based models such as the Multinomial Logit (MNL) model, the discrete Mixture of MNLs (MMNL), etc. Under MNL preferences, the problem is known to be polynomially solvable (Talluri and van Ryzin 2004, Rusmevichientong et al. 2010), and the solution methods were further advanced to handle more general settings (Rusmevichientong and Topaloglu 2012, Davis et al. 2013). However, the tractability of assortment optimization under the attraction-based models does not carry over to heterogeneous customer segments. That is, even with two segments the MMNL-based problem was shown to be NP-complete by Bront et al. (2009) and Rusmevichientong et al. (2014). For a fixed number of customer segments, Désir and Goyal (2014) developed a fully polynomial-time approximation scheme, but its computationally efficiency hinges on modeling few customer segments. Given these computational barriers, recent work in assortment optimization attempts to identify new probabilisitic models leading to tractable assortment optimization problems (Li et al. 2015, Blanchet et al. 2016, Davis et al. 2014).

On the other hand, there has been an emerging literature on preference list-based choice models (Rusmevichiengtong et al. 2006, Farias et al. 2013, Jagabathula and Rusmevichientong 2016). Here, the heterogeneity in choice is explicitly encoded through a distribution over preference lists. This approach to modeling choice is very general, e.g., the attraction-based models can be viewed as parametrized distributions over all potential preference lists. In this context, Farias et al. (2013) proposed an efficient methodology to make robust revenue predictions and derived recovery guarantees under certain technical conditions. To overcome the dimensionality of the estimation problem, van Ryzin and Vulcano (2014) proposed the 'market discovery' algorithm: starting from an initial collection of preference lists, the support of the distribution is enlarged iteratively by generating a preference list that increases the log-likelihood value, using dual information. While estimation methods have been investigated in this setting, assortment optimization remains mostly untapped. We characterized the complexity class of the problem under generic distributions in Chapter 2, while Honhon et al. (2012) developed tailor-made dynamic programming ideas for several special cases, which are subsumed by our analytical results.

Consider-then-choose literature. A steady line of research in marketing and psychology has studied various aspects of the decision-making strategies employed by customers. This literature gives rise to the following key observations.

- *Cognitive simplicity.* To alleviate the cognitive burden in multi-alternative decision tasks, individuals apply simple decision heuristics (Tversky and Kahneman 1975, Payne et al. 1996). Hence, the consideration sets are justified by the need to balance search efforts with potential gains (Hauser and Wernerfelt 1990, Roberts and Lattin 1991). Screening heuristics were shown to be rational under limited time and knowledge (Gigerenzer and Goldstein 1996, Gigerenzer and Selten 2002).

- *Consideration set heuristics.* Consequently, numerous studies in marketing have validated a consider-then-choose decision process, where customers screen products to a smaller relevant set of products before making choice decisions. For example, Pras and Summers (1975), Brisoux and Laroche (1981) and more recently Gilbride and Allenby (2004) showed empirically that customers often form their consideration sets through a 'conjunction' of elimination rules; see also Parkinson and Reilly (1979), Belonax and Mittelstaedt (1978), Laroche et al. (2003). For a detailed view on the consideration set literature, we refer the reader to the surveys by Hauser et al. (2009), Payne et al. (1996), Bettman et al. (1998).

- *Predictive power.* It has generally been observed that the incorporation of a two-stage decision process enables more accurate predictions, for example in market share forecasting (Urban 1975, Silk and Urban 1978), in choice modeling (Roberts and Lattin 1991) or in risky decision-making (Brandstatter et al.

> 2006). In fact, in his seminal work, Hauser (1978) observed that the heterogeneity in choice decisions is largely explained by the consideration sets. Even with a crude assumption on the ranking decisions (formed uniformly at random), the consideration set structure still explains nearly 80% of the heterogeneity in choice captured by a richer model, which combines the consideration sets with logit-based rankings. This observation can be explained in that the first stage decisions eliminate a large fraction of the alternatives and the resulting consideration sets are comprised of a few products in most categories (Reilly and Parkinson 1985, Belonax and Mittelstaedt 1978, Hauser and Wernerfelt 1990).

Motivated by these findings, the modeling approach we develop subsequently is centered around the notion of consideration sets. We will show that this approach to adding 'structure' to a general distribution over preference lists buys us a great deal from a computational complexity standpoint, and still allows strong predictive power. Prior to our work, the paper by Jagabathula and Rusmevichientong (2016) also incorporates a choice model based on consideration sets. The optimization problem considered therein relates more closely to combinatorial pricing.

3.2 Modeling Approach and Problem Formulation

Assortment optimization problem. Throughout this chapter we use the index $i \in \{1, \dots, n\} = [n]$ to denote one of n products, each is associated with a price P_i. In addition, we use the index $j \in \{1, \dots, K\} = [K]$ to denote one of K customer-types, each is associated with a consideration set $C_j \subseteq [n]$ that specifies the products she is willing to buy and a ranking function σ_j (that is, a permutation over products) that reflects her relative preferences. We let $(\lambda_1, \dots, \lambda_K)$ be the probability vector, where λ_j denotes the respective fraction of customer-type j in the population. The decision maker has to choose an assortment $\mathcal{A} \subseteq [n]$ that maximizes the total revenue. Specifically, let $\mathrm{Rev}\,(j, \mathcal{A})$ denote the revenue obtained from customer-type j given that assortment $\mathcal{A}$ is stocked. Note that if $\mathcal{A} \cap C_j = \emptyset$ then $\mathrm{Rev}\,(j, \mathcal{A}) = 0$ and otherwise $\mathrm{Rev}\,(j, \mathcal{A}) = P_{i(\mathcal{A},j)}$, where $i(\mathcal{A}, j) = \mathrm{argmin}_{i \in \mathcal{A} \cap C_j} \{\sigma_j\,(i)\}$ is the most preferred product of customer-types within $\mathcal{A}$. Therefore, the objective is to find an assortment $\mathcal{A}$ that maximizes the expected revenue: $\sum_{j\in[K]} \lambda_j \cdot \mathrm{Rev}\,(j, \mathcal{A})$.

We let $\mathcal{C} = \{C_j : j \in [K]\}$ be the collection of consideration sets and $\Sigma = \{\sigma_j : j \in [K]\}$ be the set of the ranking functions. In contrast to generic preference list distributions, our approach captures consider-then-choose purchasing behaviors by imposing constraints on the sets $\mathcal{C}$ and Σ, respectively. Below, we provide a high-level description of the ingredients used to model $\mathcal{C}$ and Σ, while the precise mathematical definitions are stated in the corresponding parts of this chapter.

Consideration set structure. We relate the collection of consideration sets $\mathcal{C}$ to the customers' cognitive process, where they screen products to form their consideration set. To this end, we build upon the survey by Hauser et al. (2009), which provides a unified framework to express the different consideration set models proposed in the marketing literature (see also Gilbride and Allenby (2004) for a similar mathematical formalism). Suppose that each product $i \in [n]$ is represented in by a vector $\boldsymbol{x}^{(i)} \in \mathbb{R}^d$ in latent d-dimensional feature space. A *screening rule* corresponds to a cut-off level $t_e \in \mathbb{R}$ on a given feature $e \in [d]$ that implies the elimination of all products $i \in [n]$ not satisfying $x_e^{(i)} \geq t_e$. Generally speaking, Hauser et al. (2009) explains that there are several families of cognitive processes whereby customers combine different screening rules to draw their consideration sets:

- *Conjunction of rules.* Here, a product is considered if *each one* of the specified screening rules are *all* satisfied. Namely, each consideration set $C \in \mathcal{C}$ is of the form:

$$C = \bigcap_{e\in[d]} \{i \in [n] : x_e^{(i)} \geq t_e\} \ .$$

- *Disjunction of rules.* A product is considered if *at least* one of specified screening rules is satisfied, leading to:

$$C = \bigcup_{e\in[d]} \{i \in [n] : x_e^{(i)} \geq t_e\} \ .$$

- *Compensatory models.* A product is considered based on a linear combination between different specified screening rules. In this case, there exists a utility vector $\boldsymbol{u} \in \mathbb{R}^d$ and a cut-off level $t \in \mathbb{R}$ such that the consideration set is of the form:
$$C = \{i \in [n] : \boldsymbol{u} \cdot \boldsymbol{x}^{(i)} \geq t\} .$$

In Sections 3.4 and 3.6, we investigate several classes of distributions over preference lists, where $\mathcal{C}$ is congruent with such combinations of screening rules. It is worth mentioning that, for a large enough d, each of the above models can replicate any arbitrary collection of consideration sets. Hence, for purposes of identifying tractable consideration set structures, we focus on a small number of screening rules – which is also consistent with the cognitive simplicity of customers' decisions.

Ranking decisions. Having explained how we model $\mathcal{C}$, it remains to describe Σ. To ease the exposition, our algorithmic framework is introduced in an incremental way. First, we focus on the heterogeneity of the consideration sets and start our discussion assuming that the collection of rankings Σ is a singleton. Here, we assume that there exists single ranking order common to all customer-types, i.e., $\sigma_j = \sigma$, and the heterogeneity in preferences stems only from the heterogeneity of the consideration sets. We refer to this setting as the *unique-ranking model.* As shall be seen subsequently, the unique-ranking model already subsumes several choice models studied in previous literature, and even in this setting, assortment optimization is computationally intractable. Specifically, it was shown in Chapter 2 that the problem under the unique-ranking model is NP-hard to approximate within factor $O\left(n^{1-\epsilon}\right)$ for any $\epsilon > 0$.

Our algorithmic approach and analysis carry over in the presence of heterogeneity in ranking decisions. Specifically, in Section 3.6, the unique-ranking assumption is relaxed in two ways: *(i)* by assuming that Σ is formed by similar rankings arising from the local perturbations of a central permutation, or *(ii)* by studying ranking structures motivated by behavioral assumptions (e.g., quasi-convex permutations).

3.3 Dynamic Program for Unique-Ranking Models

In this section, we present a dynamic programming (DP) formulation under unique-ranking distributions. As some obstacles must be surmounted to consummate our approach in the general case, the algorithm for arbitrary preference list distribution is described separately in Section 3.5, to ease the exposition. We formulate the dynamic program in two parallel ways, the first corresponds to a traditional recursive formulation and the second is an appropriate graph representation.

Preliminaries. Recall that an instance of the assortment problem is described by the set of parameters $n, K, \Sigma, \mathcal{C}, \sigma_j, C_j, \lambda_j$. Assuming that $\Sigma = \{\sigma\}$, without loss of generality the product indices can be rearranged according to the σ-ordering. That is item 1 is the most preferred product and n is the least preferred one – to lighten the notation, the reference to σ is omitted hereafter. In what follows, we use $[X]$ to denote the set $\{1, \dots, X\}$.

State space and objective function. The state space is formed by all pairs of subsets (S, T), where S is a subset of products in $[n]$ and T is a subset of types in $[K]$. Specifically, we let $J(S, T)$ be the maximum expected revenue that can be obtained from choosing an assortment of products within S to satisfy the customer-types in T. In the subproblem, we assume that only customers in T can occur and the arrival probabilities are directly induced from the original instance without a renormalization of the corresponding sub-vector. Formally, the subproblem (S, T) is formulated as follows:

$$J(S,T) = \max_{\mathcal{A} \subseteq S} \sum_{j \in T} \lambda_j \cdot \text{Rev}\left(j, \mathcal{A}\right) \ .$$

Graph Representation. We next introduce a bipartite graph representation G associated with each instance of the problem. The partite sets are formed by *(i)* the set of products, each of which is represented by a node, and *(ii)* the set of customer-types $[K]$. There is an edge between a customer-type node and a product node if the latter is included in the consideration set of the former. That is, we define the graph $G = ([n], [K], E)$, where $E = \{(i, j) \in [n] \times [K] : \; i \in C_j\}$. This graph induces the family of subgraphs $G[S, T]$ associated with each subproblem (S, T), that is, $G[S, T] = (S, T, E_{S,T})$, where $E_{S,T} = \{(i, j) \in E : \; i \in S \text{ and } j \in T\}$. The lemma below asserts that the partition of $G[S, T]$ into its connected components captures a decomposition of the instance (S, T) into several subproblems that can be solved independently. This decomposition scheme, represented in Figure 3-1a, is key to our recursion.

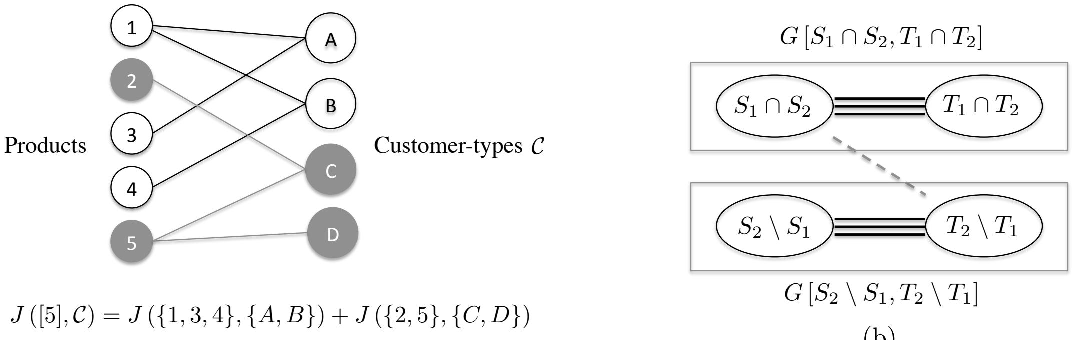
Lemma 3.3.1. *Assuming that the connected components of $G[S, T]$ are described by the collection of subgraphs $(G[S_u, T_u])_{u \in [r]}$, where S_u denotes a subset of product nodes in S and T_u is a subset of type nodes in T, then $J(S, T) = \sum_{u=1}^{r} J(S_u, T_u)$.*

Proof. It is sufficient to prove that the expected revenue generated in (S, T) by any assortment $\mathcal{A} \subseteq \mathcal{S}$ decomposes into the sum over $u \in [r]$ of the revenues generated in each subproblem (S_u, T_u) by the respective assortment $\mathcal{A}_u = \mathcal{A} \cap S_u$. Let j be a customer-type in T_u. The main observation is that customer-type j's most preferred product within the assortment $\mathcal{A}$ is the same as the one he prefers when faced with the assortment $\mathcal{A}_u$. Indeed, by connectivity, any product in $C_j \cap S$ that is considered by customer j, necessarily belongs to S_u. Since $(S_u)_{u \in [r]}$ forms a partition of S, an optimal assortment $\mathcal{A}$ for subproblem (S, T) is the union of optimal assortments $\mathcal{A}_u \subseteq S_u$ for each subproblem (S_u, T_u). □

Base case. If $S = [n]$ and $T = [K]$, then $J(S, T)$ corresponds to the original problem we wish to solve. Using Lemma 3.3.1, this problem can be broken-down into separate optimization problems according to the connected components partition. From this point on, we may assume without loss of generality that connectivity is an invariant of the subgraphs examined by the recursion.

Recursive step. We consider the subproblem (S, T) such that $G[S, T]$ is a connected subgraph. We define i as the most preferred product among product nodes S, i.e., $i = \min(S)$. The decision (or action) made by the dynamic program for state

Figure 3-1: Bi-partite Graph Representation: (a) Decomposition of the instance according to the connected components of the graph, (b) Illustration of the proof of Proposition 3.3.3.


1
2
3
4
5
A
B
C
D
Products
Customer-types $\mathcal{C}$
$J([5],\mathcal{C}) = J(\{1,3,4\},\{A,B\}) + J(\{2,5\},\{C,D\})$
$G[S_1 \cap S_2, T_1 \cap T_2]$
$S_1 \cap S_2$
$T_1 \cap T_2$
$S_2 \setminus S_1$
$T_2 \setminus T_1$
$G[S_2 \setminus S_1, T_2 \setminus T_1]$

(a)

(b)

(S,T) is whether to stock product i in the assortment or not. Next, we describe graph operations on $G[S,T]$ that correspond to each alternative. As these graph operations decompose $G[S,T]$ into more refined connected components, a natural recursion consists in examining the immediate reward and reward-to-go induced by each stocking decision.

Case 1: Product i is stocked. Let $T(i)$ be the customer-types whose consideration sets contain product i. The unique-ranking order implies that any product added to the assortment at some later point of the recursion is less preferred than i by any customer-type in $T(i)$. As a result, we can compute the expected revenue generated by their purchase of product i. In addition, since i is more preferred than any product that is stocked at some later point of the recursion, the customer-types $T(i)$ can be discarded from this point on. Thus we represent the reward-to-go as the optimal expected revenue associated with the *residual* subproblem $(S \setminus \{i\}, T \setminus T(i))$. In the graph representation, the decision to include i in the assortment corresponds to removing node i and its adjacent edges from the graph as well as deleting all nodes in $T(i)$ and their adjacent edges. Due to these graph operations, the residual subgraph $G[S \setminus \{i\}, T \setminus T(i)]$ is potentially not connected anymore. By Lemma 3.3.1, the subproblem can be broken-down according to the connected components partition. If $(S_u^+, T_u^+)_{u \in [r(+)]}$ are the resulting subproblems, the expected revenue is

$P_i \cdot \sum_{j \in T(i)} \lambda_j + \sum_{u=1}^{r(+)} J(S_u^+, T_u^+)$.

Case 2: product i is not stocked. All the customers in T remain unsatisfied and the reward-to-go is that associated with subproblem $(S \setminus \{i\}, T)$. The corresponding graph operation is the deletion of node i and its outgoing edges, and consider the residual subgraph $G[S \setminus \{i\}, T]$. Let $(G[S_u^-, T_u^-])_{u \in [r(-)]}$ describe the connected components of the residual subgraph. Then, by Lemma 3.3.1 it follows that the expected revenue is $\sum_{u=1}^{r(-)} J(S_u^-, T_u^-)$.

Combining the two decisions, the dynamic programming recursion is:

$$J(S,T) = \max \left[P_i \cdot \sum_{j \in T(i)} \lambda_j + \sum_{u=1}^{r(+)} J(S_u^+, T_u^+) \ , \ \sum_{u=1}^{r(-)} J(S_u^-, T_u^-) \right] \ . \tag{3.1}$$

State space collapse. In a naive implementation of the algorithm, one could solve the problem for all possible pair (S, T). However, this approach is inherently intractable and could be in the worst case as bad as 2^{n+K}. However, the dynamic program does not need to examine all corresponding subproblems to solve the initial instance. In fact, the recursion formula provides an algorithmic procedure to determine the precise 'minimal' number of subproblems needed to be solved. In contrast to a standard dynamic program, we will not assume that the state space is known a priori, and carefully generate a computational tree by processing the products from 1 to n (i.e., according to the unique order σ) adding nodes to the tree based on the recursion described above. The algorithm requires a two-pass implementation: first the computational tree is built by generating all subproblems of interest, using the recursive formula (3.1), then an optimal assortment is obtained by a backward induction.

Complexity analysis. We let $\mathcal{S}$ denote henceforth the exact state space that proceeds from the previous observation. Namely, $\mathcal{S}$ represents a collection of distinct subproblems, each of which belongs to the computational tree generated by the recursion. We now argue that the running time complexity is $O(nK \cdot |\mathcal{S}|)$. Indeed, building each node of the computational tree requires at most $O(nK)$ operations. This is the number of operations required to update the graph, compute the new connected components in $O(nK)$ operations and check whether each new subproblem already belongs to the computational tree in $O(n+K)$ operations using an appropriate search data structure, where each subproblem is encoded by an $n+K$-binary string. Then, the subproblems are solved backwards using the recursive formula with a total running time complexity of $O(K \cdot |\mathcal{S}|)$, taking $O(K)$ operations at each step to solve Equation (3.1). As a result, the complexity analysis boils down to estimating the size of the state space $\mathcal{S}$.

In the worst case, the number of connected subgraphs is still exponential. However, we establish in the next theorem that the state space is at most $\min(2^n, n \cdot 2^K)$, instead of the naive 2^{n+K}. Hence, the algorithm is efficient for applications in which the distribution over preference lists has a sparse support. It is worth noting that the running time is polynomial for $K = O(\log(n))$. Also, for $K = O(1)$, the running time is quadratic in the number of products, instead of the brute force approach in time $O(n^K)$.

Theorem 3.3.2. *The size of the state space is at most* $\min(2^n, n \cdot 2^K)$. *The running time complexity is quadratic in* n *for a constant number of types* K, *and polynomial for* $K = O(\log(n))$.

To the end of proving Theorem 3.3.2, we introduce a characterization of the state space, which considerably simplifies the analysis of the algorithm. We define a projection Φ from the collection of subproblems $\mathcal{S}$ (i.e., subproblems in the computational tree) onto the collection of subsets of $\{1 \dots n\}$ as follows:

$$\begin{aligned} &\Phi : \mathcal{S} \to \mathscr{P}([n]) \\ &\quad (S,T) \mapsto [\min(S)] \bigcap \left(\bigcup_{u \in T} C_u \right) . \end{aligned}$$

We establish below that Φ is injective, meaning that the size of the state space is equal to $|\Phi\langle \mathcal{S} \rangle|$.

Proposition 3.3.3. *Φ is injective, and as a result: $|\mathcal{S}| = |\Phi\langle\mathcal{S}\rangle|$.*

Proof. We assume (S_1, T_1), (S_2, T_2) are two subproblems that are generated by the recursion, such that $\Phi(S_1, T_1) = \Phi(S_2, T_2)$. By construction, $G_1 = G[S_1, T_1]$ and $G_2 = G[S_2, T_2]$ are two connected subgraphs.

Because G_1 is connected, there exists $u \in T_1$ such that $(\min(S_1), u)$ is an edge of G_1, meaning that $\min(S_1) \in C_u$. As a result, $\min(S_1) = \max(\Phi(G_1))$. By symmetry, we obtain:

$$\min(S_2) = \max(\Phi(G_2)) = \max(\Phi(G_1)) = \min(S_1) \ .$$

We infer from the connectivity of the subgraph $G[S_1, T_1]$ that $S_1 \subseteq \cup_{u \in T_1} C_u$. Since the set of products examined at previous steps of the recursion is exactly $[\min(S_1) - 1]$, we infer the equality $S_1 = \cup_{u \in T_1} (C_u \cap [\min(S_1), n])$. By a symmetric argument, $S_2 = \cup_{u \in T_2} (C_u \cap [\min(S_2), n])$.

As a result, what remains to be proven is simply that $T_1 = T_2$. Assume ad absurdum that $T_2 \setminus T_1 \neq \emptyset$ and let $u' \in T_2 \setminus T_1$. Under this assumption, we establish the following property.

Claim 3.3.4. *$C_{u'} \cap S_1 = \emptyset$.*

Proof of Claim 3.3.4. We assume otherwise and prove a contradiction. Because the two subgraphs G_1, G_2 both contain product node $\min(S_1)$, they initially lied in the same connected component of G. As a result, by looking at the sequence of algorithm iterations that generates G_1, we can define i as the minimal product examined by the algorithm after which u' gets disconnected from G_1. Then, product i has necessarily been added to the assortment, while node u' has been removed from the graph. Indeed, otherwise u' would still be connected to S_1 by hypothesis. Therefore, we obtain that $i \in C_{u'}$. It follows that $i \in \Phi(S_2, T_2)$ and thus $i \in \Phi(S_1, T_1)$. On the other hand, it is clear that i does not belong to $\cup_{u \in T_1} C_u$ otherwise some customer-types in T_1 would be discarded when i is selected in the assortment. This yields the desired contradiction. ■

The latter claim implies that there exists no edge between customer-types in $T_2 \setminus T_1$

and product nodes $S_1 \cap S_2$. In addition, there exist no edges between customer-type nodes $T_2 \cap T_1$ and product nodes $S_2 \setminus S_1$. Indeed, there would exist otherwise $u \in T_1$ and $i \in C_u \cap S_2$, such that $i \notin S_1$. Because $\min(S_2) = \min(S_1)$ we infer that $i \in C_u \cap [\min(S_1), n]$. By construction of our recursion, we obtain $i \in S_1$, which gives a contradiction. To conclude, as shown in Figure (3-1b) we observe that $G[S_2 \backslash S_1, T_2 \backslash T_1]$ and $G[S_2 \cap S_1, T_2 \cap T_1]$ are distinct connected components of $G[S_2, T_2]$, contradicting the connectivity of the latter subgraph. □

Finally, we derive a parametric bound on the state space, as a function of the consideration sets *diameter*. To this end, we define $\text{Diam}(\mathcal{C})$ as the maximal diameter of a consideration set in $\mathcal{C}$: $\text{Diam}(\mathcal{C}) = \max\{\max C - \min C : C \in \mathcal{C}\}$. The next claim comes as an immediate consequence of Proposition 3.3.3.

Corollary 3.3.5. *The size of the state space is at most* $2^{\text{Diam}(\mathcal{C})}$. *Hence, the running time complexity is polynomial when* $\text{Diam}(\mathcal{C}) = O(\log(n))$.

3.4 Modeling the Consideration Sets

In this section, we identify several consideration set models that stem from behavioral assumptions, for which $|\mathcal{S}|$ is polynomial in the input size.

3.4.1 Induced Intervals Structure

Definition 3.4.1. *A collection of consideration sets* $\mathcal{C}$ *is a family of* induced intervals *if it forms a collection of intervals when numbered according to some arbitrary permutation* $\pi : [n] \to [n]$, *i.e.,* $\pi\langle C_j \rangle$ *is an interval for any customer-type* $j \in [K]$.

Using the screening rule formalism of Section 3.2, it can be verified that this property arises when the consideration sets are formed as a conjunction of two screening rules, meaning that each consideration set in $\mathcal{C}$ is of the form $\{i \in [n] : x_1^{(i)} \geq t_1 \wedge x_2^{(i)} \geq t_2\}$ for some cut-off levels t_1, t_2, and the corresponding features are inversely related, i.e., for any products $i_1, i_2 \in [n]$, $x_1^{(i_1)} \geq x_1^{(i_2)}$ implies that $x_2^{(i_1)} \leq x_2^{(i_2)}$. As a practical example, *price* and *quality* are significant drivers of the customers' choices, who might use the following screening rules:

- *Budget constraint:* Customers would eliminate at an early stage of the purchasing process the products that they cannot afford (Fisher and Vaidyanathan 2009, Jagabathula and Rusmevichientong 2016).

- *Perceived quality cut-off:* There is empirical evidence that price is used as a cue for quality (Zeithalm 1988, Posavac et al. 2005), hence customers would eliminate all products cheaper than the given cut-off level.

The consideration sets emanating from a conjunction between budget constraints and perceived quality cut-offs are intervals with respect to the price order. Also, it is worth noting that induced interval consideration sets with unique-ranking subsumes the *downward substitution* model proposed by Pentico (1974) and Honhon et al. (2012) as the special case where the preference order $\Sigma = \{\sigma\}$ coincides with $\{\pi\}$. In contrast, for the induced intervals in question, the preference order σ is generally distinct from the inducing permutation π.

We now prove that the dynamic programming algorithm runs in polynomial time under this class of distributions, by bounding the number of connected subgraphs generated along the dynamic program. Intuitively, our counting argument utilizes the observation that a union of overlapping intervals is itself an interval.

Theorem 3.4.2. *Under induced intervals consideration sets, the dynamic program has a running time of $O(n^4 K)$. Using a specific data-structure, the running time complexity is $O(n^2 K \cdot \log(K))$ in the special case of the downward substitution model.*

Proof. Given that the function Φ is injective according to Proposition 3.3.3, it is sufficient to upper bound $|\Phi\langle\mathcal{S}\rangle|$. To this end, we let (S,T) designate a subproblem of $\mathcal{S}$. The key observation is that, due to the connectivity of $G[S,T]$, the union of the consideration sets in T is itself an interval according to the ordering π. Indeed, assume ad absurdum that there exists products $i, j \in S \cap [n]$ and a product $\alpha \in [n] \setminus S$ such that $\pi(i) < \pi(\alpha) < \pi(j)$. Then, for any customer-type j in T, since $\pi\langle C_j\rangle$ is an interval, we infer that either $\pi\langle C_j\rangle \subseteq [\pi(\alpha)-1]$ or $\pi\langle C_j\rangle \subseteq [\pi(\alpha)+1,n]$. Denoting by T_1 the customer-types that satisfy the former inclusion, and T_2 the latter one, we conclude that the subgraph $G[S,T]$ decomposes into distinct connected components $G[S_1,T_1]$ and $G[S_2,T_2]$, where S_1 is the subset of products whose π-indices belong to $[\pi(\alpha)-1]$ and S_2 corresponds to the π-indices in $[\pi(\alpha)+1,n]$. This contradicts the connectivity of $G[S,T]$.

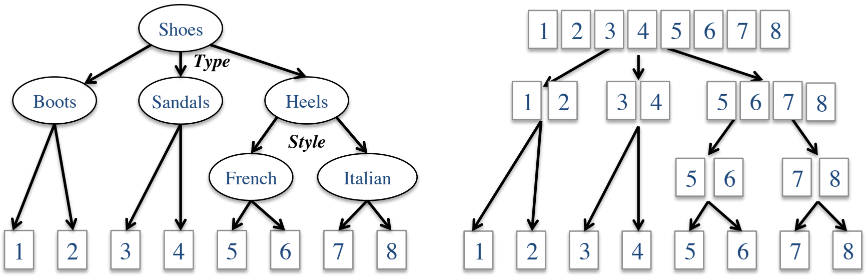
Since $\Phi(S,T) = [\min(S)] \cap \cup_{j\in T} C_j$, and we have proven that $\pi\langle \cup_{j\in T} C_j\rangle$ is an interval, we conclude that the image of Φ is a collection at most n^3 distinct subsets of $[n]$. For the special case of downward substitution, i.e., $\Sigma = \{\pi\}$, $\Phi(S,T)$ is an interval of the form $[\alpha, \min(S)]$, meaning that the state space has a cardinality of $O(n^2)$. In this case, using an interval tree data-structure, it is known that the connected components can be computed in a running time of $O(K\log(K))$ (Samet 1990), leading to a total running time of $O(n^2K\log(K))$. □

3.4.2 Laminar properties

Definition 3.4.3. *A collection of consideration sets $\mathcal{C}$ is said to be* laminar *if, for any two customer-types $j, j' \in [K]$ such that $C_j \cap C_{j'} \neq \emptyset$, the consideration sets are nested, i.e., $C_j \subseteq C_{j'}$ or $C_{j'} \subseteq C_j$.*

Elimination-by-Aspect. Such laminar structures arise in Elimination-by-Aspect (EBA) choice-making processes, which were first introduced by Tversky (1972a,b). To this end, we assume that the feature space is discrete, i.e., without loss of generality $x^{(i)} \in \{0,1\}^d$, and each screening rule on feature $e \in [d]$ is expressed as a constraint of the form $x^{(i)}_e = t$ with a cut-off level $t \in \{0,1\}$. EBA models assume that a customer picks features iteratively, entailing a random sequence $e_1,\ldots,e_M \in [d]^M$, where M is random. At each step $k \in [M]$, the customer selects a level t_{e_k}, and eliminates all products i not satisfying $x^{(i)}_{e_k} = t_{e_k}$. The sequence of features could be deterministic (this is known as the lexicographic order) or random. One probabilistic structure used to describe these processes in related models rests on a tree structure (Tversky and Sattath 1979): the next feature chosen by an individual in the sequence of eliminations is *deterministic* conditional to the prefix of levels that he chose prior. That is, e_k is a deterministic function of $(e_1, t_{e_1}), \ldots, (e_{k-1}, t_{e_{k-1}})$. Assuming this property, it can be verified that the corresponding distributions over consideration sets necessarily have a laminar support (see Figure 3-2 for a pictorial representation).

Figure 3-2: Example of an Elimination-by-Aspect screening process and the corresponding laminar tree: *shoes* category, with features *type* and *style*.



The following theorem suggests that the induced intervals structure defined in Section 3.4.1 is rather general as it subsumes laminar consideration sets as a special case.

Theorem 3.4.4. *The class of laminar consideration sets is a special case of induced intervals. The corresponding running time complexity is $O(n^2K^2)$.*

Proof. Let $\mathcal{C}$ denote a laminar consideration set system. Without loss of generality, we may assume that each consideration set corresponds to a unique customer-type (otherwise if two customer-types share the same consideration set, we represent them by a single type and aggregate their arrival probabilities) and there exists a consideration set comprised of all products $[n]$ in $\mathcal{C}$ (its arrival probability can be set to 0). We seek to prove that $\mathcal{C}$ is a family of intervals if the products are numbered according to some appropriate permutation $\pi : [n] \rightarrow [n]$.

Laminar tree. It is known that any laminar collection of subsets admits a rooted tree representation (Edmonds and Giles 1977). That is, we can build an directed tree (V, E), wherein each customer-type is represented by a single node, i.e., $V = [K]$, and there exists a directed edge $(j, k) \in E$ if k is the customer-type with a maximal consideration set contained in C_j. In other terms, we have $C_k \subset C_j$ and there exists no other $l \in [k]$ such that $C_k \subset C_l \subset C_j$. The root corresponds to the customer-type with consideration set $[n]$.

Depth first order. Now, for any $j \in K$, we define $o(j)$ as the offspring of node j in (V, E). Also, we introduce the list of products $s(j)$ formed by the difference between C_j and the products associated with the children of j, i.e. $s(j) = C_j \setminus \cup_{j' \in o(j)} C_{j'}$. Next, the permutation π is defined through the ranked list of products obtained as a concatenation of the lists $s(j)$ in a depth-first traversal of the laminar tree. It can be proven inductively that $\pi\langle C_j \rangle$ is an interval for any $j \in [K]$. Indeed, if a node j is a leaf of the laminar tree, then $s(j) = C_j$, and the concatenation procedure preserves the connectivity of $s(j)$. The inductive argument proceeds from the following observations.

1. By definition of s, for any given customer-type $j \in [K]$, $C_j = (\cup_{j' \in o(j)} C_{j'}) \cup s(j)$.

2. The collections of products associated with the children nodes are examined consecutively in a depth-first traversal. Hence, the inductive hypothesis implies that $\pi\langle \cup_{j' \in o(j)} C_{j'} \rangle$ is an interval.

3. Since this interval gets concatenated to $s(j)$, observation 1 above implies that $\pi\langle C_j \rangle$ is an interval.

□

Weakly laminar consideration sets. To demonstrate the generality of our algorithmic approach, we now introduce an extension of this model, dubbed *weakly laminar* consideration sets, which does not reduce to induced intervals, but still admits a polynomial running time guarantee.

Definition 3.4.5. *A collection of consideration sets $\mathcal{C}$ is said to be weakly laminar if any two consideration sets that intersect are nested up to the maximal product of their intersection. That is, for any customer-types $a, b \in [K]$ such that $C_a \cap C_b \neq \emptyset$, if $i = \max(C_a \cap C_b)$, then, either $C_a \cap [i] \subseteq C_b \cap [i]$ or $C_b \cap [i] \subseteq C_a \cap [i]$.*

This model captures the *conjunction* of any laminar consideration sets with any arbitrary screening rule, such as the budget and quality constraints mentioned in Section 3.4.1. In addition, it subsumes (strictly) other choice models in related literature, notably the above-mentioned *downward substitution model*, as well as the *out-tree* model proposed by Honhon et al. (2012).

Theorem 3.4.6. *Under weakly laminar consideration sets, the dynamic program runs in time $O(n^2K^2)$.*

Proof. To analyze the size of the state space $|\mathcal{S}|$ under this model, we first exhibit a structural property satisfied by the recursion, namely the existence of a 'maximal' consideration set with respect to some well-chosen inclusion order, in each connected subgraph examined by the recursion.

Lemma 3.4.7. *Assume that $(S, T) \in \mathcal{S}$ is a subproblem generated by the recursion. Then, there exists a customer-type $j^* \in T$ such that $C_j \cap [\min(S)] \subset C_{j^*} \cap [\min(S)]$ for any customer-type $j \in T$.*

This property is proven in Appendix A.1. Consequently, we can upper bound $|\Phi\langle\mathcal{S}\rangle|$. Let (S, T) be a subproblem in the state space $\mathcal{S}$. By Lemma 3.4.7, we obtain

$$\Phi(S,T) = \bigcup_{j \in T} \left(C_j \bigcap [\min{(S)}] \right) = C_{j^*} \bigcap [\min{(S)}]$$

For a fixed value of $\min(S)$, there are at most K subsets $\Phi(S, T)$, meaning that $|\Phi\langle\mathcal{S}\rangle| \leq nK$. □

3.4.3 Disjunctive consideration sets

The consideration set models discussed in the previous sections proceed from a conjunction of screening rules. As mentioned in Section 3.2, another decision-making model proposed in the marketing literature posits that the consideration sets are formed in a *disjunctive* fashion.

Definition 3.4.8. *For any $d \in \mathbb{N}$, a collection of consideration sets $\mathcal{C}$ is said to be d-disjunctive if the feature space is d-dimensional and all consideration sets in $\mathcal{C}$ are generated as a disjunction of screening rules. That is, each customer-type $j \in [K]$ is characterized by a cut-off vector denoted by $\boldsymbol{t}^{(j)} \in \mathbb{R}^d$, such that*

$$C_j = \left\{ i \in [n] : \left(x_1^{(i)} \geq t_1^{(j)} \right) \vee \cdots \vee \left(x_d^{(i)} \geq t_d^{(j)} \right) \right\} \ .$$

We now prove that the size of the state space $|\mathcal{S}|$ is polynomially bounded for a fixed parameter d. Since the d-disjunctive model can replicate arbitrary consideration set structure $\mathcal{C}$ for a large enough d, the next theorem expresses an explicit tradeoff between modeling power and tractability. In practice, one would expect that customers make use of few screening rules (Hauser et al. 2009).

Theorem 3.4.9. *Under d-disjunctive consideration sets, the dynamic program has a running time of $O(n^2 K^{d+1})$.*

3.5 General Dynamic Programming Formulation

In this section, we relax the assumption that Σ is a singleton and describe a dynamic program that applies to arbitrary preference list distributions. The key ingredients of the algorithm remain unchanged. Specifically, products are processed sequentially, which entails a decomposition of the graph representation into increasingly refined connected components, in a divide-and-conquer fashion. However, unlike the unique-ranking case, the processing order does not necessarily coincide with the customer's preference order. Thus, it is not immediate which subset of customer-types gets allocated to a given product at the time a DP decision is made. As a result, the DP action space needs to be enlarged to account for any feasible allocation of a product to a subset of customer-types. At face value, there are exponentially many potential allocations, and the approach appears to be subject to the curse of dimensionality. We work around this difficulty by proposing an auxiliary algorithm, used at each step of the recursion, that can yield tractable solutions.

Processing order. We begin by defining a *processing order* σ on the products, according to which the dynamic program makes sequential decisions (or actions). The correctness of the dynamic program does not depend on σ although, as shown in next subsections, an appropriate choice of σ may significantly reduce the running time complexity. Here, σ is chosen as an arbitrary permutation and the products are numbered accordingly (i.e., product 1 is processed first, and so on) such that the reference to the processing order is made implicit throughout the present section.

State space and value function. The state space is described by the parameters $(S, T, \boldsymbol{L})$, where S is a subset of products in $[n]$, T is a subset of customer-types in $[K]$ and $\boldsymbol{L} \in \mathbb{Z}_+^K$ is a nonnegative integer valued vector, named the *truncation vector*. We let $J(S, T, \boldsymbol{L})$ be the maximum expected revenue that can be attained from customer-types in T using an assortment within products in S, and assuming that, each customer-type $j \in T$ is willing to purchase only products of rank at most L_j within his consideration set. That is, customer-type j will only purchase products in the set $C_j(L_j) = \{i \in C_j : \sigma_j(i) < L_j\}$. (Recall that each customer-type j is associated with a ranking function $\sigma_j \in \Sigma$.) We note that only the T-coordinates of $\boldsymbol{L}$, i.e., the sub-vector $\boldsymbol{L}[T]$, matter in the definition of $J(S, T, \boldsymbol{L})$. However, to lighten the notation we use the entire vector and assume that the unnecessary coordinates are set to 0.

DP bipartite graph. Similar to the unique-ranking case, we define the bipartite graph G that has a node, for each product $i \in [n]$, on one side, and a node, for each customer-type $j \in [K]$, on the other side. There is an edge between a product node and customer-type node if the preference list of the latter contains the former. Each subproblem of the state space $(S, T, \boldsymbol{L})$ is associated with the subgraph $G_{\boldsymbol{L}}[S, T]$ with *(i)* product nodes in S; *(ii)* customer-type nodes in T; *(iii)* there exists an edge between any $i \in S$ and $j \in T$ if $i \in C_j(L_j)$. Similar to the unique-ranking case, the connected components of the subgraph capture a decomposition into independent instances. The proof, in the same spirit as that of Lemma 3.3.1, is omitted.

Lemma 3.5.1. *For each subproblem $(S,T,\boldsymbol{L})$, assuming that the connected components of $G_{\boldsymbol{L}}[S,T]$ are described by the collection of subgraphs $(G_{\boldsymbol{L}}[S_u,T_u])_{u\in[r]}$ where S_u denotes a subset of product nodes in S and T_u is a subset of type nodes in T, then $J(S,T,\boldsymbol{L}) = \sum_{u=1}^{r} J(S_u,T_u,\boldsymbol{L})$.*

Dynamic programming decisions and graph operations. We consider a subproblem $(S,T,\boldsymbol{L})$ and i is the next product to be processed $i=\min(S)$. We define $T(i)\subseteq T$ as all customer-types whose preference list contains i, i.e., $T(i)=\{j\in T : i\in C_j(L_j)\}$.

Assume we decide to allocate product i to a subset of customers $V\subseteq T(i)$, meaning that i is the most preferred product made available to the customer-types in V. We describe below some natural operations on the graph $G_{\boldsymbol{L}}[S,T]$ to enforce the decision of allocating product i to customer-types V. In particular, we make sure that the decision to satisfy V with product i is consistent with future decisions, and that it is feasible irrespective of the subsequent decisions. Specifically, we perform the following operations on the bipartite graph:

1. *T-nodes deletion.* Because they are already satisfied with a product, the nodes corresponding to consumer-types V should be discarded, we define $T^{(V)}=T\setminus V$ as the remaining customer-types.

2. *S-nodes deletion.* For each satisfied customer-type $j\in V$, we should remove from S the nodes of all items he prefers more; indeed, these products cannot be stocked in the assortment otherwise they would have been chosen by customer-type j over product i. Thus we define $S^{(V)}$ as the residual set of products:

$$S^{(V)} = S\setminus \bigcup_{j\in V}\{x\in C_j(L_j) : \sigma_j(x)<\sigma_j(i)\}\ , \tag{3.2}$$

3. *Edges deletion.* Finally, if a customer-type whose preference list contains product i, is not allocated to this product, he can only purchase a product product he prefers more at some later point of the recursion. As a result, we need to

truncate his preference list by updating the vector $\boldsymbol{L}$:

$$\forall j \in T^{(V)}, \quad L'_j = \begin{cases} L_j & \text{if } j \notin T(i) \\ \min(L_j, \sigma_j(i)) & \text{otherwise} \end{cases} . \tag{3.3}$$

We can easily verify the correctness of the above graph operations. That is, the expected revenue obtained by summing the immediate-reward, formed by the allocation of product i to customer-types V, with the reward-to-go, generated by the subproblem associated with the residual subgraph $G_{\boldsymbol{L'}}[S^{(V)}, T^{(V)}]$, is feasible. We now formally describe the recursion.

Base case. If we set $S = [n]$, $T = [K]$ and $L_j = n + 1$, then $J(S, T, \boldsymbol{L})$ reflects the original problem we are interested to solve. Using the Lemma 3.5.1, an optimal assortment is obtained by solving independently the subproblems associated with each connected component of G. From this point on, connectivity is an invariant of the subproblems examined by the recursion.

Recursive formula. We consider the subproblem $(S, T, \boldsymbol{L})$ such that $G_{\boldsymbol{L}}[S, T]$ is a connected subgraph. Recall that i denotes the next product to be processed (the minimal element of S) and $T(i)$ are all customer-types who consider product i. The decision made by the dynamic program consists in the subset of customer-types $V \subseteq T(i)$ allocated to product i. Without loss of generality, we can assume that an empty allocation $V = \emptyset$ means that he product i is not stocked in the assortment. In this case, the residual subgraph is $G_{\boldsymbol{L}}[S \setminus \{i\}, T]$, which decomposes into the connected components $(G_{\boldsymbol{L}}[S_u, T_u])_{u \in [r]}$. Each corresponding subproblems is solved independently according to Lemma 3.5.1, generating a total revenue of $\sum_{u=1}^{r} J(S_u, T_u, \boldsymbol{L})$.

For each choice of $V \subseteq T(i)$, where $V \neq \emptyset$, the allocation generates an immediate-reward $P_i \cdot \sum_{j \in T} \lambda_i$. Next, we consider the residual subgraph $G_{\boldsymbol{L'}}[S^{(V)}, T^{(V)}]$ after performing the operations previously described. Namely, we remove the most preferred products according to Equation (3.2) while we delete edges according to $\boldsymbol{L'}$ defined in Equation (3.3), where the vector $\boldsymbol{L'}$ does not depend on the choice of the allocation $V \subseteq T(i)$. Using Lemma 3.5.1, the subgraph $G_{\boldsymbol{L'}}[S^{(V)}, T^{(V)}]$ can be decomposed into its connected components $(G_{\boldsymbol{L'}}[S_u^{(V)}, T_u^{(V)}])_{u \in [r(V)]}$. Therefore, the optimality conditions yield the following the recursive formula:

$$J(S,T,\boldsymbol{L}) \ = \ \max \left(\sum_{u=1}^{r} J(S_u, T_u, \boldsymbol{L}) \ , \ \max_{V \subseteq T(i)} \quad P_i \cdot \sum_{j \in V} \lambda_j \ + \sum_{u=1}^{r(V)} J\left(S_u^{(V)}, T_u^{(V)}, \boldsymbol{L'}\right) \right) \tag{3.4}$$

Preliminary complexity analysis. In a naive implementation of the algorithm, one would solve the problem for all possible tuple $(S, T, \boldsymbol{L})$. Similar to the dynamic program presented in Section 3.3, the effective computational tree is in fact comprised of a much smaller fraction of the state space. However, in contrast to the unique-ranking case, the recursive formula (3.4) describes a maximization problem over exponentially many allocations $V \subseteq T(i)$, each associated by a family of descendant subproblems of the form $(S_u^{(V)}, T_u^{(V)}, \boldsymbol{L'})$. As a result, we cannot readily leverage this formula to build the computational tree. In addition, even if one tightly characterizes the state space, it is still not obvious how to solve efficiently the optimization problem described by (3.4).

The marginalized algorithm. We address the difficulty raised at the end of the previous section by proposing an efficient algorithm to generate the computational tree and solve the recursive formula (3.4), while avoiding an enumeration over all allocations $V \subseteq T(i)$. Because the detailed exposition is rather technical, we only provide the high-level idea, and state the resulting complexity analysis. The specifics of this auxiliary algorithm are detailed in Appendix A.2. The crux of our approach is to observe that the descendant subproblems in equation (3.4) are not necessarily distinct and may very well be equivalent: there is potentially overlap in the offspring generated across all choices of $V \subseteq T(i)$. Most of the redundancy that happen in a brute force enumeration can in fact be eliminated (up to a polynomial factor) by *marginalizing* the allocation decision. That is, the choice of the allocation $V \subseteq T(i)$ is broken-down into a sequence of binary decisions, each of which applies to a single customer-type in $T(i)$. These binary allocation decisions are made sequentially, entailing refined connected subgraphs and potentially new descendant subproblems in the computational tree. By constructing and updating an appropriate data-structure, we avoid the unnecessary exploration of equivalent subproblems. As a result, the marginalized algorithm runs in time polynomial in n, K and $|\mathcal{S}|$, yielding the following complexity result.

Proposition 3.5.2. *The running time complexity of the marginalized dynamic program is polynomial in the size of the state space $\mathcal{S}$ and the input size. In addition, in the worst case, we have $|\mathcal{S}|\leq n \cdot 2^n$.*

3.6 Modeling the Ranking Heterogeneity

. We now study the tractability of distributions over preference lists which combine consideration set heterogeneity along with ranking heterogeneity. Our analytical results extend the computational settings studied in Section 3.4.

3.6.1 Similar rankings

In this section, we relax the unique-ranking assumption, and derive parametric computational bounds for the consideration set models studied in Section 3.4 when the rankings are similar, i.e., Σ is formed by small perturbations of a central permutation σ. Namely, assuming that S_n designates the set of all permutations of $[n]$, we let $B(\sigma, d)$ designate the L_∞-ball of radius d centered on σ. That is,

$$B(\sigma, d) = \{\sigma' \in S_n : \forall i \in [n] |\sigma'(i) - \sigma(i)|\leq d\} \ .$$

This definition implies that for any permutations $\sigma_1, \sigma_2 \in B(\sigma, d)$, two products $i, j \in [n]$ such that $|\sigma(i) - \sigma(j)|\geq 2d$ necessarily have the same relative order in σ_1 and σ_2. In other terms, only local 'swaps' may occur between products at distance less than $2d$. This structure is somewhat similar to the d-sorted pricing structure proposed by Jagabathula and Rusmevichientong (2016).

The next theorem, proven in Appendix A.3, asserts that, for a fixed parameter d, the state space complexity associated with unique-ranking $\Sigma = \{\sigma\}$ is preserved up to a polynomial factor under the generalization $\Sigma = B(\sigma, d)$. In particular, the polynomial running time guarantees established in Section 3.4 carry over to $\Sigma = B(\sigma, d)$. Again, this result permits a parametric tradeoff between modeling power and tractability.

Theorem 3.6.1. *Let $\mathcal{S}(\mathcal{C}, \sigma)$ denote the state space under a collection of consideration sets $\mathcal{C}$ and a unique-ranking function $\Sigma = \{\sigma\}$. Then, the size of the state space of the general dynamic program with processing order σ under the consideration sets $\mathcal{C}$ with rankings $\Sigma = B(\sigma, d)$ is at most $2^{4d-2} \cdot |\mathcal{S}(\mathcal{C}, \sigma)|$.*

3.6.2 Quasi-convex preference lists

We now study a class of preference list distributions that allows for high levels of heterogeneity in the ranking decisions, but the ranking functions exhibit a *quasi-convex* structure.

Definition 3.6.2. *Suppose that the product indices are numbered according to a* central *permutation σ over products. A distribution over preference lists belongs to the quasi-convex model if the consideration sets $\mathcal{C}$ are intervals and the ranking functions Σ are quasi-convex. That is, for all $j \in [K]$, there exists $i \in C_j$ such that σ_j is decreasing over $[1, i] \cap C_j$ and increasing over $[i, n] \cap C_j$.*

The quasi-convex property captures several common preference patterns. To flesh out this model with practical examples, suppose that the consideration sets are formed as a conjunction of the screening rules relative to price and perceived quality described in Section 3.4.1. The quasi-convex family simultaneously captures a variety of ranking behaviors: customers can be price-driven, or quality-driven, or maximize any price/quality ratio function which is quasi-convex in price.

It is worth noting that the quasi-convex model subsumes the one-dimensional *locational choice model* (Lancaster 1966, 1975). In the latter model, customer-types and products are each represented by a scalar value, and a customer-type picks the closest product to him made available in the assortment (i.e., with minimal absolute distance between their respective scalars). It is not difficult to show that the ranking functions arising from this model are quasi-convex with respect to the central permutation formed by increasing scalars.

Observe that the quasi-convex model substantially 'enriches' the degree of freedom of the distributions up to $O(2^n)$ - in comparison to the $O(n^2)$ parameters of the intervals model or the $O(n^3)$ parameters associated with the locational model (see Claim A.4.1 proven in Appendix A.4).

Theorem 3.6.3. *Under the quasi-convex model with central permutation σ, the dynamic program with processing order σ has a state space of size $O(n^3)$.*

Proof. We construct an injective mapping from any connected subgraph generated along the recursion, i.e., belonging to the computational tree, onto 3-tuples of products. Specifically, Ψ maps any subproblem $(S, T, \boldsymbol{L})$ to the tuple (a, b, c) where (a, b) is the ordered pair of the last products stocked along the recursion before generating $(S, T, \boldsymbol{L})$ while c is the next product to be processed in S, i.e., $c = \min(S)$. To prove that this mapping is injective, we consider $(S_1, T_1, \boldsymbol{L})$ and $(S_2, T_2, \boldsymbol{L}') \in \mathcal{S}$ two subproblems of the computational tree such that

$$(a, b, c) = \Psi(S_1, T_1, \boldsymbol{L}) = \Psi(S_2, T_2, \boldsymbol{L}') \ . \tag{3.5}$$

Now, assume ad absurdum that $T_1 \backslash T_2 \neq \emptyset$. Without loss of generality, we can pick a customer-type j in $T_1 \setminus T_2$ whose consideration set C_j has been truncated, meaning that $L_j < n + 1$. Indeed, $T_1 \setminus T_2$ would otherwise be comprised of customer-types not affected by the decisions made at the parent nodes of the computational tree, relative to products in $[b]$. But this contradicts that at least one node in $T_1 \setminus T_2$ gets disconnected from the subgraph $G_{\boldsymbol{L}'}[S_2, T_2]$ along the recursion.

Since $L_j < n + 1$, there exists a product in $C_j \cap [c - 1]$, which is processed before attaining subproblem $(S_1, T_1, \boldsymbol{L})$. By connectivity, $C_j(L_j) \cap S_1$ is not empty, meaning that C_j contains a product in $[c, n]$ as well. Given that C_j is an interval, this implies that $c \in C_j$. Also, the preference order between the products $\{a, b, c\}$ is given by

$\sigma_j(c) < \sigma_j(b) < \sigma_j(a)$. Indeed, the ranking function being quasi-convex, any product with index larger than b would otherwise be less preferred than a and b, and as a result the truncated consideration set $C_j(L_j)$ would not intersect with S_1. This shows that customer-type j prefers product c over all products in $[b]$. Since $j \notin T_2$ and $c \in C_j$, the customer-type node j necessarily gets disconnected from the subgraph $G_{\boldsymbol{L'}}[S_2, T_2]$ along the recursion through an allocation to some product $i \in [b]$. However, had this allocation been decided at a parent node of the computational tree, all products that customer-type j prefers over i would have been discarded by now. In particular, product c would not belong to S_2, which contradicts equation (3.5).

Finally, we remark that the truncation of preference lists only depends on the last product being stocked. Indeed, as previously shown, for any remaining preference lists $j \in T_1$ such that $L_j < n+1$, then product b lies in C_j and $\sigma_j(b) < \sigma_j(a)$. Quasi-convexity induces that b is preferred over any other product stocked before a. Thus, $L_j = \sigma_j(b) = L'_j$. We conclude by observing that

$$S_1 = \bigcup_{j \in T_1} C_j(L_j) = \bigcup_{j \in T_2} C_j(L'_j) = S_2 \ .$$

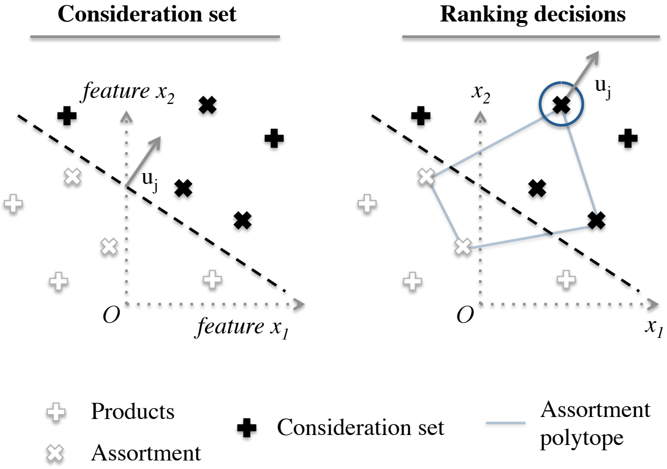
□

3.6.3 Two-feature compensatory model

We consider a preference list-based model where the screening rules are combined in a compensatory fashion (Einhorn and Hogarth 1975, Dawes 1979). Here, low levels on a given feature can be offset by high levels on other features as discussed in Section 3.2. Specifically, preference lists are formed according to utility maximization, as illustrated by Figure 3-3.

Definition 3.6.4. *Suppose that the feature space has dimension $d = 2$. An instance belongs to the two-feature compensatory model if each customer-type $j \in [K]$ can be described by a utility vector $\boldsymbol{u}^{(j)} \in \mathbb{R}^2$ and a cut-off level t_j such that* (i) *C_j contains all products which utility is above the cut-off t_j, i.e., $C_j = \{i \in [n] : \boldsymbol{u}^{(j)} \cdot \boldsymbol{x}^{(i)} \geq t_j\}$, and*

Figure 3-3: Consideration sets and ranking decisions driven by linear utility maximization in a two-featureal feature space.



(ii) *for any pair of products* $i, k \in [n]$, $\sigma_j(i) < \sigma_j(k)$ *if and only if* $\boldsymbol{u}^{(j)} \cdot \boldsymbol{x}_k < \boldsymbol{u}^{(j)} \cdot \boldsymbol{x}_i$ *(we assume there are no ties between products).*

By exploiting the geometric structure of this model, we prove that the state space is of polynomial size under the class of preference list distributions described by the two-feature linear model. The proof, detailed in Appendix A.5, is of same spirit as that of Theorem 3.6.3: we construct an injective mapping of subproblems in $\mathcal{S}$ onto the last dynamic programming decisions in the computational tree.

Theorem 3.6.5. *Under the two-feature compensatory model, for any arbitrary processing order, the size of the state space is of* $O(n^3 K^2)$.

3.7 Computational performance

In this section, we showcase the computational efficiency of the proposed dynamic program, through empirical comparisons with a state-of-the-art Integer Programming (IP) solver. We generate synthetic instances pertaining to the quasi-convex model, as well as instances of the unique-ranking model with arbitrary consideration sets.

3.7.1 Benchmark: Integer Programming formulation

The assortment optimization problem can be formulated as $0-1$ binary program. We define the binary decision variables y_i to decide whether a product is added to the assortment, $x_{i,j}$ encodes the assignment of product $i \in C_j$ to customer-type $j \in [K]$. The problem is formulated as follows:

$$
\begin{aligned}
\text{max} \quad & \sum_{i=1}^{n}\sum_{j=1}^{K} P_i \cdot \lambda_j \cdot x_{i,j} & & \\
\text{s.t.} \quad & x_{i,j} \leq y_i & \forall (i,j) \in [n] \times [K] & \qquad (3.6)\\
& x_{i,j} + y_l \leq 1 & \forall j \in [K], l \in C_j \text{ and } i \in \{x \in C_j : \sigma_j\left(l\right) < \sigma_j\left(i\right)\} & \qquad (3.7)\\
& \sum_{i \in C_j} x_{i,j} \leq 1 & \forall j \in [K] & \qquad (3.8)\\
& x_{i,j}, y_i \in \{0,1\} \ , & &
\end{aligned}
$$

where the coupling constraints (3.6) enforce that a customer may only pick a product made available in the assortment while the inequalities (3.7) ensure that a given customer-type could only choose the highest rank product made available to him. Finally, the constraints (3.8) mean that at most one product is assigned to each customer. The additional constraints (3.8) tighten the relaxation of the binary program. It is worth noting that that similar formulations were introduced prior to this work by McBride and Zufryden (1988) and Anupindi et al. (2009). This integer program (IP) is implemented on a commercial solver GUROBI (Gurobi Optimization 2015), which arguably combines state-of-the-art methodologies and implementation.

3.7.2 Computational set-up

The experiments are conducted using a MacBook Pro with processor 2.5 GHz Intel Core $i5$ (two cores). Our dynamic program is implemented using the programming language Julia. The commercial solver GUROBI (v.6.5) is run in parallel mode. We impose termination when the incumbent solution has an optimality gap of 1%, or after the running time reaches 1000 seconds for computational convenience. In contrast, our algorithm provides *exact solutions* for all instances. We run two series of experiments with different generative models. In the former, we generate instances of the quasi-convex model described in Section 3.6, arguably one of the 'richest' consider-then-choose model discussed in previous sections that admits a provable polynomial running time guarantee. In the latter, we compare the algorithms on generic instances with unique-ranking preferences, not pertaining to any specific structure of consideration sets. The consideration sets arise from i.i.d Bernoulli trials with a parameter $\alpha \in (0, 1)$. In Appendix A.7, we describe more precisely our generative models and provide additional details on the implementation.

Table 3.2: Runtime of our algorithm (DP) against the commercial solver (IP) under the quasi-convex preference model.

Parameters		Average runtime (s)		Coeff. of var (%)	
n	K	DP	IP	DP	IP
50	500	0.9	45.9	17.0	47.4
50	1000	1.2	398.5	6.2	52.4
50	2000	1.8	777.1	$< 10^{-3}$	46.2
50	2500	2.4	$> 10^3$	$< 10^{-3}$	-
100	2500	16.8	$> 10^3$	$< 10^{-3}$	-
200	2500	138.9	$> 10^3$	$< 10^{-3}$	-

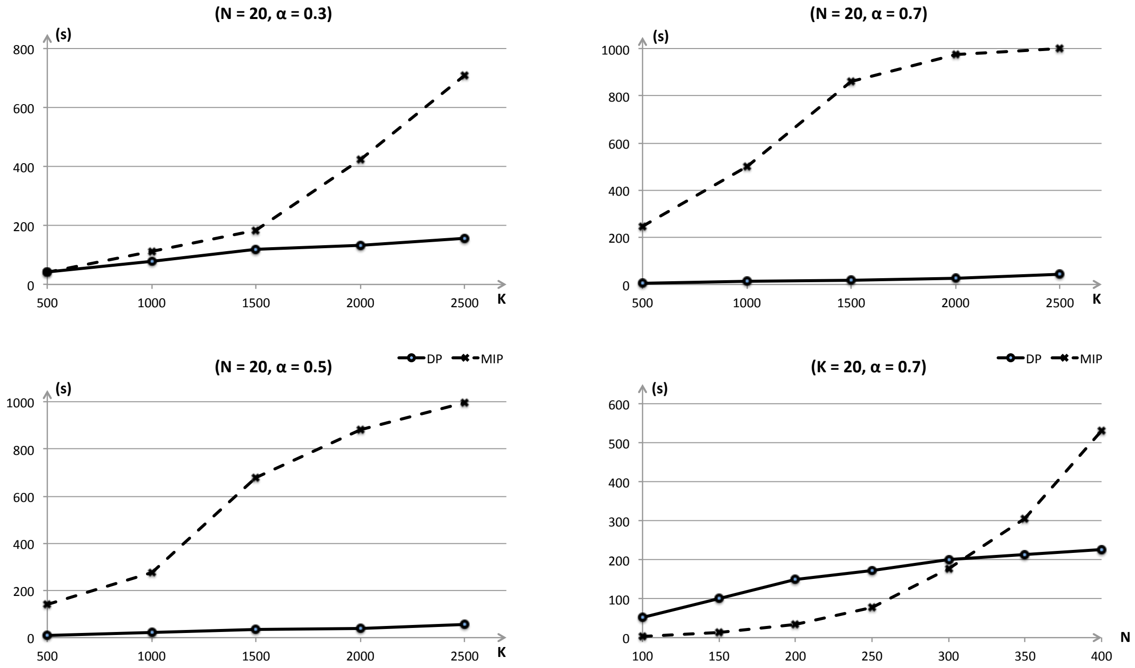
The IP is terminated after 1000 seconds. Each entry is obtained by sampling 50 instances unless the average running time exceeds 800 seconds, in which case we sample 20 instances.

3.7.3 Numerical results

Our numerical results indicate that our algorithmic approach substantially outperforms the IP solver in several regimes of parameters (see Table 3.2 and Figure 3.1). The dynamic programming approach shows more running time stability across instances due to its combinatorial nature, while one potential shortcoming of the IP approach for practitioners resides in the large variability of running time across instances.

Under the quasi-convex model. As shown in Table 3.2, the dynamic program dominates the commercial solver by an order of magnitude. The IP approach scales unfavorably with the number of customer-types K and it becomes intractable for large scale instances (e.g., with 200 products) where the dynamic program is still very efficient.

Figure 3-4: Average runtime of our algorithm (DP) against the commercial solver (IP) on synthetic instances.



Note that the asymptotic complexity of the IP is not captured here since we impose termination after 1000 *seconds. The running time is averaged over* 50 *instances. Recall that* α *is the Bernoulli parameter that controls the size of the consideration sets.*

Under arbitrary consideration sets. Recall that in this general setting the problem is NP-hard even to approximate within $O(n^{1-\epsilon})$. In several cases, our approach dominates the IP as shown in Figure 3-4.

Similar to the quasi-convex model, the IP solver scales poorly with the number of customer-types. That is, for a fixed number of products ($n = 20$), the running time of the IP is highly affected by the number of customer-types K. The difference between the algorithms is more pronounced for larger consideration sets (larger Bernoulli parameter α). On the other hand, as one would expect, the dynamic program is less efficient when $n >> K$, since our algorithm enumerates over product stocking decisions. The results obtained for large consideration sets ($\alpha = 0.7$) suggest that the dynamic program could asymptotically dominate the IP solver in this regime.

The observed computational efficiency proceeds from the state space collapse performed by our algorithm. When comparing our approach to a "naive" recursion, the state space is reduced by a factor ranging between 75% to over 99% (see Table A.1 in Appendix A.8).

3.8 Predictive Experiments

Practical applications of choice modeling, such as the assortment optimization problem studied here, begin with *transactional data*. The generic approach is to fit a specific type of choice model to this data and then employ an assortment optimization algorithm designed for that choice model. As such, the choice model employed must strike a balance between its ability to fit the data on the one hand, and admit efficient algorithms for assortment optimization on the other. In this regard it is well known that the MMNL model has the ability to represent any choice model satisfying the strong axiom of revealed preferences, that is, arbitrary distributions over preference lists (McFadden and Train 2000). Of course, this expressive power comes at a price: assortment optimization under the MMNL model is difficult in all but a restricted set of cases. Specifically, Désir and Goyal (2014) provide an algorithm for assortment optimization under the MMNL model whose complexity scales exponentially with the number of customer segments[2]. Consequently, optimization is practical only for a mixture over a relatively 'small' number of customer segments (a notion we will make precise shortly). In summary, one may regard MMNL models with a small number of customer segments as a valid alternative to the models (and corresponding algorithms) we consider in this chapter. The goal of this section is

[2] Their algorithm constitutes a *fully polynomial-time approximation scheme* for a fixed number of customer segments

to flesh out this comparison. Specifically, we consider the following experiments on synthetic and industry data:

1. *Synthetic data from an MMNL model:* Using a synthetic dataset generated from an MMNL model with a relatively large number of customer segments, we fit two types of models to this data: *(i)* an MMNL model with a small number of customer segments and *(ii)* the quasi-convex consider-then-choose model studied in Section 3.6.2. We show that in several cases the latter model provides a better fit to the data (out-of-sample) under a variety of metrics. In particular, we show that the quasi-convex model does an excellent job of capturing choice patterns that may have arise from an MMNL model.

2. *Synthetic data from a consider-then-choose model:* As a counterpart to synthetic data from an MMNL model, we consider fitting both types of model in the experiment above to synthetic data generated this time from a simple, intervals-based consider-then-choose model. As one would expect, the quasi-convex model provide a better fit by a large margin. In particular, we show that MMNL models with a small number of customer segment do not do an adequate job of fitting choice data arising from consider-then-choose behavior.

3. *Real industry data:* Using transactional data across a panel of hundreds of thousands of customers in three distinct product categories (containing hundreds of products) collected by an industry partner, we again run the same experiment, and evaluate predictive power on a holdout sample. Again we show that the quasi-convex consider-then-choose model provides a very significant improvement in predictive accuracy on the hold out set. This improvement can be as high as 60% in certain categories, and never lower than 4% – a striking improvement.

In the sequel, we designate by MMNL(c) the class of mixtures with c customer segments. The MMNL instances (and the MNL as a special case) are parametrized by the preference weights $w_{i,j} \in \mathbb{R}^+$ where $i, j \in [n] \times [c]$, along with the probability vector $(\mu_1, \ldots, \mu_c)$ of the mixture. Here n is the number of products and c is the number of customer segments. With this definition at hand, the purchase probability for product i in an assortment $\mathcal{A} \subseteq [n]$ is, under the MMNL(c) model, expressed as:

$$\Pr\left[i|\mathcal{A}\right] = \sum_{\alpha=1}^{c} \mu_\alpha \cdot \frac{w_{i,\alpha}}{1 + \sum_{j \in \mathcal{A}} w_{j,\alpha}} \ .$$

3.8.1 Estimation Methodology

To calibrate these models with the data, we leverage standard estimation methods proposed in the literature (McFadden 1973, Talluri and Van Ryzin 2006, van Ryzin and Vulcano 2014, Bertsimas and Mišic 2015). Specifically, we employ a column generation algorithm to generate the quasi-convex preference lists, and use maximum likelihood estimation for the MMNL family. Interestingly, in contrast with general nonparametric choice models, we show that the column generation step can be solved in polynomial time under the quasi-convex model.

Recall that the data takes the form of a collection of assortments $\{\mathcal{A}_1, \ldots, \mathcal{A}_s\}$, with corresponding purchase probability p_{ij} of product i in the assortment $\mathcal{A}_j$, for each $i \in [n]$ and $j \in [s]$. Also, we let $\mathcal{P}$ designate the probability vector obtained by flattening the matrix $(p_{ij})_{ij}$ in column-major order.

Quasi-convex model. In order to calibrate the quasi-convex model with data, we make use of the column generation ideas developed in related literature by van Ryzin and Vulcano (2014) and Bertsimas and Mišic (2015). To this end, let $\mathcal{L}$ be the collection of *all* quasi-convex preference lists for a given instance and let m be the number of distinct such lists (the dependency of m on n has been made explicit by Claim A.4.1). To ease the notation, we assume that the no-purchase option is captured by an alternative in $[n]$. Hence, we introduce the observation tensor $\mathcal{O} = (\mathcal{O}_{i,j,k}) \in \{0,1\}^{[s]\times[n]\times[m]}$, where $\mathcal{O}_{i,j,k} = 1$ if the preference list $\mathcal{L}_k$ purchases product j in the assortment $\mathcal{A}_i$, and $\mathcal{O}_{i,j,k} = 0$ otherwise. In what follows, by abuse of notation, $\mathcal{O}$ will also designate the corresponding $s \cdot n$-by-m matrix (known as the mode-3 unfolding).

In order to estimate a probability distribution $\Lambda = (\lambda_1, \dots, \lambda_m)$ over the quasi-convex preference lists, ideally we would like to solve the following convex program:

$$\begin{array}{ll} \min & \|\mathcal{O} \cdot \Lambda - \mathcal{P}\|_1 \\ \text{s.t.} & \|\Lambda\|_1 \leq 1 \\ & \Lambda \geq 0 \ . \end{array}$$

Through standard techniques, the latter problem can be recast a linear program with $O(s \cdot n)$ equality constraints. However, the number of variables of the resulting linear program remains exponential. Indeed, although quasi-convex preference lists are fewer than the $(n+1)!$ potential lists, m still grows exponentially in the number of products according to Claim A.4.1. On the other hand, the number of equality constraints is small, thus, one could resort to a column generation procedure. This procedure has notably been developed and discussed in the paper by Bertsimas and Mišic (2015)– we refer the reader to this paper for a more detailed technical description. The algorithm alternates between solving a *master* problem and a *column generation* subproblem.

Specifically, at step $t \in \mathbb{N}$, given an incumbent (fixed) collection $\mathcal{L}_t \subseteq \mathcal{L}$ of quasi-convex preference lists, the master problem solves the ℓ_1-minimization program to find a distribution over $\mathcal{L}_t$ that best fits the data (note we could use other norms, such as ℓ_2). Next, the column generation subproblem attempts to identify a new quasi-convex list $L \in \mathcal{L} \setminus \mathcal{L}_t$ with lowest reduced cost. While the master problem can be cast as a linear program, the column generation step is general NP-hard (this problem subsumes the widely-studied rank aggregation problem, where data is formed by pairwise comparisons). Thus, prior literature has investigated heuristic procedures based on integer programming, local search methods or sampling. In contrast, the next claim shows that the column generation step can be solved in polynomial time. Specifically, we are given the reduced cost matrix $(h_{i,j})_{i\in[s],j\in[n]}$ (to wit, the reduced cost associated with the equality regarding the probability of "buying product j in the assortment $\mathcal{A}_i$"). For any list $L \in \mathcal{L}$, we define $I(i, j, L)$ as the binary function that indicates whether the preference list L buys product j in the assortment $\mathcal{A}_i$.

Lemma 3.8.1. *The rank aggregation problem, that consists in finding the optimal quasi-convex list $L \in \mathcal{L}$ to minimize the reduced cost $\phi(L) = \sum_{i=1}^{s} \sum_{j=1}^{n} h_{i,j} \cdot I(i, j, L)$, can be solved in time $O(n^3)$.*

Proof. Without loss of generality, we can pick a ranking permutation whose rank values range in $[1, n]$. The optimization problem is cast as a dynamic program, whose state space $\mathcal{S}$ is formed by the collection of 3-tuples (a, b, c) where $a, b, c \in [n]$. Specifically, we define $L(a, b, c) \in \mathcal{L}$ as the optimal quasi-convex list $L \in \mathcal{L}$ (i.e., with lowest reduced cost $\phi(L)$), whose consideration set is formed by the interval $[c, c + \max\{a, b\} - 1]$ with $\sigma_L(c) = a$ and $\sigma_L(c + \max\{a, b\} - 1) = b$ (σ_L is the ranking permutation associated with L). Finally, we let $F(a, b, c)$ be the corresponding total reduced cost.

Now, let (a, b, c) be a given state of the recursion, with $a < b$ and $b - a \geq 2$. Since $L(a, b, c)$ is quasi-convex, we necessarily have that, for every $d \in [c, c + b - a - 1]$, $\sigma_{L(a,b,c)}(d) = a + (d - c)$. In other words, the rank values associated with the interval of products $[c, c + b - a - 1]$ are already determined: they form the interval $[a, b - 1]$. Therefore, our next dynamic programming decision is to choose a product whose rank value is exactly $b + 1$. Due to the quasi-convex structure, there can be at most two options: either the leftmost unassigned product or the rightmost unassigned product, in the consideration set. That is, we can assign $\sigma^{-1}_{L(a,b,c)}(b + 1) = c + b - a$ or $\sigma^{-1}_{L(a,b,c)}(b + 1) = c + b - 2$. Letting $c_1 = c + b - a$, $c_2 = c + b - 1$, $\text{Ass}_1 \subseteq [s]$ be the collection of indices j so that $\mathcal{A}_j \cap [c_1, c_2] = \emptyset$ while $\mathcal{A}_j \cap [c, c_1 - 1] \neq \emptyset$, and $\text{Ass}_2 \subseteq [s]$ be the collection of indices j so that $\mathcal{A}_j \cap [c, c_2 - 1] = \emptyset$ while $c_2 \in \mathcal{A}_j$, we obtain the recursion:

$$F(a, b, c) = \min\{F(b + 1, b, c_1) + \sum_{j \in \text{Ass}_1} h_{j, \max \mathcal{A}_j \cap [c, c_1 - 1]}, F(a, b + 1, c) + \sum_{j \in \text{Ass}_2} h_{j, c_2}\} \ .$$

In the latter expression, the left-side terms of the min-expression compute the reduced cost associated with products $[c, c_1 - 1]$ when the rank value $b + 1$ is assigned to the leftmost product c_1. The right-side terms of the min-expression compute the reduced cost associated with product c_2 when the rank value $b + 1$ is assigned to the rightmost product $c_2 - 1$.

For the remaining cases, it can be shown that: (i) when $b > a$ and $b - a \geq 2$, we formulate a symmetrical recursion, (ii) for the boundary cases $b = a + 1$ or $a = b + 1$, there exists a single quasi-convex preference list that satisfies the constraints, thus the dynamic programming value can be computed immediately. □

Implementation details. For simplicity, we do not attempt to optimize the central permutation of the quasi-convex structure: as a heuristic, we choose the increasing price order (to avoid endogeneity issues, this average price is computed on an anterior dataset). To control for the risk of over-fitting, our stopping criterion picks the best final step out of $\{100, 200, 300, 400, 500, 600\}$ through cross-validation. We will also decide on the norm ℓ_1 or ℓ_2, used in the final calibration step, through cross-validation.

MNL and MMNL models. In order to estimate the MNL and MMNL parameters from data, we use standard maximum likelihood estimation (McFadden 1973, Talluri and Van Ryzin 2006). The corresponding maximum log-likelihood problem is implemented using the optimization software Ipopt (Wächter and Biegler 2006). This estimation method is a standard approach used to calibrate discrete mixtures of MNL (Bierlaire 2003, Hess et al. 2007). Contrary to the MNL model, the log-likelihood function associated with the MMNL model is non-concave and global optimization in this setting is not guaranteed.

3.8.2 Synthetic data

In this section, we evaluate the predictive power of the quasi-convex model on synthetic datasets, against MMNL models with up to 3 customer segments. Assortment optimization with a larger number of segments is effectively impractical as noted earlier (the computational complexity scales exponentially with the number of segments). Against such benchmarks, we demonstrate the predictive power of the quasi-convex model. We first explain how the synthetic datasets are generated, and then describe our numerical results. The experiments are conducted on random instances, with a fixed number of products $n = 50$. The dataset is constructed by randomly generating 100 assortments $\mathcal{A}_1, \dots, \mathcal{A}_{100}$, each formed by drawing n independent Bernoulli trials with probability of success 0.5. To generate the purchase probability data, we make use of the following ground truth models:

- *MMNL models:* For our first set of synthetic data, we generate random MMNL(5) instances. It is worth noting that assortment optimization against a 5 segment model is effectively impractical, and as such we will eventually fit an MMNL model with a smaller number of segments to this data. The preference weights $w_{i,j}$ are drawn independently from a log-normal distribution of scale σ, where σ is varied in the set $\{1, 10, 20, 40\}$. Each customer segment occurs with probability $\mu_1 = \cdots = \mu_5 = 1/5$. Here, σ intuitively controls the amount of heterogeneity in choice behavior across segments; we will momentarily see that predictive performance is sensitive to this parameter.

- *Consider-then-choose model:* For our second set of synthetic data, the purchase probabilities arise from the intervals model introduced in Section 3.4.1. Indexing the set of all possible intervals by $k \in \{1, 2, \dots, K\}$, the probability vector $(\lambda_1, \dots, \lambda_K)$ is drawn uniformly at random from the unit simplex.

Given the above generative settings, our synthetic datasets take the form of a random matrix $(p_{ij})_{i \in [n], j \in [100]}$, where the entry p_{ij} is the empirical probability of purchases of product i within the assortment $\mathcal{A}_j$ according to the ground truth choice model.

Prediction task and error metrics. For each dataset thus formed, we carried out a 5-fold cross-validation to estimate the prediction accuracy of the different models out-of-sample. We report two prediction error metrics: the mean square error (MSE), expressed in normalized form as a percentage of the total variance of the data and the mean absolute percentage error (MAP). Specifically, letting $\mathcal{OS} \subseteq [100]$ designate the collection of out-of-sample assortments and $(\hat{p}_{ij})_{i \in [n] j \in \mathcal{OS}}$ be the prediction matrix, we

Table 3.3: Prediction errors of the models estimated from the synthetic data, in different generative settings.

Ground truth		Quasi-convex		MNL		MMNL(2)		MMNL(3)	
		MSE	MAP	MSE	MAP	MSE	MAP	MSE	MAP
MMNL(5)	$\sigma = 1$	0.059	0.204	0.005	0.050	**0.002**	0.048	0.004	**0.048**
	$\sigma = 10$	**0.169**	**0.463**	0.207	0.529	0.204	0.510	0.197	0.511
	$\sigma = 20$	**0.197**	**0.489**	0.224	0.520	0.217	0.518	0.216	0.514
	$\sigma = 40$	**0.199**	**0.472**	0.245	0.533	0.240	0.524	0.226	0.519
Intervals		**0.003**	**0.010**	0.286	0.509	0.229	0.442	0.225	0.437

Recall that σ is the scale of the log-normal generator used to draw the preference weights in the MMNL instance. Each entry is obtained by sampling 10 random instances. After learning the quasi-convex lists through column generation, the final model is calibrated using the ℓ_1 or the ℓ_2-norm, based on cross-validation.

have:

$$\text{MSE} = \frac{\sum_{j\in\mathcal{OS}}\sum_{i\in\mathcal{A}_j}(\hat{p}_{ij}-p_{ij})^2}{\sum_{j\in\mathcal{OS}}\sum_{i\in\mathcal{A}_j}p_{ij}^2} \ ,$$

and

$$\text{MAP} = \sum_{j\in\mathcal{OS}}\frac{1}{|\mathcal{OS}|\cdot|\mathcal{A}_j|}\cdot\sum_{i\in\mathcal{A}_j}\frac{|\hat{p}_{ij}-p_{ij}|}{0.01+p_{ij}} \ .$$

Results. The numerical results, show that the quasi-convex family has relatively accurate predictions in *all* generative settings, and outperforms the parametric models in the plurality of cases. As one might expect, when the intervals model is posited as ground truth, the estimated quasi-convex instances provide low prediction errors out-of-sample. On the other hand, the out-of-sample errors incurred under any of the MMNL models is substantially larger.

More interestingly, when data is generated according to an MMNL model, we see that as the scale parameter that controls customer heterogeneity across segments, σ, grows (i.e., $\sigma \geq 10$), the quasi-convex model provides more accurate out of sample predictions than the estimated MNL, MMNL(2) and MMNL(3) instances. For example, for $\sigma = 10$, the MSE is smaller by a factor of between 14 and 22%, while the MAP is smaller by a factor of 10% to 12%. The cases where the MMNL models provide an improvement in prediction error are when $\sigma \leq 5$. In such cases the heterogeneity in choice across distinct segments is so small that an MNL model would be expected to provide a good fit to the data – a fact that is borne out in the experiment. In this setting the *absolute* prediction errors are relatively small across all models, including the quasi-convex model.

3.8.3 Purchase panel data

Our industry partner tracks the daily transactions made by hundreds of thousands of consumers across several product categories and retailers. In order to form our input datasets, we had access to three product categories with frequent purchases, namely *Bath tissue*, *Shampoo and Conditioners*, and *Dog Food and Treats*. In each category, the time horizon considered varies from 2 to 5 months to obtain around 1 million transactions. Transactions are aggregated at the brand level. Each assortment corresponds to the combination of a retail chain and a US state: it is defined as the collection of products with at least one transaction. We make the assumption that, over the time periods considered, retailers in the dataset carry near identical assortments at the brand level – this assumption is effectively verified over other time windows. For any given state and product category, we only consider those retailers with greater than 500 transactions in that state for that category. Having specified the assortments $\mathcal{A}_1, \ldots, \mathcal{A}_s$, the purchase probabilities p_{ij} are obtained by computing the market shares of products according to the observed transactions. Note that, since we do not observe the no-purchases (when customers leave the store without making any purchase), our choice probability distribution is defined over the assortment of products.

Hence, the *Bath tissue*, *Dog Food and Treats*, and *Shampoo and Conditioners* datasets are respectively formed by 58, 211 and 257 assortments, with $\sim$ 400K to 600K transactions after we perform the data filtering steps. To verify the robustness of our conclusions, multiple experiments were conducted by randomly sub-sampling 50 products in each category.

It is worth mentioning that our simplifying assumptions do not alter the estimation of the MNL choice model – the probability distribution obtained from the data after

Table 3.4: % Improvements in the predictive accuracy of our quasi-convex model against the chosen benchmarks.

(% Improvement)		MNL		MMNL(2)		MMNL(3)	
Datasets		MSE	MAP	MSE	MAP	MSE	MAP
Dog Food & Dog Treats	$k = 2$	12%	28%	7.5%	14%	6.8%	14%
	$k = 5$	21%	23%	13%	15%	15%	12%
Bath tissue	$k = 2$	79%	59%	71%	50%	65%	42%
	$k = 5$	67%	35%	58%	31%	51%	26%
Shampoo & Conditioners	$k = 2$	15%	22%	9.3%	13%	11%	13%
	$k = 5$	12%	18%	8.2%	14%	4.0%	10%

Note: k is the cross-validation parameter. Each entry is computed by averaging over 10 cross-validation estimates.

truncating an assortment is still compatible with MNL model – while it might very well impair the predictive power of the quasi-convex distribution.

Results. Table 3.4 reports the percentage improvement in predictive accuracy for each metric and product category. For example, denoting by MSE_Q the chi-square errors on the predictions of the quasi-convex model and MSE_{MNL} those associated with the MNL model, the percentage improvement relative to the MNL model is given by

$$\frac{\text{MSE}_{MNL} - \text{MSE}_Q}{\text{MSE}_{MNL}} \,.$$

The table reports these quantities averaged across multiple experiments (each experiment obtained via a random sampling of products), and each cross-validation fold. Our quasi-convex modeling approach outperforms the parametric benchmark models by providing smaller out-of-sample prediction errors in *all* cases. The gain is smallest (as expected) for the MMNL(3) model, but remains substantial in absolute terms – as large as 60% for certain product categories. Our results are robust to using other metrics (not reported here) such as chi-square. In summary, we conclude that the quasi-convex consider-then-choose model provides strong predictive power on real-world choice data.

3.9 Concluding Remarks

General objective function. A close examination of the algorithm reveals that our results apply in fact to a more general class of objective functions that we describe below. We introduce the *pay-off* function $f : [n] \times [K] \to \mathbb{R}$, where $f(i, j)$ is the contribution to the objective due to the purchase of product i by the preference list j. Letting $i(\mathcal{A}, j)$ denote the product purchased by preference list j when faced with assortment $\mathcal{A}$, the objective value for the assortment $\mathcal{A}$ is given by the expected pay-off: $\sum_{j \in [K]} \lambda_j \cdot f(i(\mathcal{A}, j))$. We may mention two application of practical interest captured by this more general family of objective functions. It accounts for potentially heterogeneous per-selling price over customer-types, e.g., targeted promotions or loyalty programs. Another interesting problem formulation consists in maximizing the customers' utility. In this case $f(i, j)$ is interpreted as the utility garnered by customer $j \in [k]$ when purchasing product $i \in [n]$.

Future work. This work opens exciting perspectives for future research. A natural lead is to further investigate the interplay between the behavioral heuristics identified by the marketing literature and the running time complexity of our dynamic program. In addition, the implementation allows for several refinements, such as using heuristics to prune the computational tree or exploring the subproblems in parallel. Another important question is to investigate the identifiability of the models discussed in this work from data, in particular the quasi-convex model, and study the computational and sample complexity associated with the estimation problem.

Chapter 4

Joint Assortment and Inventory Decisions: Nonparametric Models

4.1 Introduction

Matching supply with uncertain demand is a key driver of strategic and operational success in many industries (Fisher 2011). The supply side hinges on choosing an assortment, that specifies the subset of products offered to customers, as well as deciding on their inventory levels, i.e., the initial number of units stocked of each product. Typically, inventory and assortment decisions are very likely to inform each other. Indeed, as demand evolves during the selling season, some products may be fully consumed (or stock-out) due to inventory limitations. Consequently, customers arriving at different times could very well be facing different assortments, thereby affecting their purchasing behavior. Despite this fundamental interplay between inventory and assortment, the corresponding decision-making problems have been studied for the most part in separate frameworks, arguably due to computational intractability.

In this chapter, we consider the joint assortment optimization and inventory management problem, whereby a retailer wishes to maximize expected revenue in the face of stochastic demand, consisting of a random number of customers with dynamic substitution across products. A complete mathematical description of this model, referred to as *dynamic assortment planning*, is given in Section 4.1.2. Somewhat informally, the retailer selects an assortment of products, and determines their initial inventory levels, given a capacity constraint on the total number of units to be stocked. These decisions are made at the beginning of the selling period (similar to the newsvendor problem). Next, the consumption process consists in a random sequence of arriving customers, each of which purchases at most one unit. The problem formulation depends on the distribution of the number of arriving customers, named the *demand* hereafter, and on the probabilistic structure of their purchasing preferences. This choice model describes the probability that a unit of any given product is purchased, amongst the assortment products available upon the customer's arrival.

In recent years, the availability of highly detailed purchase data has motivated a nonparametric approach to modeling the customer choice preferences through a distribution over ranked preference lists Rusmevichientong et al. (2006), Farias et al. (2013). Generally speaking, each list describes a ranking over the product alternatives considered by a customer, among which she chooses her most preferred product out of those available upon arrival, or leaves without making any purchase if none is currently stocked. Unfortunately, as choice models become more detailed, and subsume more general families of preference list distributions, the corresponding optimization problems rapidly become intractable. In this chapter, we develop general-purpose algorithms under distributions over preference lists. Our approach enjoys best-possible theoretical guarantees and demonstrates superior practical performance against existing heuristics. Moreover, we show that additional structure on customer choices, in the vein of the behavioral assumptions discussed in Chapter 3, can be leveraged to obtain improved guarantees.

Directly-related work. With a single customer arrival, the problem is referred as the *static* formulation, which corresponds to the standard assortment optimization models studied in the previous chapters. Even in this seemingly simple context, the problem is inapproximable within any factor sub-linear in the number of products as shown in Chapter2. On the positive side, there has been an ever-growing line of work in recent revenue management literature on this static formulation, investigating tractable approaches for special cases of the aforementioned choice model Ryzin and Mahajan (1999), Talluri and van Ryzin (2004), Honhon et al. (2012), Blanchet et al. (2016), Davis et al. (2014), Li et al. (2015). In contrast, under multiple customer arrivals, the problem becomes considerably harder due to the additional *dynamic* aspect. Indeed, the initial assortment is altered along the sequence of arrivals due to stock-out events, and the dynamic substitution behavior of customers depends on each sample path realization. In fact, even the evaluation of the expected revenue for a given offer set is by itself a challenging computational question.

For the dynamic models in question, most of the work we are aware of makes use of heuristics based on continuous relaxations and probabilistic assumptions Smith and Agrawal (2000), Mahajan and van Ryzin (2001), Kök and Fisher (2007), Nagarajan and Rajagopalan (2008), Honhon et al. (2010), Topaloglu (2013). In particular, these approaches either give rise to exponential-time algorithms, do not admit provable non-trivial approximation guarantees, or apply to inventory models of very different nature. Interestingly, even in rather restricted settings, such as that of horizontally differentiated products (where prices are uniform), not much is known at present time. In this context, Gaur and Honhon Gaur and Honhon (2006) studied a newsvendor-like inventory model with dynamic substitution, and devised a heuristic for the locational choice model, assuming a parametric demand distribution. While they propose lower and upper bounds on the optimal revenue, the analysis thereof does not translate into an efficient algorithm with theoretical guarantees. Chen and Bassok Chen and Bassok (2008) considered a similar setting where prices are uniform, assuming a less-realistic allocation rule, where products are assigned to customers at the retail's discretion after seeing the entire sequence of arriving customers, instead of sequentially.

To our knowledge, prior to our work, the paper of Goyal et al. Goyal et al. (2016) and Segev Segev (2015) are the only papers that study dynamic optimization models with discrete demand realizations and a fully stochastic sequence of arrivals through the lens of approximation algorithms. Specifically, Goyal et al. devised a polynomial-time approximation scheme (PTAS) assuming that the demand follows an increasing failure rate (IFR) distribution, when the choice model consists in a distribution over nested preference lists, referred to in the sequel as the nested choice model. This model assumes that customers always prefer cheaper products, and have a budget constraint that dictates the most expensive product they are willing to purchase. Their algorithm is based on efficient enumeration methods, by observing that there are near-optimal assortments comprised of a constant number of products.

While this algorithm approximates the optimal revenue within factor $1 - \epsilon$, the overall approach suffers from two major drawbacks. First, since it resorts to enumerating all inventory levels over a predetermined assortment with poly$(1/\epsilon)$ products, the resulting running time is exponential in $1/\epsilon$, and becomes impractical even for medium-scale instances. Second, in some practical settings the demand is 'heavy tailed', while retailers wish to hedge against extreme demand realizations, especially for newly launched products with limited data and forecast accuracy. However, the algorithm and its analysis do not carry over when the IFR property is relaxed. The latter drawback has been bypassed by Segev Segev (2015), who proposed a quasi-PTAS for general demand distributions (again, under the nested choice model), based on an dynamic programming approach with an approximate state space representation. However, this result is more theoretical in nature, and still leaves open the question of efficiently approximating nested preference lists under general demand distributions.

4.1.1 Our results

Our main contribution is to provide the first polynomial-time algorithms with provable approximation guarantees for a broad class of demand and choice model specifications in dynamic assortment planning. From a technical perspective, we introduce a number of novel algorithmic ideas of independent interest, that could very well be applicable in a wide range of settings, and unravel hidden relations to submodular maximization. In addition, our algorithms employ a mixture of greedy procedures and low-dimensional dynamic programs, that are suitable for solving instances of practical nature and scale. In computational experiments, these algorithms are shown to be faster than existing heuristics by an order of magnitude and to result in substantially better expected revenues. Our main results can be briefly summarized as follows.

General choice model. We first devise an approximation algorithm in the context of horizontal differentiation, where product prices are identical, assuming an IFR demand distribution. As previously mentioned, a similar modeling approach has been taken by Gaur and Honhon Gaur and Honhon (2006) and Chen and Bassok Chen and Bassok (2008). In this setting, we obtain a constant-factor approximation, without any structural assumptions on the distribution over ranked lists. Our algorithm, formally described in Section 4.2, is based on a two-step "selective-greedy" approach. In the selection step, we restrict attention to a subset of products, identified by approximately solving the underlying *static* problem, i.e., assuming a single customer arrival, or equivalently, without inventory limitations. Next, the inventory levels are determined by greedily optimizing a multi-item newsvendor problem, whose optimal solution provides a lower bound on the optimal expected revenue. To analyze this approach, we explicitly construct a feasible solution to the latter problem that generates an $(e-1)/(4e-2)$ fraction of the optimal expected revenue.

This result extends to the general class of random utility choice models (preference list distributions with potentially exponential support), as long as there exists an efficient oracle to evaluate the purchase probabilities of products in any given assortment. As explained later on, this setting subsumes most choice models studied in the revenue management literature. Moreover, when the static version of the problem admits an approximation ratio of $\alpha \geq 1 - 1/e$, we argue that the above-mentioned guarantee can be improved to $\alpha/4$.

In the presence of price differentiation, the selection step is complemented by price thresholding. That is, we eliminate all products cheaper than an appropriate price threshold, and apply our horizontal differentiation algorithm on the remaining products, assuming identical prices. This approach yields an approximation guarantee of $O(\log(P_{\max}/P_{\min}))$, where $P_{\max}$ and $P_{\min}$ stand for the maximum and minimum price of any product, respectively. The latter ratio is best possible in this setting, as shown in Chapter 2 (to wit, the static problem is a special case of the dynamic problem with IFR demand distributions).

Intervals choice model. We proceed by proving that the above-mentioned logarithmic ratio, obtained under price differentiation, can be beaten in more structured settings. We first investigate a generalization of nested preference lists, known as the intervals choice model. This model subsumes any distribution over lists that are comprised of an interval of products, ranked by increasing price order. Such preference lists find behavioral justification in capturing common screening rules, used by customers to generate their choice set (see the survey by Hauser Hauser et al. (2009) in the marketing literature). Indeed, the intervals structure naturally arises when the customers' choice set is formed by the conjunction of a budget constraint and a quality requirement, assuming that price and perceived quality are inversely related, which is commonly observed in practical settings Zeithalm (1988).

Assuming an IFR distribution of customer arrivals, we develop in Section 4.3 an algorithmic approach with an approximation guarantee of $O(\log\log(P_{\max}/P_{\min}))$ under the intervals choice model. First, we show that the problem can be approximated within a factor logarithmic in the number of products. This is achieved through a recursive decomposition of the preference lists into a logarithmic number of classes, thereby creating independent and highly-structured instances, that can be approximated within constant factors. The log-logarithmic ratio is attained via a refined decomposition, where products are initially grouped into nearly-uniform price buckets, which allow us to employ our algorithm for horizontally differentiated products as a subroutine.

Nested choice model with general demand distributions. For arbitrary (non-IFR) demand distributions, we provide the first polynomial-time approximation algorithm under the nested choice model, attaining a performance guarantee of $1-1/e$. Our algorithm, whose specifics are given in Section 4.4, reveals a hidden submodular structure within this setting, and relies on a distinct selective-greedy approach (very dissimilar to the one described in Section 4.2). We initially eliminate sub-optimal products by leveraging solutions to multiple instances of the static version. Next, the resulting problem formed by the residual products is cast as a capacity-constrained maximization of a certain submodular function. In contrast to existing enumeration-based approaches Goyal et al. (2016), Segev (2015), our algorithm is very efficient in practice.

Computational experiments. In Section 4.5, we conduct extensive computational experiments on randomly-generated instances, showing that our general-purpose algorithm largely outperforms existing heuristics, in terms of both revenue and efficiency. Specifically, our approach is compared to the following heuristics:

1. A local-search heuristic based on greedily exchanging units between pairs of products.

2. A gradient-descent approach based on a continuous extension of the revenue function.

3. A discrete-greedy algorithm, where in each step a single unit is added to the product with the largest marginal expected revenue.

4. The approximation scheme of Goyal et al. (2016) for the nested choice model.

On average, the expected revenue is (on average) increased by a factor ranging between 8% and 30% relative to the best heuristic, while reducing the running time in most configurations. In structured settings, with interval and nested preference lists, our comparative experiments again validate the practicality of the algorithms we propose against existing heuristics.

Counter-examples. For each of the settings considered in Sections 4.2-4.4, we construct carefully-made instances, showing that the objective function of the respective formulations is neither concave, nor submodular. These counter-examples, described in Appendix B.1.3, suggest that generic optimization methods do not directly apply to the problems considered in this chapter.

4.1.2 Problem formulation

Products and inventories. We are given a collection of n products, with per-unit selling prices $P_1 \leq \cdots \leq P_n$. In addition, there is a capacity bound of C on the total number of units to be stocked. In the (single-period) dynamic assortment problem, the retailer has to jointly decide on an assortment, i.e., a subset products to be stocked, as well as on the initial inventory levels of these products, which are not replenished later on. In other words, a feasible solution specifies the initial inventory levels of all products, represented by an integer-valued vector $U = (u_1, \ldots, u_n)$ that meets the capacity constraint, $\sum_{i=1}^{n} u_i \leq C$.

The consumption process. Independently of stocking decisions, a random number of customers M arrive sequentially, where the distribution of M is assumed to be known to the decision-maker. Each customer j picks a random preference list L_j that describes a sub-collection of products in decreasing order of preference. This list is drawn from a common known distribution over a collection of preference lists, independently of the other lists and the number of customers M. Unless mentioned otherwise, this distribution is encoded explicitly as a collection of preference lists $\mathcal{L}$, that are specified as an input along with their respective probabilities. Upon arrival, each customer purchases a single unit of the most preferred product on her list available at that time. In other words, the customer first attempts to purchase her most preferred product, and if that product has stocked out or was not initially selected in the assortment, the customer substitutes to the second most preferred product, so forth and so on. If none of the products in her preference list is available, this customer leaves without purchasing any product. Therefore, at each step, the inventory vector is decremented by at most one unit, corresponding to the customer's purchasing decision.

Objective. When the sequence of customer arrivals ends, we use $\mathcal{R}(U)$ to denote the revenue resulting from an initial inventory vector U. Based on the preceding discussion, this revenue is clearly random, due to the stochasticity in the number of customers and in their choice of preference lists. The objective is to compute a feasible inventory vector, so that the expected revenue is maximized, i.e.,

$$\max_{(u_1,\ldots,u_n)\in\mathbb{Z}_+^n} \left\{ \mathbb{E}\left[\mathcal{R}(u_1,\ldots,u_n)\right] : \sum_{i=1}^{n} u_i \leq C \right\} .$$

Structural properties. We have previously discussed several structural properties that give rise to different settings studied in this chapter. Below, detailed definitions of these modeling assumptions are provided.

- *Nested choice model:* This model describes distributions over a collection of preference lists $\mathcal{L}$ that consists of intervals of the form $(1,\ldots,\ell)$, where $\ell \in [n]$. Namely, there are $n+1$ possible preferences lists, $(), (1), (1,2), \ldots, (1,2,\ldots,n)$, where the respective probabilities of these lists are arbitrary. Here, $()$ denotes the empty preference list, for customers who are not interested in purchasing any product.

- *Intervals choice model:* This model describes a more general class of preference list distributions, where $\mathcal{L}$ consists of intervals of the form $(\ell,\ldots,k)$, with $1 \leq \ell \leq k \leq n$. Again, the respective probabilities of these lists can be arbitrary.

- *Increasing Failure Rate:* Here, the distribution of the number of customers M is assumed to have an increasing failure rate (IFR), meaning that $\Pr\left[M = k\right]/\Pr\left[M \geq k\right]$ is non-decreasing over the integer domain. This definition is equivalent to requiring that the sequence of random variables $[M - k | M \geq k]_{k\in\mathbb{Z}}$ is stochastically non-increasing in k. For definitions of stochastic orders and stochastic monotonicity, we refer the reader to Shaked and Shanthikumar Shaked and Shanthikumar (1994). It is worth mentioning that the IFR property is satisfied by many distributions considered in operations management applications, including Normal, Exponential, Geometric, Poisson, and Beta (for certain parameters).

Remark 1: Static formulation. The static case corresponds to the situation where there is a single customer arrival. This setting is equivalent to relaxing the capacity constraint (i.e., $C = \infty$), and the problem reduces to the standard assortment optimization formulation. Indeed, since there are no stock-out events, we can always offer the optimal assortment to each arriving customer. The assortment computed in this setting is referred to as the *optimal static assortment*.

Remark 2: Notation. In contrast with the previous chapter, because we consider simpler choice structures, our notation does not disambiguate the consideration sets from the ranking permutation. That is, in certain settings, we will treat the ranked preference lists in $\mathcal{L}$ as subsets of products. In addition, for each list $L_\ell \in \mathcal{L}$, we use λ_ℓ to denote the probability that it is picked by an arriving customer. We allow $\mathcal{L}$ to interchangeably designate the set of preference lists, as well as the corresponding collection of customer-type indices.

4.2 General choice model

In what follows, we consider the most general setting, where the underlying choice model is expressed as a distribution over ranked preference lists, potentially with exponentially-large support. As discussed subsequently, this setting coincides with the class of random-utility choice models, and therefore subsumes most models of practical interest proposed in the literature.

Our approximation algorithm is introduced in an incremental way. In Sections 4.2.1-4.2.3, we begin by investigating the setting of horizontally differentiated products. Here, products are assumed to be associated with uniform prices, meaning that without loss of generality $P_1 = \cdots = P_n = 1$, and the retailer wishes to maximize his expected sales quantity. When the number of customers M follows an IFR distribution, we show how to efficiently compute an inventory vector that approximates the optimal expected revenue within a constant factor. For ease of presentation, we do not attempt to optimize the latter constant. It is worth noting that the uniform-price problem is NP-hard to approximate within factor larger than $1 - 1/e$, since it subsumes the maximum coverage problem Feige (1998) as a special case, even with a single customer.

Theorem 4.2.1. *When M is IFR-distributed and the product prices are uniform, the dynamic assortment planning problem under general preference list distributions can be approximated within factor $(e-1)/(4e-2)$ in polynomial time.*

In Section 4.2.4, this constant-factor approximation is leveraged as a subroutine to solve the general problem with price differentiation. Here, without loss of generality, we assume that products are indexed such that $P_1 \leq \cdots \leq P_n$. As stated in the next theorem, our performance guarantee scales logarithmically with the ratio of extremal prices. The latter ratio is best possible in this setting (up to constant terms), and matches the inapproximability bound established in Chapter 2.

Theorem 4.2.2. *When M is IFR-distributed, the dynamic assortment planning problem under general preference list distributions can be approximated within factor $O(\log(P_n/P_1))$ in polynomial time.*

4.2.1 Algorithm under horizontal differentiation

At a high level, our algorithm is based on two-step approach, termed selective-greedy:

- *Selection step.* First, we ignore the inventory limitations and select an assortment of products that approximates the static (single-customer) problem. To this end, we observe that the static problem is equivalent to computing a weighted maximum coverage of a set system defined by the collection of preference lists. Thus, a constant-factor approximation for the static case is obtained by a greedy allocation rule.

- *Greedy step.* Next, to allocate the inventory capacity over the assortment products in the dynamic setting, we consider a lower bound on the expected revenue, that can be viewed a multi-item newsvendor formulation. The latter (simplified) objective is then optimized greedily.

In what follows, we provide a more detailed description of the algorithm.

Step 1 (selection): approximating the static solution. Recall that $\mathcal{L}$ designates the original collection of preference lists, where each list L_ℓ is picked by a single customer with probability λ_ℓ. We begin by considering the static variant of the problem, that seeks to maximize the expected revenue extracted from a single arriving customer. Since all prices are identical, out of all subsets of products with cardinality at most C, we wish to pick one that maximizes the total probability of all preference lists in $\mathcal{L}$ that are being hit. In other words, we would like to identify a subset of products that satisfies the cardinality constraint and achieves a maximum coverage of the preference lists, where each product covers (or hits) the subset of lists that contain it. This is precisely an instance of the maximum coverage problem, that can be approximated within factor $1 - 1/e$ by a classic greedy procedure Nemhauser et al. (1978). Specifically, a single product is greedily added at each step to maximize the coverage quantity, defined as the combined probability of all lists intersecting the assortment. As a result, we use $\mathcal{Q} \subseteq [n]$ to denote the corresponding assortment picked for the static problem. In addition, we assume without loss of generality that the assortment $\mathcal{Q}$ is minimal with respect to inclusion, that is, removing any product would decrease the combined probability of preference lists that contain any of the products in $\mathcal{Q}$.

Newsvendor-like lower bound. We begin by defining $\mathcal{L}_\mathcal{Q} \subseteq \mathcal{L}$ as the subset of lists that intersect with the assortment $\mathcal{Q}$. We construct an assignment $\mathcal{A} : \mathcal{L}_\mathcal{Q} \to \mathcal{Q}$ that maps each list in $\mathcal{L}_\mathcal{Q}$ to its most preferred product in $\mathcal{Q}$, which exists by definition of $\mathcal{L}_\mathcal{Q}$. Hence, $\mathcal{A}^{-1}(i)$ is the subset of lists in which product $i \in \mathcal{Q}$ is the most preferred when faced with the assortment $\mathcal{Q}$. Note that, since $\mathcal{Q}$ is minimal with respect to inclusion (see step 1), each product in this assortment is necessarily assigned with at least one preference list in $\mathcal{L}_\mathcal{Q}$, meaning that $\mathcal{A}^{-1}(i) \neq \emptyset$ for every $i \in \mathcal{Q}$. We highlight a basic property attained by the assignment $\mathcal{A}$. Consider some product $i \in \mathcal{Q}$, and suppose that we are looking on a customer who has just arrived. The key observation is that, if product i has at least one unit in stock at the moment, the current customer will purchase i with probability at least $\psi_i = \sum_{\ell \in \mathcal{A}^{-1}(i)} \lambda_\ell$, regardless of the inventory levels of all other products. The reason is that, for any list $\ell \in \mathcal{A}^{-1}(i)$, which occurs with probability λ_ℓ, product i is preferred over any other product in the assortment $\mathcal{Q}$, and we are not stocking any of the products in $[n] \setminus \mathcal{Q}$. Therefore, the number of units purchased from i if this product had an infinite (unlimited) inventory level is stochastically larger than $Y_i \sim B(M, \psi_i)$. However, assuming that u_i units of product i are initially stocked, we would actually be considering the truncated random variable $\bar{Y}_i(u_i) = \min\{Y_i, u_i\}$. Therefore, letting $(u_1, \dots, u_n)$ be an inventory vector stocking only products in $\mathcal{Q}$, an immediate lower bound on its expected revenue is given by:

$$\mathbb{E}\left[\mathcal{R}(u_1, \dots, u_n)\right] \geq \sum_{i \in \mathcal{Q}} \mathbb{E}\left[\bar{Y}_i(u_i)\right] \ . \tag{4.1}$$

Since the above lower bound is separable by products, it can be viewed as the revenue function of a multi-item newsvendor problem (without stock-out substitution), over the products in $\mathcal{Q}$.

Step 2 (greedy): solving a multi-item newsvendor problem. The inventory levels of products in $\mathcal{Q}$ are now set in order to optimize the lower bound described by inequality (4.1). As noted above, this formulation is a special case of the multi-item newsvendor problem subject to a single cardinality constraint, with salvage value and cost 0. It is well-known that an exact solution can be derived in polynomial time (see, e.g., (Muckstadt and Sapra 2010), Chapter 5). For instance, one may employ a greedy procedure that, at each step, augments the current inventory vector by a single unit that incurs the largest marginal increase of the expected revenue. It is worth noting that our lower bound on the revenue contribution of any given unit can easily be computed in polynomial time. In contrast, evaluating the exact expected revenue it generates is an open question by itself, as explained in Section 4.1, and our algorithm is surprisingly able to bypass this difficulty.

Running time. The selection step requires n greedy increments, each with at most n evaluations of the static expected revenue, whereas the greedy step involves C increments of the inventory vector, each leading to n evaluations of the multi-item newsvendor objective function. The static expected revenue of a given assortment can be computed in time $O(n \cdot |\mathcal{L}|)$ while the newsvendor objective can be evaluated through dynamic programming in time $O(nC\bar{M})$, where $\bar{M}$ is the maximal number of arrivals (see Goyal et al. Goyal et al. (2016)). Hence, the overall running time is $O(n^3 \cdot |\mathcal{L}| + n^2 C^2 \bar{M})$. In fact, by observing that each incremental action only affects a constant number of the revenue terms (and their corresponding purchase probabilities), a refined implementation of the evaluation oracles leads to a running time of $O(n^2 \cdot |\mathcal{L}| + nC\bar{M})$. Practically speaking, for common parametric random variables (e.g., gaussian) where $\bar{M}$ is infinite, the newsvendor objective can be evaluated using numerical integration techniques, with high accuracy.

4.2.2 Analysis under horizontal differentiation

To analyze our algorithm, we explicitly construct a 'good' candidate solution, making use of products in $\mathcal{Q}$. The subsequent analysis reveals that, under the choice of such inventory levels, the newsvendor-like lower bound approximates the optimal revenue within a constant factor. This candidate solution is constructed in two steps.

Step 1: Rounding up the capacity. We begin by modifying the original capacity C. Specifically, we round the capacity value up to the nearest multiple of $2 \cdot |\mathcal{Q}|$, denoted by $\bar{C}$. Note that since $|\mathcal{Q}| \leq C$, we must have $\bar{C} \leq 2C$. Our approach to design a feasible solution first creates a solution under the relaxed capacity $\bar{C}$. We overload notation by reusing C as the current capacity, instead of $\bar{C}$, in the next algorithmic steps. However, once the final inventory vector is computed, it remains to restore the original capacity by selecting the 'best' C units. Namely, we select the C units with largest contribution to the lower bound described in Section 4.2.1. This leaves us with at least half of the lower bound attained by the relaxed solution.

Step 2: Setting inventory levels. Based on the assignment $\mathcal{A}$ defined earlier, we proceed by explaining how to spread the capacity of C over the underlying set of products $\mathcal{Q}$. Intuitively, we would like the number of units stocked from each product $i \in \mathcal{Q}$ to be proportional to $\psi_i / \Lambda_{\mathcal{Q}}$, where $\Lambda_{\mathcal{Q}} = \sum_{i \in \mathcal{Q}} \psi_i$. Namely,

$$\tilde{u}_i = \frac{\psi_i}{\Lambda_{\mathcal{Q}}} \cdot C \ .$$

However, this quantity may not be integral, and is therefore rounded down to the nearest multiple of $C/(2 \cdot |\mathcal{Q}|)$, which is necessarily integral by step 1. For this purpose, we can uniquely write

$$\tilde{u}_i = \mu_i \cdot \frac{C}{2 \cdot |\mathcal{Q}|} + \alpha_i \ ,$$

for some integer $\mu_i \in [0, 2 \cdot |\mathcal{Q}|]$ and some real $\alpha_i \in [0, C/(2 \cdot |\mathcal{Q}|))$. With these definitions in place, for each product $i \in \mathcal{Q}$, the number of units to be stocked is

$$u_i = \mu_i \cdot \frac{C}{2 \cdot |\mathcal{Q}|} \ ,$$

while other products are not stocked at all. This way, each product indeed has an integer number of units stocked, and furthermore, we do not exceed the overall capacity, since

$$\sum_{i \in \mathcal{Q}} u_i \leq \sum_{i \in \mathcal{Q}} \tilde{u}_i = \frac{C}{\Lambda_{\mathcal{Q}}} \cdot \sum_{i \in \mathcal{Q}} \psi_i = C \ .$$

Deriving the approximation ratio. For the remainder of this section, let $(u_1, \ldots, u_n)$ be the inventory vector that has just been constructed. Since this vector stocks only products in $\mathcal{Q}$, it is a feasible solution to the multi-item newsvendor instance solved (exactly) by our algorithm. Therefore, to prove Theorem 4.2.1, it remains to show that the expected revenue generated by $(u_1, \ldots, u_n)$ can be lower bounded in terms of the optimal expected revenue.

We begin by stating two technical lemmas, that prove useful for analyzing the consumption process of the inventory vector $(u_1, \ldots, u_n)$. Although written in slightly different terms, the first lemma has been proven by Goyal et al. (Goyal et al. 2016, Lem. 4). The second lemma is easy to establish, as shown in Appendix B.1.2.

Lemma 4.2.3. *Let M be a non-negative integer-valued IFR random variable. For any $\alpha \in [0,1]$, the random variable $X \sim B(M,\alpha)$ also follows an IFR distribution.*

Lemma 4.2.4. *Let X be a non-negative IFR random variable, and let $\bar{X} = \min\{X, C\}$, for some constant C. Suppose that $\mathbb{E}[\bar{X}] \leq \delta C$ for some $\delta \in [0,1]$. Then, $\mathbb{E}[\bar{X}] \geq (1-\delta) \cdot \mathbb{E}[X]$.*

Upper bounds on the optimal revenue. The important observation is that, for any inventory vector with a total capacity of at most C, an arriving customer will purchase a unit with probability at most $(e/(e-1)) \cdot \Lambda_{\mathcal{Q}}$. This follows by noting that, as explained in step 1, the assortment $\mathcal{Q}$ approximates the optimal maximal coverage solution within factor $1-1/e$. Therefore, the expected revenue of the optimal inventory vector $(u_1^*, \dots, u_n^*)$ can be bounded by

$$\mathbb{E}\left[\mathcal{R}(u_1^*, \dots, u_n^*)\right] \leq \min\left\{C, \frac{e}{e-1} \cdot \mathbb{E}\left[M\right] \cdot \Lambda_{\mathcal{Q}}\right\} . \tag{4.2}$$

Frequent and rare products. The key idea of our analysis is to distinguish between two types of products. For a parameter $\delta \in [0,1]$ whose value will be optimized later, we say that product $i \in \mathcal{Q}$ is frequent when, in expectation, at least a δ-fraction of the units stocked are purchased in the consumption process, i.e., $\mathbb{E}[\bar{Y}_i(u_i)] \geq \delta u_i$. Otherwise, this product is said to be rare. We denote the sets of frequent and rare products by $\mathcal{F}$ and $\mathcal{R}$, respectively. Note that, by the relation between u_i and $\tilde{u}_i$,

$$\sum_{i \in \mathcal{F}} u_i + \sum_{i \in \mathcal{R}} u_i = \sum_{i \in \mathcal{Q}} u_i = \sum_{i \in \mathcal{Q}} \tilde{u}_i - \sum_{i \in \mathcal{Q}} \alpha_i \geq C - |\mathcal{Q}| \cdot \frac{C}{2 \cdot |\mathcal{Q}|} = \frac{C}{2} . \tag{4.3}$$

We separately examine the contribution of each product type to the lower bound stated in inequality (4.1). For the contribution of frequent products, by definition we clearly have

$$\sum_{i \in \mathcal{F}} \mathbb{E}\left[\bar{Y}_i(u_i)\right] \geq \delta \cdot \sum_{i \in \mathcal{F}} u_i . \tag{4.4}$$

We now lower bound the contribution of rare products. Based on Lemma 4.2.3, since the number of customers M is assumed to be IFR distributed, we know that $Y_i \sim B(M, \psi_i)$ follows an IFR distribution as well. As a result, by Lemma 4.2.4, we infer that the expectations of Y_i and $\bar{Y}_i(u_i)$ are closely-related for every rare product i, meaning that $\mathbb{E}[\bar{Y}_i(u_i)] \geq (1-\delta) \cdot \mathbb{E}[Y_i]$, and therefore,

$$\sum_{i \in \mathcal{R}} \mathbb{E}\left[\bar{Y}_i(u_i)\right] \geq (1-\delta) \cdot \sum_{i \in \mathcal{R}} \mathbb{E}\left[Y_i\right] \ . \tag{4.5}$$

Also, by definition of u_i and $\tilde{u}_i$, we observe that

$$\mathbb{E}\left[Y_i\right] = \psi_i \cdot \mathbb{E}\left[M\right] = \frac{\Lambda_{\mathcal{Q}}}{C} \cdot \tilde{u}_i \cdot \mathbb{E}\left[M\right] \geq \frac{\Lambda_{\mathcal{Q}}}{C} \cdot u_i \cdot \mathbb{E}\left[M\right] \ .$$

Therefore, combining this with inequality (4.5), we obtain

$$\sum_{i \in \mathcal{R}} \mathbb{E}\left[\bar{Y}_i(u_i)\right] \geq (1-\delta) \cdot \mathbb{E}\left[M\right] \cdot \frac{\Lambda_{\mathcal{Q}}}{C} \cdot \sum_{i \in \mathcal{R}} u_i \ . \tag{4.6}$$

Conclusion. By substituting (4.4) and (4.6) into the lower bound stated in inequality (4.1) and setting $\delta = (e-1)/(2e-1)$, we infer that

$$\begin{aligned}
\mathbb{E}\left[\mathcal{R}(u_1, \ldots, u_n)\right] &\geq \delta \cdot \sum_{i \in \mathcal{F}} u_i + (1-\delta) \cdot \mathbb{E}\left[M\right] \cdot \frac{\Lambda_{\mathcal{Q}}}{C} \cdot \sum_{i \in \mathcal{R}} u_i \\
&\geq \mathbb{E}\left[\mathcal{R}(u_1^*, \ldots, u_n^*)\right] \cdot \left(\frac{\delta}{C} \cdot \sum_{i \in \mathcal{F}} u_i + \frac{1-\delta}{C} \cdot \left(1 - \frac{1}{e}\right) \cdot \sum_{i \in \mathcal{R}} u_i \right) \\
&= \frac{e-1}{2e-1} \cdot \mathbb{E}\left[\mathcal{R}(u_1^*, \ldots, u_n^*)\right] \cdot \frac{1}{C} \cdot \left(\sum_{i \in \mathcal{R}} u_i + \sum_{i \in \mathcal{F}} u_i \right) \\
&\geq \frac{e-1}{4e-2} \cdot \mathbb{E}\left[\mathcal{R}(u_1^*, \ldots, u_n^*)\right] \ ,
\end{aligned}$$

where the second inequality is derived from the upper bound in (4.2), and the last inequality follows from (4.3). Finally, as explained in step 1, we restore the original capacity by selecting at least half of the units stocked, based on their individual contributions to the lower bounds in (4.4) and (4.6). This alteration yields an approximation guarantee of $(e-1)/(8e-4)$.

For a more careful analysis, we need to distinguish in the lower bound above between the rounded capacity $\bar{C}$ and the initial capacity C, yielding

$$\mathbb{E}\left[\mathcal{R}(u_1, \ldots, u_n)\right] \geq \frac{e-1}{2e-1} \cdot \mathbb{E}\left[\mathcal{R}(u_1^*, \ldots, u_n^*)\right] \cdot \frac{1}{\bar{C}} \cdot \left(\sum_{i \in \mathcal{R}} u_i + \sum_{i \in \mathcal{F}} u_i \right) .$$

We now observe that, when restoring the original capacity C, our lower bound scales-down by a factor of C/β, where β is the total number of units in the relaxed solution, i.e., $\beta = \sum_{i \in \mathcal{R}} u_i + \sum_{i \in \mathcal{F}} u_i$. For this reason, we obtain an expected revenue of at least

$$\frac{e-1}{2e-1} \cdot \mathbb{E}\left[\mathcal{R}(u_1^*, \ldots, u_n^*)\right] \cdot \frac{\beta}{\bar{C}} \cdot \frac{C}{\beta} \geq \frac{e-1}{4e-2} \cdot \mathbb{E}\left[\mathcal{R}(u_1^*, \ldots, u_n^*)\right] ,$$

concluding the proof of Theorem 4.2.1.

4.2.3 Refined performance guarantee

A close investigation of our algorithm shows that the factor of $1 - 1/e$ is incurred due to employing a general-purpose maximum coverage algorithm to solve the static problem in step 1. However, for numerous special cases of preference lists (such as nested, intervals, laminar, just to name a few), this variant can be solved either exactly or within a greater degree of accuracy. The following claim explicitly relates between our approximation guarantee for the dynamic model and the best achievable one for the static variant.

Corollary 4.2.5. *Suppose that, for a certain class of preference lists, the static variant can be efficiently approximated within factor α. Then, the corresponding dynamic formulation, with identical prices and IFR demand distribution, admits an $\alpha/(2\alpha+2)$-approximation in polynomial time.*

It is important to point out that our results extend to the case where the distribution over preference lists is not explicitly specified as part of the input, potentially having an exponentially-large support. In fact, to efficiently implement our algorithm, we only require a polynomial-time procedure for computing the probability that each product in a given assortment is purchased under a single customer arrival.

Indeed, this property is sufficient to greedily approximate the static problem (step 1 in Section 4.2.1) and to compute each of the ψ_i-probabilities defining the newsvendor objective function. In particular, such procedures can easily be devised for most choice models proposed in the revenue management literature, including mixtures of logits Talluri and van Ryzin (2004), Rusmevichientong and Topaloglu (2012), nested logit Li et al. (2015), Davis et al. (2014), as well as the Markov chain model Blanchet et al. (2016), Feldman and Topaloglu (2014), Désir et al. (2015).

4.2.4 Price differentiation

Algorithm. In what follows, we explain how the algorithm of Section 4.2.1 can be adapted in the presence of price differentiation to obtain an $O(\log(P_{\max}/P_{\min}))$-approximation. The basic idea is based on the classify-and-select paradigm, where products are initially partitioned into classes with nearly-uniform prices, and then, we employ for each class our constant-factor approximation as a subroutine, treating these products as if they are associated with uniform prices. Specifically, assuming that products are indexed such that $P_1 \leq \cdots \leq P_n$, our algorithm picks the most profitable inventory vector out of $U^1, \ldots, U^K$, where $K = \lceil \log(P_n/P_1) \rceil$. Each vector U^k is generated as follows:

1. Let a_k be the minimum index of a product whose price is at least $P_1 \cdot 2^{k-1}$, that is, $a_k = \min\{i \in [n] : P_i \geq P_1 \cdot 2^{k-1}\}$. Given this parameter, we define the collection of products $\mathcal{A}_k = [a_k, a_{k+1} - 1]$.

2. The inventory vector $U^k = (u_1^k, \ldots, u_n^k)$ is constructed by applying the horizontal differentiation procedure (see Section 4.2.1) to the subproblem formed by products in $\mathcal{A}_k$, with identical prices of $\tilde{p}_{a_k} = \cdots = \tilde{p}_{a_{k+1}-1} = 1$.

It is worth noting that, in order to pick the "most profitable" inventory vector, we make the final comparisons in terms of the multi-item newsvendor lower bound, that can be computed in polynomial time.

Analysis. In order to establish the performance guarantee attained by our algorithm, we begin by highlighting a fundamental property of the consumption process under an arbitrary collection of preference lists. To avoid deviating from the overall discussion, we prove the next claim in Appendix B.1.1.

Lemma 4.2.6. *The expected revenue function* $\mathbb{E}[\mathcal{R}(\cdot)]$ *is subadditive.*

Now, for every $k \in [K]$, let $U^{*(k)}$ be the projection of the optimal inventory vector U^* on the products $\mathcal{A}_k$. That is, $U^{*(k)}$ is obtained from U^* by setting to 0 the inventory levels of all products in $[n] \setminus \mathcal{A}_k$. We proceed by showing that U^k, the inventory vector constructed earlier by our algorithm, generates a constant fraction of the expected revenue of $U^{*(k)}$.

Lemma 4.2.7. $\mathbb{E}[\mathcal{R}(U^k)] \geq \frac{e-1}{8e-4} \cdot \mathbb{E}[\mathcal{R}(U^{*(k)})]$.

Proof. Recall that the inventory vector U^k constitutes an $(e-1)/(4e-2)$-approximation for the optimal sales quantity (i.e., expected number of units purchased) when the underlying set of products is $\mathcal{A}_k$. Therefore, since $U^{*(k)}$ stocks only products in $\mathcal{A}_k$, the expected sales quantity with respect to U^k is at least $(e-1)/(4e-2)$ times the analogous quantity with respect to $U^{*(k)}$. The claim follows by observing that, by definition of $\mathcal{A}_k$, the ratio between the extremal prices within this class is at most 2. □

Based on the preceding discussion, we are now ready to show that the most profitable inventory vector out of $U^1, \ldots, U^K$ guarantees an expected revenue within factor $O(\log(P_n/P_1))$ of optimal. Indeed,

$$\begin{aligned}
\max_k \left\{\mathbb{E}\left[\mathcal{R}\left(U^k\right)\right]\right\} &\geq \frac{e-1}{8e-4} \cdot \max_k \left\{\mathbb{E}\left[\mathcal{R}\left(U^{*(k)}\right)\right]\right\} \\
&\geq \frac{e-1}{8e-4} \cdot \frac{1}{K} \cdot \sum_{k=1}^{K} \mathbb{E}\left[\mathcal{R}\left(U^{*(k)}\right)\right] \\
&\geq \frac{e-1}{8e-4} \cdot \frac{1}{K} \cdot \mathbb{E}\left[\mathcal{R}\left(U^*\right)\right] \\
&= \Omega\left(\frac{1}{\log(P_n/P_1)}\right) \cdot \mathbb{E}\left[\mathcal{R}\left(U^*\right)\right] ,
\end{aligned}$$

where the first inequality follows from Lemma 4.2.7 and the third inequality is implied by the subadditivity of the expected revenue function (see Lemma 4.2.6).

4.3 Approximation Algorithms for the Intervals Choice Model

In what follows, we consider the dynamic assortment planning problem under interval preference lists. When the number of customers M satisfies the IFR property, we show how to efficiently compute an inventory vector that approximates the optimal expected revenue within factor $O(\log\log(P_n/P_1))$, where P_1 and P_n are the minimal and maximal prices, respectively.

Since our approach employs recursive decompositions of the preference lists, it is instructive to start off by presenting some of the high-level ideas, followed by a simpler $O(\log n)$ approximation. We then explain how to make use of our algorithm for uniform prices, given in Section 4.2, to establish the main result of this section.

4.3.1 General outline

The main algorithmic idea, exploited in different forms in Sections 4.3.2 and 4.3.3, consists in partitioning the collection of preference lists $\mathcal{L}$ into a small number of classes, L. By separating customer purchases according to their different classes, the expected revenue function decomposes into L terms. Hence, to obtain an $O(L)$-approximation, we propose the following approach:

1. Consider separately each of the L subproblems, where the consumption process is limited to customers picking preference lists from a single class of the partition.

2. Approximately solve each of these L subproblems. The crux would be to design a partition such that the corresponding subproblems have a simplified structure, admitting constant-factor approximations.

3. Pick the best solution among these L inventory vectors.

However, this approach is generally insufficient to claim the desired approximation ratio. Indeed, the decomposition into separate subproblems does not take into account the dependency of the revenue functions across the different preference list classes, in the joint sequence of arrivals. In other words, the expected revenue restricted to a single class in the full consumption process (i.e., when all preference lists in $\mathcal{L}$ could be picked) is different from the one generated by the distribution induced on that class. Consequently, our decomposition approach may very well under-estimate the potential expected revenue.

Motivated by this observation, for any class of preference lists $\mathcal{V} \subseteq \mathcal{L}$ and inventory vector U, we distinguish between two types of revenues, captured by the following random variables:

- *Original model:* $\mathcal{R}^{+}_{\mathcal{V}}(U)$ designates the revenue generated by the arrival of M customers who draw a preference list in $\mathcal{V}$, assuming that the consumption process is formed by the original model, where all preference lists in $\mathcal{L}$ occur according to the initial distribution.

- *$\mathcal{V}$-restricted model:* $\mathcal{R}^{-}_{\mathcal{V}}(U)$ denotes the revenue generated by the arrival of M customers who draw a preference list in $\mathcal{V}$, assuming that the consumption process is formed by the $\mathcal{V}$-restricted model. Here, only preference lists in $\mathcal{V}$ can occur and their probabilities remain unchanged, whereas all lists in $\mathcal{L} \setminus \mathcal{V}$ are replaced by an empty list.

Now assume that the classes of our partition are denoted by $\mathcal{V}_1, \dots, \mathcal{V}_L$. Generally speaking, the expected revenues in the restricted and original models are unrelated; elementary examples demonstrate that neither one dominates the other. What we need to argue to utilize this approach is that these revenues are within constant factors of each other, due to the specific properties of our decomposition. Formally, for every $\ell \in [L]$, we construct a feasible inventory vector U_ℓ satisfying

$$\mathbb{E}\left[\mathcal{R}^{-}_{\mathcal{V}_\ell}(U_\ell)\right] = \Omega(1) \cdot \mathbb{E}\left[\mathcal{R}^{+}_{\mathcal{V}_\ell}(U^*)\right] \ , \tag{4.7}$$

where U^* is the optimal inventory vector for the original model.

As a result, the best inventory vector out of $U_1, \ldots, U_L$ guarantees an $O(L)$-approximation for the original model. Indeed, by considering any realization of the consumption process, it is easy to verify that the revenue generated in the original model is stochastically larger than that of the $\mathcal{V}_\ell$-restricted model. By combining this observation and equation (4.7),

$$\begin{aligned}\max_{\ell \in [L]} \mathbb{E}\left[\mathcal{R}(U_\ell)\right] &\geq \max_{\ell \in [L]} \mathbb{E}\left[\mathcal{R}^-_{\mathcal{V}_\ell}\left(U_\ell\right)\right] \\ &\geq \frac{1}{L} \cdot \sum_{\ell=1}^{L} \mathbb{E}\left[\mathcal{R}^-_{\mathcal{V}_\ell}\left(U_\ell\right)\right] \\ &= \Omega\left(\frac{1}{L}\right) \cdot \sum_{\ell=1}^{L} \mathbb{E}\left[\mathcal{R}^+_{\mathcal{V}_\ell}\left(U^*\right)\right] \\ &= \Omega\left(\frac{1}{L}\right) \cdot \mathbb{E}\left[\mathcal{R}\left(U^*\right)\right] \ .\end{aligned}$$

4.3.2 Logarithmic approximation in the number of products

We begin by describing our partition of the preference lists into $L = O(\log n)$ classes $\mathcal{V}_1, \ldots, \mathcal{V}_L$. For the resulting partition, we devise a polynomial-time algorithm for computing an inventory vector U_ℓ satisfying

$$\mathbb{E}\left[\mathcal{R}^-_{\mathcal{V}_\ell}(U_\ell)\right] \geq \frac{1}{8} \cdot \mathbb{E}\left[\mathcal{R}^+_{\mathcal{V}_\ell}(U^*)\right] \ .$$

4.3.2.1 The recursive decomposition

In order to formalize our decomposition approach, we first introduce a sequence of increasingly refined partitions of the products in $[n]$, denoted by $\mathcal{S}_1, \ldots, \mathcal{S}_L$. In turn, this sequence induced the desired partition of preference lists into $\mathcal{V}_1, \ldots, \mathcal{V}_L$.

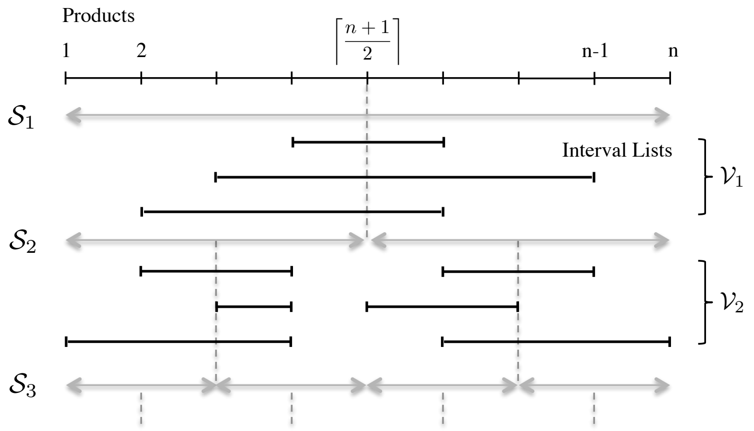
Partitions of products. We define the middle product of a segment $[a, b] \subseteq [n]$ as the product $\lceil (a+b)/2 \rceil$. The sequence $\mathcal{S}_1, \ldots, \mathcal{S}_L$ is obtained by the following recursive procedure:

- First, we define $\mathcal{S}_1$ as the trivial partition of $[n]$, comprised of a single segment consisting of all products, i.e., $\mathcal{S}_1 = \{[n]\}$.
- The next partition, $\mathcal{S}_2$, is obtained by breaking the segment $[n]$ at its middle product, that is, $\mathcal{S}_2 = \{[1, \lceil (n+1)/2 \rceil], [\lceil (n+1)/2 \rceil + 1, n]\}$.
- This process continues recursively, that is, we define $\mathcal{S}_\ell$ as the partition of products obtained by breaking each segment of $\mathcal{S}_{\ell-1}$ at its middle product into two parts.

Partition of preference lists. Given any subset of lists $\mathcal{V} \subseteq \mathcal{L}$ and a partition $\mathcal{S}$ of the products $[n]$ into pairwise-disjoint segments, we define $\mathrm{mid}(\mathcal{V}, \mathcal{S})$ as the subset of lists in $\mathcal{V}$ that contain the middle product of at least one segment in $\mathcal{S}$. With this definition at hand, we construct the partition of the preference lists into $\mathcal{V}_1, \ldots, \mathcal{V}_L$ as follows:

- First, we have $\mathcal{V}_1 = \mathrm{mid}(\mathcal{L}, \mathcal{S}_1)$.
- Then, $\mathcal{V}_2 = \mathrm{mid}(\mathcal{L} \setminus \mathcal{V}_1, \mathcal{S}_2)$.
- This process continues recursively, as illustrated in Figure 4-1. That is, we define $\mathcal{V}_\ell$ as the subset of residual preference lists that contain the middle product of a segment in $\mathcal{S}_\ell$, i.e., $\mathcal{V}_\ell = \mathrm{mid}(\mathcal{L} \setminus (\cup_{j=1}^{\ell-1} \mathcal{V}_j), \mathcal{S}_\ell)$.

Figure 4-1: The recursive decomposition of $\mathcal{L}$ into $\mathcal{V}_1, \ldots, \mathcal{V}_L$.



Structural properties. Since the maximum length of any segment shrinks by a constant factor at each level of the decomposition, it immediately follows that the resulting number of classes is $L = O(\log n)$. In addition, each partition of products $\mathcal{S}_\ell$ can be viewed as a collection of pairwise-disjoint segments, satisfying the next two properties:

- *Property 1:* each interval list in $\mathcal{V}_\ell$ is fully contained in precisely one of the segments in $\mathcal{S}_\ell$.
- *Property 2:* for each segment $S \in \mathcal{S}_\ell$, there exists a product in S that intersects all intervals in $\mathcal{V}_\ell$ contained in this segment.

These are precisely the sufficient properties that will enable us to compute a feasible inventory vector U_ℓ satisfying

$$\mathbb{E}\left[\mathcal{R}^-_{\mathcal{V}_\ell}(U_\ell)\right] \geq \frac{1}{8} \cdot \mathbb{E}\left[\mathcal{R}^+_{\mathcal{V}_\ell}(U^*)\right] \ .$$

4.3.2.2 Proving the existence of U_ℓ

Single segment analysis. In order to construct U_ℓ, it is sufficient to show that, for every segment of products S in the partition $\mathcal{S}_\ell$, there exists an inventory vector U^S_ℓ such that:

- The vector U^S_ℓ only makes use of products in S.
- Letting $\mathcal{V}^S_\ell$ be the set of interval preference lists in $\mathcal{V}_\ell$ that are fully contained in S, we have
$$\mathbb{E}\left[\mathcal{R}^-_{\mathcal{V}^S_\ell}\left(U^S_\ell\right)\right] \geq \frac{1}{8} \cdot \mathbb{E}\left[\mathcal{R}^+_{\mathcal{V}^S_\ell}\left(U^*\right)\right] \ . \tag{4.8}$$
- The combined number of units stocked in $\{U^S_\ell : S \in \mathcal{S}_\ell\}$ is at most C.

Indeed, given property 1, since the segments in $\mathcal{S}_\ell$ are pairwise disjoint, the expected revenue of the combined vector $\sum_{S \in \mathcal{S}_\ell} U^S_\ell$ decomposes into the sum of expected revenues generated by each vector U^S_ℓ. This decomposition applies in both the original model and the $\mathcal{V}_\ell$-restricted model. In other words, assuming that each U_ℓ^S satisfies inequality (4.8), we obtain the desired inequality:

$$\begin{aligned}
\mathbb{E}\left[\mathcal{R}^-_{\mathcal{V}_\ell}\left(\sum_{S\in\mathcal{S}_\ell} U_\ell^S\right)\right] &= \sum_{S\in\mathcal{S}_\ell} \mathbb{E}\left[\mathcal{R}^-_{\mathcal{V}_\ell^S}\left(U_\ell^S\right)\right] \\
&\geq \frac{1}{8}\cdot\sum_{S\in\mathcal{S}_\ell} \mathbb{E}\left[\mathcal{R}^+_{\mathcal{V}_\ell^S}\left(U^*\right)\right] \\
&= \frac{1}{8}\cdot\mathbb{E}\left[\mathcal{R}^+_{\mathcal{V}_\ell}\left(U^*\right)\right] \ .
\end{aligned}$$

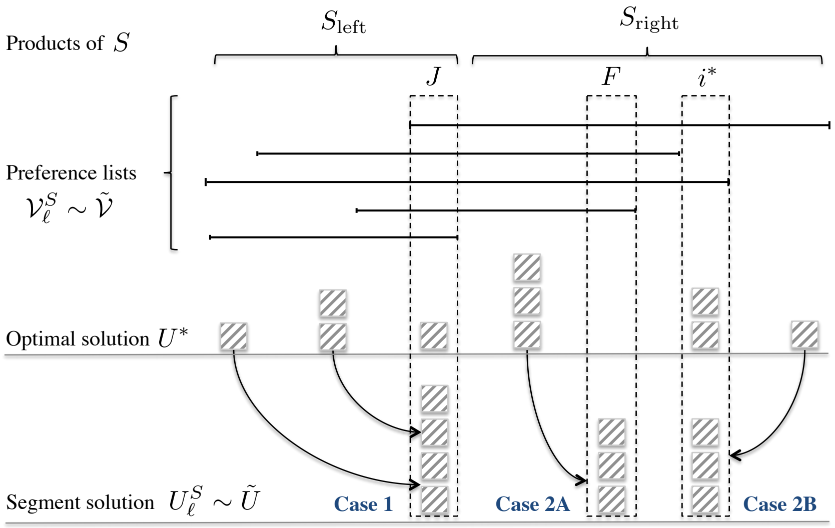
Simplified problem. The preceding discussion implies that we can focus on a single segment S from this point on. We prove the existence of an inventory vector U_ℓ^S that satisfies the above properties by analyzing the revenue generated under the optimal vector U^* by the lists in $\mathcal{V}_\ell^S$. In fact, in the course of proving the existence of U_ℓ^S, we implicitly describe an efficient algorithmic procedure to construct such a vector. To simplify the presentation, the corresponding algorithm is made explicit in Section 4.3.2.3. Also, we use simplified notation throughout this section, where $\mathcal{V}_\ell^S$ and U_ℓ^S are replaced by $\tilde{\mathcal{V}}$ and $\tilde{U}$, respectively, i.e., we do not explicitly mention the dependency of these variables on S and ℓ.

Revenue decomposition. Based on property 2, we define J to be the highest index product that intersects all intervals in $\tilde{\mathcal{V}}$. We now break the segment S into a left part $S_{\text{left}} = S \cap [1, J]$ and a right part $S_{\text{right}} = S \cap [J+1, n]$, noting that the latter part could be empty. These definitions, in turn, are used to further divide the expected revenue $\mathbb{E}[\mathcal{R}^+_{\tilde{\mathcal{V}}}(U^*)]$ based on whether units are purchased from the left or right part of S, that is,

$$\mathbb{E}\left[\mathcal{R}^+_{\tilde{\mathcal{V}}}(U^*)\right] = \mathbb{E}\left[\mathcal{R}^+_{\tilde{\mathcal{V}},S_{\text{left}}}(U^*)\right] + \mathbb{E}\left[\mathcal{R}^+_{\tilde{\mathcal{V}},S_{\text{right}}}(U^*)\right] \ .$$

The proof proceeds by considering two cases, depending on whether most of the expected revenue is coming from the left or right parts. To better understand this case analysis, we advise the reader to consult Figure 4-2.

Figure 4-2: The inventory vectors examined by the algorithm to construct U_ℓ^S.



Case 1: $\boldsymbol{E[R^{+}{}_{\tilde{V},S_{left}}(U^*)] \geq E[R^{+}{}_{\tilde{V}}(U^*)]/2}$. This case can be handled rather easily. To construct the inventory vector $\tilde{U}$, we consider the restriction of the optimal vector U^* to the left segment $S_{\text{left}} = S \cap [1, J]$, and relocate all stocked units to product J. It is not difficult to verify that, in every realization of the consumption process, the number of units of the left segment S_{left} consumed in the $\tilde{\mathcal{V}}$-restricted model is greater or equal to the number of units consumed in the original model by preference lists in $\tilde{\mathcal{V}}$. In addition, J is the most expensive product in S_{left}, meaning that $\mathcal{R}^{-}{}_{\tilde{\mathcal{V}},S_{\text{left}}}(\tilde{U}) \geq_{\text{st}} \mathcal{R}^{+}{}_{\tilde{\mathcal{V}},S_{\text{left}}}(U^*)$, and therefore

$$\mathbb{E}\left[\mathcal{R}^{-}{}_{\tilde{\mathcal{V}}}\left(\tilde{U}\right)\right] = \mathbb{E}\left[\mathcal{R}^{-}{}_{\tilde{\mathcal{V}},S_{\text{left}}}\left(\tilde{U}\right)\right] \geq \mathbb{E}\left[\mathcal{R}^{+}{}_{\tilde{\mathcal{V}},S_{\text{left}}}\left(U^*\right)\right] \geq \frac{1}{2} \cdot \mathbb{E}\left[\mathcal{R}^{+}{}_{\tilde{\mathcal{V}}}\left(U^*\right)\right] \ ,$$

where the first equality holds since $\tilde{U}$ only contains the product J, and the last inequality is due to the case hypothesis.

Case 2: $\boldsymbol{E[R^{+}{}_{\tilde{V},S_{right}}(U^*)] \geq E[R^{+}{}_{\tilde{V}}(U^*)]/2}$. This case is more involved. Let us focus on some product i in the segment $S_{\text{right}} = S \cap [J+1, n]$, and let ψ_i be the probability that an arriving customer picks one of the intervals in $\tilde{\mathcal{V}}$ that contains i (necessarily on its right part). Note that since all non-empty right parts have $J+1$ as a left endpoint, it follows that $\psi_{J+1} \geq \psi_{J+2} \geq \cdots$. We begin by defining a pair of random variables, whose exact meaning will be revealed later on. These are $Y_i \sim B(M, \psi_i)$ and $\bar{Y}_i = \min\{Y_i, C^*_S\}$, where C^*_S is the total capacity used by the optimal inventory vector U^* over the segment S_{right}.

With these random variables at hand, we say that product i is frequent when $\mathbb{E}[\bar{Y}_i] \geq C^*_S/2$. Otherwise, this product is rare. Since $\psi_{J+1} \geq \psi_{J+2} \geq \cdots$, it follows that there is a product F such that the set of frequent products $\mathcal{F}$ is precisely those in $[J+1, F]$, whereas the rare ones $\mathcal{R}$ are those in $S_{\text{right}} \setminus [J+1, F]$. As a result, we can break the revenue $\mathcal{R}^+_{\tilde{\mathcal{V}},S_{\text{right}}}(U^*)$ into purchases of frequent and rare products, obtaining that

$$\mathbb{E}\left[\mathcal{R}^+_{\tilde{\mathcal{V}},\mathcal{F}}(U^*)\right] + \mathbb{E}\left[\mathcal{R}^+_{\tilde{\mathcal{V}},\mathcal{R}}(U^*)\right] = \mathbb{E}\left[\mathcal{R}^+_{\tilde{\mathcal{V}},S_{\text{right}}}(U^*)\right] \geq \frac{1}{2} \cdot \mathbb{E}\left[\mathcal{R}^+_{\tilde{\mathcal{V}}}(U^*)\right] ,$$

where the last inequality follows from the case hypothesis. It remains to consider two cases, depending on which set of products (frequent or rare) is contributing more in the above inequality.

Case 2A: $\boldsymbol{E[R^{+}{}_{\tilde{V},F}(U^*)]{\geq}E[R^{+}{}_{\tilde{V}}(U^*)]/4}$. Note that $C^*_S \cdot P_F$ is a trivial upper bound on the random variable $\mathcal{R}^+_{\tilde{\mathcal{V}},\mathcal{F}}(U^*)$, and consequently on its expectation. This follows by observing that, in each realization, at most C^*_S units are purchased among frequent products, and each purchase generates a revenue of at most P_F, given that F is the right endpoint of $\mathcal{F}$. To construct the inventory vector $\tilde{U}$, we simply stock C^*_S units of F, the most expensive frequent product. We observe that in the $\tilde{\mathcal{V}}$-restricted model, the distribution of the number of units consumed is identical to that of $\bar{Y}_F$. Indeed, similar to the consumption process considered in Section 4.2, under the $\tilde{\mathcal{V}}$-restricted model, a unit of product F is consumed with probability ψ_F as long as this product has not stocked-out, corresponding to a sequence of M independent Bernoulli trials whose sum is capped by the capacity C^*_S. This means that the expected revenue under the restricted model would be

$$\begin{aligned}
\mathbb{E}\left[\mathcal{R}^{-}{}_{\tilde{\mathcal{V}}}\left(\tilde{U}\right)\right] &= P_F \cdot \mathbb{E}\left[\bar{Y}_F\right] \\
&\geq P_F \cdot \frac{C_S^*}{2} \\
&\geq \frac{1}{2} \cdot \mathbb{E}\left[\mathcal{R}^{+}{}_{\tilde{\mathcal{V}},\mathcal{F}}\left(U^*\right)\right] \\
&\geq \frac{1}{8} \cdot \mathbb{E}\left[\mathcal{R}^{+}{}_{\tilde{\mathcal{V}}}\left(U^*\right)\right] \ ,
\end{aligned}$$

where the first inequality follows from the definition of frequent products while the last inequality is due to the case hypothesis.

Case 2B: $\boldsymbol{E[R^{+}{}_{\tilde{V},R}(U^*)]\geq E[R^{+}{}_{\tilde{V}}(U^*)]/4}$. Here, we derive an upper bound on the expected revenue of $\mathcal{R}^{+}{}_{\tilde{\mathcal{V}},\mathcal{R}}(U^*)$ by considering an unrealistic model, where prior to the arrival of any customer, the current inventory vector can be re-optimized. Specifically, suppose that we may substitute to any vector (still, using only rare products), without any capacity restrictions. In this model, since we are only interested in maximizing the expected revenue due to purchases made by (the right part of) intervals in $\tilde{\mathcal{V}}$, the optimal strategy is to stock a single unit of i^*, which is the product that maximizes $\psi_i P_i$ over all rare products. Indeed, assuming i is the minimal-index product stocked, the expected revenue generated by a single arrival of the lists in $\tilde{\mathcal{V}}$ is exactly $\psi_i P_i$. Therefore,

$$\mathbb{E}\left[\mathcal{R}^{+}_{\tilde{\mathcal{V}},\mathcal{R}}\left(U^*\right)\right] \leq \mathbb{E}\left[M\right] \cdot \psi_{i^*} P_{i^*} \ .$$

Now, to construct the inventory vector $\tilde{U}$, we simply stock C_S^* units of product i^*. Once again, in the $\tilde{\mathcal{V}}$-restricted model, the distribution of the number of units consumed will be identical to that of $\bar{Y}_{i^*} \sim B(M, \psi_{i^*})$. Indeed, similar to the consumption process considered in Section 4.2, under the $\tilde{\mathcal{V}}$-restricted model, a unit of product i^* is consumed with probability ψ_{i^*} as long as this product has not stocked-out. Therefore, the resulting expected revenue is

$$\begin{aligned}
\mathbb{E}\left[\mathcal{R}^{-}_{\tilde{\mathcal{V}}}\left(\tilde{U}\right)\right] &= P_{i^*} \cdot \mathbb{E}\left[\bar{Y}_{i^*}\right] \\
&\geq P_{i^*} \cdot \frac{\mathbb{E}\left[Y_{i^*}\right]}{2} \\
&= \frac{1}{2} \cdot \mathbb{E}\left[M\right] \cdot \psi_{i^*} P_{i^*} \\
&\geq \frac{1}{2} \cdot \mathbb{E}\left[\mathcal{R}^{+}_{\tilde{\mathcal{V}},\mathcal{R}}\left(U^*\right)\right] \\
&\geq \frac{1}{8} \cdot \mathbb{E}\left[\mathcal{R}^{+}_{\tilde{\mathcal{V}}}\left(U^*\right)\right] ,
\end{aligned}$$

where the first inequality follows from Lemmas 4.2.3 and 4.2.4, recalling that i^* is a rare product, and the last inequality is due to the case hypothesis.

4.3.2.3 Dynamic program

A careful review of the arguments used to prove the existence of U_ℓ reveals that we actually describe an efficient way to construct this vector, assuming that the number of units C_S of the optimal solution within each interval $S \in \mathcal{S}_\ell$ is known a-priori. We prove that this assumption is not needed, explaining why a similar approximation ratio can be attained by means of dynamic programming. In the following, J, F, and i^* play precisely the same roles as in the previous section. It is not difficult to verify that each of these products can easily be identified in polynomial time.

The general idea is to formulate a dynamic program that tries out all feasible capacities for each single segment $S \in \mathcal{S}_\ell$, and chooses the best vector among those constructed in cases 1, 2A, and 2B. Specifically, for any positive capacity c, we define $U^{(1)}_{S,c}$ as the inventory vector described in case 1 that stocks c units of product J. Similarly, $U^{(2A)}_{S,c}$ is the inventory vector described in case 2A that stocks c units of product F, and $U^{(2B)}_{S,c}$ is the vector of case 2B that stocks c units of product i^*. One important observation is that we can efficiently compute the expected revenue associated with each of these vectors under the $\mathcal{V}^S_\ell$-restricted model, as it is equivalent to computing the expected value of a truncated binomial random variable. Finally, we let $\{S_1, \ldots, S_r\}$ designate the segments of products in the partition $\mathcal{S}_\ell$, numbered in increasing order of product indices.

For any $j \in [r]$ and $\bar{c} \in [C]$, we define the objective function $G(j, \bar{c})$ as the maximal expected revenue in the restricted model, generated by a vector with at most $\bar{c}$ units, obtained by concatenating the candidate solutions $U^{(1)}_{S,c}$ or $U^{(2A)}_{S,c}$ or $U^{(2B)}_{S,c}$, over the first j segments of the partition, $S_1, \ldots, S_j$. It is not difficult to verify that the function G satisfies the following recursion formula:

$$G(j, \bar{c}) = \max_{c \leq \bar{c}} \left\{ G\left(j-1, \bar{c}-c\right) + \max\left\{ \mathbb{E}\left[\mathcal{R}^{-}_{\mathcal{V}^S_\ell}\left(U^{(1)}_{S_j,c} \right)\right], \mathbb{E}\left[\mathcal{R}^{-}_{\mathcal{V}^S_\ell}\left(U^{(2A)}_{S_j,c} \right)\right], \mathbb{E}\left[\mathcal{R}^{-}_{\mathcal{V}^S_\ell}\left(U^{(2B)}_{S_j,c} \right)\right] \right\} \right\} .$$

By solving this recursion forward, we infer the quantity $G(r, C)$. Given the optimality conditions satisfied by the above dynamic program, it follows that $G(r, C) \geq \mathbb{E}[\mathcal{R}^{-}_{\mathcal{V}^S_\ell}(U_\ell)]$, where U_ℓ is the vector constructed in Section 4.3.2.2. Indeed, the allocation of capacity across the different segments, as described in the previous section, can be replicated by the dynamic program, and for each segment, the dynamic program selects a vector that maximizes the expected revenue.

4.3.2.4 Running time analysis

The recursive decomposition is computed in time $O(|\mathcal{L}| \cdot \log n)$. Indeed, we identify the location of each interval list with respect to at most $\lceil \log n \rceil$ middle products along the recursion. Next, the dynamic program in Section 4.3.2.3 is solved in time $O(|\mathcal{S}_\ell| \cdot C^2 \bar{M})$ for the collection of preference lists $\mathcal{V}_\ell$. Indeed, the state space has size $O(|\mathcal{S}_\ell| \cdot C)$. Each recursive step is computed in $O(\bar{M}C)$ time by comparing the expected revenues of three inventory vectors, each stocking C units. The latter require to evaluate the expectations of truncated binomial variables, with at most $\bar{M}$ trials. Summing over all $\ell \in [L]$, the overall running time is $O(|\mathcal{L}| \cdot \log n + nC^2\bar{M})$.

4.3.3 Log-logarithmic approximation in the price ratio

In this section, we explain how the main technical ideas of Section 4.3.2 can be utilized in order to attain an approximation guarantee of $O(\log \log(P_n/P_1))$, where P_n and P_1 stand for the maximum and minimum price of any product. Here, we employ a different decomposition of $\mathcal{L}$, that allows us to make use of the constant-factor approximation for uniform prices (see Section 4.2) as a subroutine.

4.3.3.1 The recursive decomposition

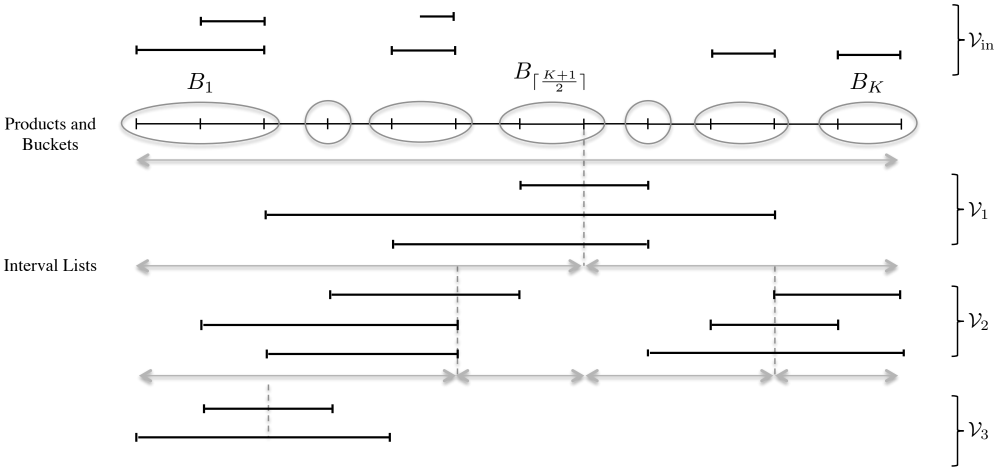
We initially break the interval $[1, n]$ into $K = O(\log(P_n/P_1))$ buckets $B_1, \ldots, B_K$ according to prices, geometrically by powers of 2. That is, the first bucket B_1 consists of products with prices in $[P_1, 2P_1)$, the second bucket B_2 corresponds to prices in $[2P_1, 2^2P_1)$, so forth and so on.

The recursive partition here resembles the one in Section 4.3.2.1, at the exception that segments are now defined with respect to the indexing $1, \ldots, K$, and the middle product depends on the collection of buckets $B_1, \ldots, B_K$. Specifically, the middle product associated with a segment $[a, b] \subseteq [K]$ is defined as the right-most product of the middle bucket $B_{\lceil (a+b)/2 \rceil}$. Given any subset of lists $\mathcal{V} \subseteq \mathcal{L}$ and a partition $\mathcal{K}$ of $[K]$ into pairwise-disjoint segments, we define $\text{mid}(\mathcal{V}, \mathcal{K})$ as the set of interval lists in $\mathcal{V}$ that contain the middle product of at least one segment in $\mathcal{K}$. With this definition at hand, we define the classes of lists $\mathcal{V}_1, \ldots, \mathcal{V}_L, \mathcal{V}_{\text{in}}$ as follows:

- The special class $\mathcal{V}_{\text{in}}$ is comprised of all intervals in $\mathcal{L}$ that are fully contained in one of the buckets $B_1, \ldots, B_K$.
- The remaining classes are determined as follows:
  - First, we have $\mathcal{V}_1 = \text{mid}(\mathcal{L} \setminus \mathcal{V}_{\text{in}}, \mathcal{K}_1)$, where $\mathcal{K}_1 = \{[K]\}$.
  - Then, $\mathcal{V}_2 = \text{mid}(\mathcal{L} \setminus (\mathcal{V}_1 \cup \mathcal{V}_{\text{in}}), \mathcal{K}_2)$, where $\mathcal{K}_2$ is obtained by breaking the segment $[K]$ at its middle product.
  - This process continues recursively, as illustrated in Figure 4-3. That is, we define $\mathcal{K}_\ell$ as the partition of $[K]$ obtained by breaking each segment of $\mathcal{K}_{\ell-1}$ at its middle product into two parts. Then, $\mathcal{V}_\ell$ is the residual subset of lists that contain the middle product of at least one segment in $\mathcal{K}_\ell$, i.e., $\mathcal{V}_\ell = \text{mid}(\mathcal{L} \setminus ((\cup_{j=1}^{\ell-1} \mathcal{V}_j) \cup \mathcal{V}_{\text{in}}), \mathcal{K}_\ell)$.

The decomposition above terminates as soon as we reach a level L, where $\mathcal{K}_L$ consists of only singletons of $[1, K]$. Once again, since the maximum length of any segment shrinks by a constant factor at each level, it follows that the depth of this decomposition is $L = O(\log K) = O(\log\log(P_n/P_1))$.

Figure 4-3: The decomposition of $\mathcal{L}$ into $\mathcal{V}_1, \ldots, \mathcal{V}_L, \mathcal{V}_{\text{in}}$.



4.3.3.2 Proving the existence of U_ℓ and U_{in}

We now argue that there is an efficient way of meeting the fundamental inequality (4.7) that relates between the restricted and original models for each class of the partition. Formally, for every $\ell \in [L]$, we devise a polynomial-time procedure to compute a feasible inventory vector U_ℓ satisfying

$$\mathbb{E}\left[\mathcal{R}^-_{\mathcal{V}_\ell}(U_\ell)\right] \geq \frac{1}{8} \cdot \mathbb{E}\left[\mathcal{R}^+_{\mathcal{V}_\ell}(U^*)\right] \ ,$$

where U^* is the optimal inventory vector for the original model. We also construct U_{in} such that

$$\mathbb{E}\left[\mathcal{R}^-_{\mathcal{V}_{\text{in}}}(U_{\text{in}})\right] = \Omega(1) \cdot \mathbb{E}\left[\mathcal{R}^+_{\mathcal{V}_{\text{in}}}(U^*)\right] \ .$$

Following the discussion in Section 4.3.1, we obtain an $O(\log\log(P_n/P_1))$ approximation for the original model by picking the best vector out of $U_1, \ldots, U_L, U_{\text{in}}$.

Handling $\boldsymbol{V_1, \ldots, V_L}$. The important observation is that the intervals in each class $\mathcal{V}_\ell$ satisfy the sufficient properties mentioned at the end of Section 4.3.2.1. For this reason, the exact same algorithm, now applied to a different collection of segments, enables us to compute a feasible inventory vector U_ℓ such that

$$\mathbb{E}\left[\mathcal{R}^-_{\mathcal{V}_\ell}(U_\ell)\right] \geq \frac{1}{8} \cdot \mathbb{E}\left[\mathcal{R}^+_{\mathcal{V}_\ell}(U^*)\right] \ .$$

Handling $\boldsymbol{V_{in}}$. Let us focus attention on a single bucket B, corresponding to a power-of-2 price range, say $[\Delta, 2\Delta)$, where the set of intervals contained in this bucket are denoted by $\mathcal{V}^B_{\text{in}}$. The important observation is that, in every realization, the number of units consumed in the $\mathcal{V}^B_{\text{in}}$-restricted model is greater or equal to the number of units consumed in the original model by these intervals. Indeed, this claim can be proven inductively over the arrival rank of customers and by arguing that, at any point in time during the arrival sequence, the number of units left within each interval in $\mathcal{V}^B_{\text{in}}$ in the former model is greater or equal to the corresponding number of units in the latter model. Therefore, since all products in bucket B have prices in $[\Delta, 2\Delta)$, it follows that $\mathcal{R}^-_{\mathcal{V}^B_{\text{in}}}(U^*) \geq_{\text{st}} \mathcal{R}^+_{\mathcal{V}^B_{\text{in}}}(U^*)/2$, and consequently,

$$\mathbb{E}\left[\mathcal{R}^-_{\mathcal{V}^B_{\text{in}}}(U^*)\right] \geq \frac{1}{2} \cdot \mathbb{E}\left[\mathcal{R}^+_{\mathcal{V}^B_{\text{in}}}(U^*)\right] \ . \tag{4.9}$$

Now, based on our constant-factor approximation for uniform prices (see Section 4.2), assuming that the capacity used by the optimal vector U^* over the bucket B is known in advance, we can efficiently compute an inventory vector U^B_{in} satisfying

$$\mathbb{E}\left[\mathcal{R}^-_{\mathcal{V}^B_{\text{in}}}\left(U^B_{\text{in}}\right)\right] = \Omega(1) \cdot \mathbb{E}\left[\mathcal{R}^-_{\mathcal{V}^B_{\text{in}}}(U^*)\right] = \Omega(1) \cdot \mathbb{E}\left[\mathcal{R}^+_{\mathcal{V}^B_{\text{in}}}(U^*)\right] \ ,$$

where the last equation follows from (4.9). By gluing the inventory vectors U^B_{in} over all buckets $B_1, \ldots, B_K$, we obtain an expected revenue of $\Omega(1) \cdot \mathbb{E}[\mathcal{R}^+_{\mathcal{V}_{\text{in}}}(U^*)]$. Finally, since the capacities used by U^* are not known a-priori, the assumption above can be bypassed by means of dynamic programming, similar to that of Section 4.3.2.3.

4.4 Nested Choice Model and General Demand Distribution

In this section, we provide a constant-factor approximation for the nested choice model under a general (non-IFR) demand distribution. This result is obtained through a sequence of structural transformations, allowing us to formulate the resulting instance as a (monotone) submodular maximization problem subject to a cardinality constraint. By leveraging the existing machinery in this context, we derive the following result.

Theorem 4.4.1. *Under the nested choice model, the dynamic assortment planning problem can be approximated within factor $1 - 1/e$ in polynomial time.*

4.4.1 Technical overview

For ease of exposition, we focus here on presenting the overall idea, and defer most of the technicalities to Sections 4.4.2 and 4.4.3.

Selection step: elimination of suboptimal products. The first step consists in simplifying the problem by identifying a well-structured collection of products, while preserving the optimal expected revenue. We begin by defining the quantity r_i, for each product $i \in [n]$, that denotes the expected revenue generated by a single customer arrival assuming that i is the most preferred product available. Namely, $r_i = P_i \cdot \sum_{\ell \in \mathcal{L}_i} \lambda_\ell$, where $\mathcal{L}_i \subseteq \mathcal{L}$ is the subset of lists containing product i. Next, we define the i-maximal product as the highest-index product that maximizes the quantity r_j over $j \in [i, n]$. We show that, without any loss in optimality, we can restrict our attention to assortments included in the collection of i-maximal products, over all $i \in [n]$. This subset of products is designated by $\mathcal{V}$, while $\mathcal{V}(i)$ denotes the i-maximal product.

Lemma 4.4.2. *There exists an optimal inventory vector that stocks only products of $\mathcal{V}$.*

The proof of this claim is given in Section 4.4.2. The main observation is that there is no point in stocking any of the products strictly between two successive products in $\mathcal{V}$. Specifically, we prove that the expected revenue can only increase by shifting any unit of product $i \in [n] \setminus \mathcal{V}$ to the i-maximal product $\mathcal{V}(i)$. As a result, while preserving the optimal revenue, all products in $[n] \setminus \mathcal{V}$ are eliminated. Thus, we assume from this point on that r_i is non-increasing over $i \in [n]$.

Greedy step: set decision formulation. We now argue that the problem can equivalently be recast as the maximization of a set function subject to a capacity constraint. This modified problem is referred to as the 'set decision' formulation hereafter. Specifically, each product is represented by C distinct copies of identical price, that are consecutive in the preference order. (Recall that C represents the maximal number of units due to the capacity constraint.) Thus, there are exactly $N = n \cdot C$ distinct products. Each preference list is now represented by the interval of $[N]$ containing all copies of its initial products. Finally, a decision is made relative to each product, whether to stock it or not in the assortment. In other words, the inventory level decisions are replaced by a set decision over products. In this new formulation, our objective is to maximize the expected revenue over all subsets of products that satisfy the cardinality constraint.

Establishing submodularity. The expected revenue generated by a subset $S \subseteq [N]$ is denoted by $f(S)$. We now state our main technical result, which is established in Section 4.4.3.

Lemma 4.4.3. *The set function $f : 2^N \to \mathbb{R}^+$ is submodular and monotone.*

Interestingly, submodularity does not hold for arbitrary instances of the nested choice model, i.e., ones that were not processed by our elimination procedure, as demonstrated in Lemma B.1.2. In fact, the revenue function is also not concave, as we argue in Lemma B.1.3. To avoid deviating from the overall discussion, the proofs of these claims are given in Appendix B.1.3.

Submodular maximization problems have extensively been studied in combinatorial optimization, and in particular, when the input function is also monotone, this problem can be approximated within factor $1 - 1/e$ under a cardinality constraint Nemhauser et al. (1978). Moreover, the algorithm thereof is based on a greedy procedure that admits very efficient implementations when the function has an evaluation oracle. In our particular case, Goyal et al. Goyal et al. (2016) showed that the revenue function can be evaluated by dynamic programming in time $O(\bar{M} \cdot N \cdot k)$, where $\bar{M}$ is the maximal number of arrivals. By leveraging this algorithm, Theorem 4.4.1 immediately follows.

4.4.2 Proof of Lemma 4.4.2

For any inventory vector U and integer m, we define the random variable $\alpha_m(U)$ to denote the most preferred product available (i.e., with positive inventory level) for the m-th arriving customer, when initially stocking the vector U. If there are fewer than m arrivals, or if no units are left, $\alpha_m(U) = \infty$, denoting a dummy product with price $P_\infty = 0$. With this definition, the expected revenue function can be rewritten by conditioning on the most preferred product available upon each arrival, yielding

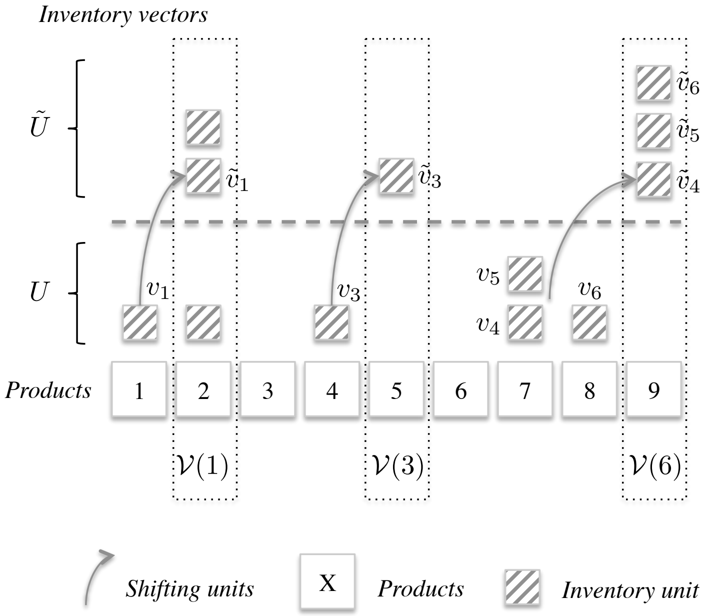
$$\mathbb{E}\left[\mathcal{R}(U)\right] = \mathbb{E}\left[\sum_{m=1}^{\infty} r_{\alpha_m(U)}\right] . \tag{4.10}$$

The desired result is proven by mapping any inventory vector $U = (u_1, \ldots, u_n)$ to a vector $\tilde{U} = (\tilde{u}_1, \ldots, \tilde{u}_n)$ that only stocks products in $\mathcal{V}$, has the same number of units, and generates at least as much expected revenue as U. The vector $\tilde{U}$ is constructed as follows: each unit of product i in U is represented in $\tilde{U}$ by a distinct unit of the maximal product $\mathcal{V}(i)$. That is,

$$\tilde{u}_i = \begin{cases} \sum_{k:\mathcal{V}(k)=i} u_k & \text{if } i = \mathcal{V}(i), \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, the number of units in $\tilde{U}$ is identical to that of U. This construction is illustrated in Figure 4-5.

Figure 4-4: Construction of $\tilde{U}$ by shifting each unit in U towards its corresponding maximal product.



Claim 4.4.4. *For $m \geq 1$, every realization of the consumption process satisfies $\alpha_m(\tilde{U}) \leq \mathcal{V}(\alpha_m(U))$.*

Proof. To prove this inequality, we interpret each inventory vector as a non-decreasing sequence of products, that enumerates (with repetition) the units in the preference order. In other words, if X units are stocked of a given product, then this product is repeated X times consecutively in the sequence. Let $(v_j)_{j \leq C}$ and $(\tilde{v}_j)_{j \leq C}$ denote the sequences associated with the inventory vectors U and $\tilde{U}$, respectively. Without loss of generality, we assume that these vectors stock precisely C units.

Our construction of $\tilde{U}$ implies that $v_k \leq \tilde{v}_k = \mathcal{V}(v_k)$. Now assume that, just before the m-th arrival, the most preferred product remaining when stocking initially the vector $\tilde{U}$ corresponds to the j-th unit of the sequence, meaning that $\alpha_m(\tilde{U}) = \tilde{v}_j$, and that all units in $\{v_1, \ldots, v_{j-1}\}$ have been consumed by previously arriving customers. Each of these units can be mapped to the arrival rank of the customer who purchases it, entailing the subsequence of arrivals $(m_1, \ldots, m_{j-1})$. Clearly, each product v_k

belongs to the preference list L_{m_k}. The key observation is that, since $v_k \leq \tilde{v}_k$, the preference list L_{m_k} also contains product v_k for each $k \in [j-1]$. Consequently, when initially stocking the vector U, after the first $m-1$ arrivals of customers, whereby the preference lists $(L_{m_1}, \ldots, L_{m_{j-1}})$ occurred precisely in this order, the units $\{v_1, \ldots v_{j-1}\}$ are consumed as well. Indeed, each unit v_k would necessarily be consumed by the list L_{m_k}, if it were not purchased by a previously arriving customer. We thus obtain that $v_j \leq \alpha_m(U)$, meaning that $\alpha_m(\tilde{U}) = \mathcal{V}(v_j) \leq \mathcal{V}(\alpha_m(U))$. □

To conclude Lemma 4.4.2, recall that r_i is the expected revenue generated by a single arrival, conditional on product i being the most preferred one available. Therefore, the expected revenue generated by the m-th arrival satisfies

$$r_{\alpha_m(U)} \leq r_{\mathcal{V}(\alpha_m(U))} \leq r_{\mathcal{V}(\alpha_m(\tilde{U}))} = r_{\alpha_m(\tilde{U})} \ ,$$

where the first inequality follows from the definition of maximal products, and the second inequality is due to Claim 4.4.4 and the monotonicity of $(r_i)_{i \in \mathcal{V}}$. Therefore, based on the revenue decomposition given by equation (4.10), we conclude that

$$\mathbb{E}\left[\mathcal{R}(U)\right] = \sum_{m=1}^{\infty} \mathbb{E}\left[r_{\alpha_m(U)}\right] \leq \sum_{m=1}^{\infty} \mathbb{E}\left[r_{\alpha_m\left(\tilde{U}\right)}\right] = \mathbb{E}\left[\mathcal{R}(\tilde{U})\right] \ .$$

4.4.3 Proof of Lemma 4.4.3

Notation. Following the previous section, for every subset $S \subseteq [N]$ we define $\alpha_m(S)$ as the most preferred product available at the m-th arrival, when initially stocking the set S. If all products have stocked out, or if the number of arrivals is smaller than m, the value of $\alpha_m(S)$ is set to ∞, which corresponds to a dummy product with price 0. Using these random variables, the expected revenue can be decomposed similar to equation (4.10), namely $f(S) = \mathbb{E}[\sum_{m=1}^{\infty} r_{\alpha_m(S)}]$.

Monotonicity. Consider a subset $S \subseteq [N]$ and some product $i \in [N] \setminus S$. For each realization of the consumption process, it is easy to verify that, just before each arrival, the most preferred product available under the initial set decision S, is larger or equal to the one under the initial set $S \cup \{i\}$. That is, $\alpha_m(S) \geq \alpha_m(S \cup \{i\})$ for any realization. Thus, the revenue function f is indeed monotone since

$$f(S) = \sum_{m=1}^{\infty} \mathbb{E}\left[r_{\alpha_m(S)}\right] \leq \sum_{m=1}^{\infty} \mathbb{E}\left[r_{\alpha_m(S\cup\{i\})}\right] = f(S \cup \{i\}) \ ,$$

where the above inequality follows from the monotonicity of $(r_i)_{i\in[N]}$.

Submodularity. To prove that f is also submodular, it is sufficient to show that for any subset S and distinct products $i, j \in [n] \setminus S$, the expected revenue function satisfies

$$f(S \cup \{i,j\}) - f(S \cup \{j\}) \leq f(S \cup \{i\}) - f(S) \ .$$

One important observation is that we can assume without loss of generality that $j < i$; otherwise, by permuting i and j we obtain an equivalent inequality. For ease of exposition, we introduce the subset $S_1 = S$, $S_2 = S \cup \{i\}$, $S_1' = S \cup \{j\}$, and $S_2' = S \cup \{i,j\}$. With this notation, the desired inequality is

$$f(S_2') - f(S_1') \leq f(S_2) - f(S_1) \ . \tag{4.11}$$

Note that, if inequality (4.11) is satisfied for any deterministic number of arrivals, it generalizes to the case where M is stochastic. Thus, we restrict our attention to a deterministic number of arrivals M. In this case, the expected revenue increment can be written as

$$f(S_2') - f(S_1') = \sum_{m=1}^{M} \mathbb{E}\left[r_{\alpha_m(S_2')}\right] - \sum_{m=1}^{M} \mathbb{E}\left[r_{\alpha_m(S_1')}\right] \ . \tag{4.12}$$

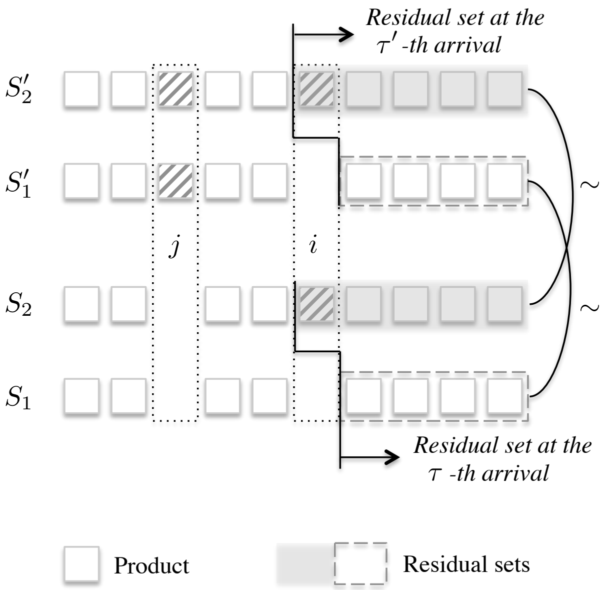
We define τ as the stopping time corresponding to the first arrival where the minimal product available is i, when initially stocking S_2. Namely, we have $\alpha_\tau(S_2) = i$, and $\alpha_{\tau-1}(S_2) < i$ or $\tau = 1$. (In case τ is not defined, the stopping time is set to ∞.) Note that the stopping time τ corresponds to the first arriving customer faced by distinct minimal products, when initially stocking S_1 and S_2. Hence, the revenue difference between these two sets is a function of the arrivals after τ. Formally, since $\alpha_m(S_2) = \alpha_m(S_1)$ with probability 1 for all arrivals $m < \tau$, we infer from an analog of equality (4.12) for S_1 and S_2 that

$$\begin{aligned} f(S_2) - f(S_1) &= \sum_{m=1}^{\tau-1} \mathbb{E}\left[r_{\alpha_m(S_2)} - r_{\alpha_m(S_1)}\right] + \sum_{m=\tau}^{M} \mathbb{E}\left[r_{\alpha_m(S_2)} - r_{\alpha_m(S_1)}\right] \\ &= \sum_{m=\tau}^{M} \mathbb{E}\left[r_{\alpha_m(S_2)} - r_{\alpha_m(S_1)}\right] \\ &= \sum_{k=0}^{M-\tau} \mathbb{E}\left[r_{\alpha_{\tau+k}(S_2)} - r_{\alpha_{\tau+k}(S_1)}\right] \ . \end{aligned} \tag{4.13}$$

This simplification is made intuitive by Figure 4-5. Similarly, by defining τ' as the minimal arrival m such that $\alpha_m(S'_2) = i$, we obtain:

$$f(S'_2) - f(S'_1) = \sum_{k=0}^{M-\tau'} \mathbb{E}\left[r_{\alpha_{\tau'+k}(S'_2)} - r_{\alpha_{\tau'+k}(S'_1)}\right] \ . \tag{4.14}$$

Figure 4-5: Equivalence between the residual sets at the stopping times τ and τ'.



Now observe that, by an inductive argument, it is not difficult to prove that, for any realization of the consumption process, $\tau \leq \tau'$, meaning that $M - \tau \geq M - \tau'$.

As a result, combining inequalities (4.13) and (4.14), we have

$$\begin{aligned}(f(S_2)-f(S_1))-(f(S'_2)-f(S'_1)) \quad = \quad & \sum_{k=0}^{M-\tau'} \mathbb{E}\left[r_{\alpha_{\tau+k}(S_2)} - r_{\alpha_{\tau'+k}(S'_2)}\right] - \sum_{k=0}^{M-\tau'} \mathbb{E}\left[r_{\alpha_{\tau+k}(S_1)} - r_{\alpha_{\tau'+k}(S'_1)}\right] \\ & + \sum_{k=M-\tau'+1}^{M-\tau} \mathbb{E}\left[r_{\alpha_{\tau+k}(S_2)} - r_{\alpha_{\tau+k}(S_1)}\right] \ . \qquad (4.15)\end{aligned}$$

Now, the important observation is that the consumption process starting from the τ-th arrival only depends on the residual set of products available at that time. Hence, we can exploit the equivalence between the residual sets of products, reflected in Figure 4-5, to infer an equivalence between revenues:

- *Equivalence* $\alpha_{\tau'+k}(S'_2) \sim \alpha_{\tau+k}(S_2)$. When initially stocking S'_2, the residual set of products at the τ'-th arrival is the set $(\{i\} \cup S) \cap [i, N]$, which is equal to the residual set at the τ-th arrival when initially stocking S_2. Thus, we infer that the random variables $\alpha_{\tau'+k}(S'_2)$ and $\alpha_{\tau+k}(S_2)$ are identically distributed for all $k \leq M - \tau'$.

- *Equivalence* $\alpha_{\tau'+k}(S'_1) \sim \alpha_{\tau+k}(S_1)$. When initially stocking S'_2, the residual set of products at the τ'-th arrival is the set $S \cap [i, N]$, which is exactly the residual set at the τ-th arrival when initially stocking S_2. Thus, we infer that the random variables $\alpha_{\tau'+k}(S'_1)$ and $\alpha_{\tau+k}(S_1)$ are identically distributed for all $k \leq M - \tau'$.

As a result, equation (4.15) simplifies as follows:

$$(f(S_2)-f(S_1))-(f(S'_2)-f(S'_1)) = \sum_{k=M-\tau'+1}^{M-\tau} \mathbb{E}\left[r_{\alpha_{\tau+k}(S_2)} - r_{\alpha_{\tau+k}(S_1)}\right] \ . \qquad (4.16)$$

Since $S_1 \subseteq S_2$, it is easy to verify that $\alpha_m(S_2) \leq \alpha_m(S_1)$ with probability 1, for any arrival m. Therefore, $r_{\alpha_m(S_2)} \geq r_{\alpha_m(S_1)}$, as our selection step guarantees that $r_1 \geq \cdots \geq r_n$. Combining this observation with inequality (4.16), we infer the desired inequality (4.11), proving that f is indeed submodular.

4.5 Computational Experiments

In this section, we demonstrate the practical merits of our algorithms against existing heuristics. In order to run such comparisons on instances of realistic scale, we take as a benchmark several efficient heuristics proposed in previous related work. Specifically, our algorithms are compared against the following: (i) a discrete-greedy algorithm; (ii) a local search heuristic; and (iii) a gradient-descent algorithm on a continuous extension of the expected revenue function. The latter two heuristics are directly inspired by the work of Mahajan and van Ryzin (2001) and Goyal et al. (2016). For the nested choice model, we have also implemented the enumeration-based algorithm of Goyal et al. (2016).

4.5.1 Heuristics and their implementation

In what follows, we summarize the different algorithms implemented for our computational experiments.

Discrete-greedy. The greedy algorithm starts with zero inventory levels for all products, and iteratively augments the current vector by a single unit of the product that incurs the largest marginal increase in the expected revenue, until reaching C units. As explained toward the end of this section, the expected revenue is evaluated by averaging random realizations of the revenue function, which are sampled by simulating the consumption process and choice behavior of arriving customers.

Local search. Starting from an initial inventory vector, the local search algorithm iteratively improves the expected revenue, by greedily transferring a single inventory unit from a one product to another. Formally, letting $U^{(k)}$ denote the inventory vector obtained at the beginning of step k, a swap is represented by an ordered pair of products (i, j) for which the current inventory level $u_i^{(k)}$ of product i is strictly positive. The inventory vector $U_{i \to j}^{(k)}$ resulting from this swap is derived from $U^{(k)}$ through decreasing $u_i^{(k)}$ by one unit and augmenting $u_j^{(k)}$ by one unit. With this definition, we either proceed to step $k+1$ with the inventory vector $U_{i \to j}^{(k)}$ that maximizes $\mathbb{E}[\mathcal{R}(U_{i \to j}^{(k)})]$

over all swaps (i, j), or terminate the algorithm when none of these swaps improves the expected revenue by at least 0.5%. Once again, the expected revenue function is estimated through sampling, while the initial inventory vector $U^{(1)}$ is defined by stocking C units of the best single-product assortment.

Gradient-descent approach. We consider an adaptation of the stochastic gradient-descent algorithm of Mahajan and van Ryzin (2001). In contrast to their setting, here the revenue function is defined only for integer-valued inventory vectors. Hence, similar to the approach of Goyal et al. (2016), we develop a continuous relaxation of the revenue function, defined through the Lovász extension of a discrete function. Letting $f : \mathbb{Z}^n \to \mathbb{R}$ denote the expected revenue function, its Lovász extension $\hat{f} : \mathbb{R}^n \to \mathbb{R}$ is defined as

$$\hat{f}(U) = f(\lfloor U \rfloor) + \sum_{i=1}^{n} \left(u_{\pi(i)} - u_{\pi(i-1)} \right) \cdot \left[f\left(\lfloor U \rfloor + \sum_{k=1}^{i} e_{\pi(k)} \right) - f\left(\lfloor U \rfloor + \sum_{k=1}^{i-1} e_{\pi(k)} \right) \right] ,$$

where the permutation π sorts products by the increasing fractional part of their inventory, namely, $u_{\pi(1)} - \lfloor u_{\pi(1)} \rfloor \leq \cdots \leq u_{\pi(n)} - \lfloor u_{\pi(n)} \rfloor$. The Lovász extension is piecewise linear, and its gradient can be approximately computed through sampling.

Starting with the initial solution $U^{(0)} = 0$, and letting $U^{(k)}$ denote the solution obtained at the end of step k, each iteration consists of computing $U^{(k+1)} = \max\{0, U^{(k)} + \epsilon_{k+1} \nabla f(U^{(k)})\}$, where ϵ_{k+1} is the step size. When the latter vector does not lie in the feasible region $\{U \in \mathbb{R}^n : \|U\|_1 \leq C\}$, it is projected onto the boundary by linear rescaling. Through extensive trial and error, we chose an adaptive step size of

$$\epsilon_{k+1} = \max \left\{ \frac{C - \|U^{(k)}\|_1}{2}, 0.05 \cdot C \right\} .$$

Intuitively, the step size is larger when the vector $U^{(k)}$ is farther from the boundary, while still enforcing a minimal step size. The algorithm terminates when the objective value does not improve by a factor greater than 0.5%, or after 2000 iterations. Finally, it remains to round the resulting inventory vector to an integral one. Suppose that

$U^{(k+1)}$ is the inventory vector obtained following the gradient-descent algorithm; then $\lfloor U^{(k+1)} \rfloor$ is augmented greedily, by stocking at each step a unit of the product with maximal marginal expected revenue, until reaching C units.

Approximation scheme for the nested choice model. We also implemented the approximation scheme developed by Goyal et al. Goyal et al. (2016) for the nested choice model. This algorithm begins by partitioning products into two classes, frequent and rare, the latter defined as the collection of products for which fewer than $\epsilon^2 C$ units are purchased in expectation when C units of that product are initially stocked. Next, all inventory vectors consisting of $O((1/\epsilon) \cdot \log(1/\epsilon))$ frequent products and a single rare product are approximately enumerated. However, even when we choose $\epsilon = 0.3$, corresponding to a guaranteed approximation ratio of only 0.1^5, this enumeration procedure is impractical and incurs exorbitant running times. In fact, time limits that still enable running the required experiments in practice (within less than an hour) resulted in only partial enumeration, even for the smallest instances considered. Therefore, to improve the observed performance, the enumeration order over inventory vectors is determined such that more expensive products are stocked with higher priority. After considering several different options, this heuristic was superior in balancing between performance and speed.

Revenue evaluation. To approximately evaluate the expected revenue function in each of the above-mentioned heuristics, we use a sample mean estimator over 100 realizations of the consumption process. The number of samples is uniform over all heuristics to provide “fair” comparisons. In contrast, our approximation algorithms, both for the general model and for intervals, make use of the newsvendor-based lower bound, that can be computed exactly and very efficiently.

4.5.2 Instance generation

We fix the number of products at $n = 20$, while the capacity parameter C is varied in $\{20, 40, 80, 150\}$. The prices of these products are generated randomly from a log-normal distribution with location $\mu = 0$ and scale $\sigma = 0.3$. The parameters underlying the consumption process are sampled randomly according to the following generative models.

Choice models. We run three series of experiments:

- *General choice model:* The number of preference lists (in the support of the distribution) is given by $|\mathcal{L}|\in \{50, 160\}$. The preference lists $L_j \in \mathcal{L}$ are generated independently and identically through the following random procedure. We first determine the set of products in L_j through a sequence of Bernoulli trials with $p = 0.35$, i.e., each of the 20 products is independently picked to appear in this list with probability 0.35. Next, for each list L_j, the ranking order over product alternatives in L_j is sampled uniformly over all $|L_j|!$ permutations.

- *Intervals choice model:* Similarly, we vary the number of interval preference lists $|\mathcal{L}|$ in $\{50, 200\}$. To create $\mathcal{L}$, we sample uniformly at random over all subsets formed by $|\mathcal{L}|$ distinct intervals (out of $n(n + 1)/2$ intervals overall).

- *Nested choice model:* Here, we pick $|\mathcal{L}|= n$, meaning that every nested list is included in $\mathcal{L}$.

In each of the above models, the probability distribution over $\mathcal{L}$ is sampled uniformly over the unit simplex.

Number of customers $\boldsymbol{M}$. We test several parametric distributions for the demand M. Specifically, letting $G \sim N(30, 40)$ and $P \sim \exp(0.02)$, we alternatively examine $M = \min\{[\lceil G\rceil|G \geq 0], 100\}$ and $M = \min\{\lceil P\rceil, 100\}$. For the nested choice model, we also test a nonparametric non-IFR demand distribution, where the support of M is constructed by randomly sampling 5 integers in the interval $[1, 100]$ (without replacement), and the corresponding probability measure is generated uniformly at random from the unit simplex.

4.5.3 Results

We implemented our algorithms, as well as the heuristics described in Section 4.5.1, using the Julia programming language. The experiments described in this section were conducted on a standard laptop with 2.7GHz Intel Core i5 processor and 8GB of RAM. In order to execute the required experiments within several days, we imposed a time limit of 10 minutes (per-instance) for all heuristics tested, noting that each of our algorithms terminates within a few seconds.

Relative performance. For each instance tested, in practice one cannot simply compute the optimal expected revenue through brute-force enumeration, as the latter involves considering all combinations of feasible inventory vectors, number of customers, and their choice preferences. Similarly, we found integer programming approaches to be impractical; for example, using a sample average approximation with 100 realizations of the consumption process, the running time exceeds 30 minutes for the smallest instance tested. Therefore, rather than estimating the exact optimality gaps, we compare the different algorithms on a relative basis. Specifically, for each instance, the benchmark is set as the expected revenue of the most profitable inventory vector obtained through all algorithms tested. Then, the relative performance of each algorithm (reported subsequently) is defined as the ratio between its expected revenue and the benchmark. For example, if our algorithm attains an expected revenue of 1, while all tested heuristics generate an expected revenue of 0.9, the relative performance is 100% for our algorithm, and 90% for the other heuristics.

Performance analysis. The results of our experiments are summarized in Tables 1, 2, and 3, where each entry is obtained by averaging over 30 random instances. Our algorithms dominate the other heuristics revenue-wise in all configurations. Under a general generative model, the discrete-greedy algorithm emerges as the most effective heuristic on average, but it still falls behind our general approximation by 7% to 36%. There is a single configuration where the gradient-descent algorithm outperforms the other heuristics, specifically when $C = 80$. Interesting, in the latter regime, the inventory capacity approximately matches the demand in expectation. Since this observation is consistent across all choice models and demand distributions, our experiments seem to indicate that the gradient-descent is particularly effective in such a regime (while still falling short of the algorithms we propose).

Table 1: Results under the general choice model (|*L*|=160)

parameters		**avg. performance (%)**				**avg. running time (sec.)**			
M	*C*	GA	DG	GD	LS	GA	DG	GD	LS
Gaussian	20	100	87.6	66.5	77.4	6.6	3.1	55.9	7.9
	40	100	70.6	77.2	62.1	7.1	6.4	216.2	16.9
	80	100	64.1	78.9	54.4	7.3	13	374.8	21.4
	150	100	66.8	62.2	50.6	8.1	24.3	384.2	19.7
Poisson	20	99.9	93.1	66.4	91.4	6.9	3.7	36.6	10.7
	40	100	77.3	80.7	65.6	6.9	7.4	238.8	17.1
	80	100	72.6	86.5	57.6	7.6	15.2	435.3	27.5
	150	100	71	67.8	53.4	8.3	28.4	442	20.4

Here, GA designates our general approximation algorithm, DG is the discrete-greedy algorithm, GD is the gradient-descent approach, and LS corresponds to the local search heuristic.

Surprisingly, even in the more specialized settings, our general-purpose approximation still enjoys strong practical performance. Specifically, its performance is comparable to the decomposition algorithm of Section 4.3 in the intervals case. Under the nested model, the selective-greedy algorithm developed in Section 4.4 improves revenue by 1% on average against our general approximation. It is worth mentioning that, as one would expect, the discrete-greedy and selective-greedy algorithms have identical performance under the nested model. However, our selection step restricts the incremental actions examined upon each iteration, as opposed to considering all possible stocking decisions at each iteration, and therefore reduces the running time significantly.

On the computational front, our algorithms tend to outperform the other heuristics while the gradient-descent algorithm is the slowest one. In contrast to other algorithms, the running time of the general approximation grows sub-linearly in the capacity value; hence, when $C = 150$, its running time is better by a factor of 3 to 40 against other heuristics. Under the nested choice model, the selective-greedy al-

Table 2: Results under the intervals choice model ($|L|$=160)

parameters		**avg. performance (%)**					**avg. running time (sec.)**				
M	C	GA	IN	DG	GD	LS	GA	IN	DG	GD	LS
Gaussian	20	99.2	99.4	89.6	52.7	67.4	5.6	6.8	2.0	33.7	5.3
	40	99.2	99.0	62.1	60.6	25.6	6.0	10.7	4.0	136.2	8.8
	80	97.5	98.4	49.1	76.3	16.6	6.2	11.7	8.1	229.5	19.6
	150	99.5	97.7	48.3	27.6	15.7	6.7	7.1	15.1	244.8	13.5
Poisson	20	99.7	99.6	97.3	52.7	93.3	8.1	10.8	3.6	23.3	9.3
	40	99.4	99.2	76.8	63.4	37.2	8.5	9.0	7.3	232.7	16.6
	80	99.0	99.1	57.2	79.7	17.7	9.0	10.8	14.5	399.4	15
	150	98.4	99.3	57.3	37.1	18	9.9	10.4	27.2	430.1	51.9

Here, GA designates our general approximation algorithm, IN is our approximation for the intervals choice model, DG is the discrete-greedy algorithm, GD is the gradient-descent approach, and LS corresponds to the local search heuristic.

gorithm is an order of magnitude faster than any other algorithm for small instances ($C \leq 80$). In this setting, the approximation scheme of Goyal et al. (2016) is particularly inefficient, and reaches the time limit of 10 minutes prior to completing its full enumeration procedure. As a result, the revenue performance is not near-optimal, particularly for large instances ($C = 150$), where the average optimality gap can be as large as 18%.

4.6 Concluding Remarks

Robustness under a mixture of nested models. Given the very structured nature of nested preference lists, it is interesting to investigate whether our algorithms can be utilized in the case of a model misspecification, specifically, under a mixture of nested instances. Here, each segment of the mixture is uniquely described by its left endpoint product; this setting is a special case of the intervals choice model. It is not difficult to verify that, when the mixture is formed by K customer segments, we can derive an $O(\log K)$-approximation by adapting the recursive decomposition of Section 4.3.2. Specifically, at every step of the recursion, each segment of the partition is broken at the median left endpoint product of all remaining lists contained in that

Table 3: Results under the nested choice model ($|\boldsymbol{L}|$=20)

parameters		**avg. performance (%)**						**avg. running time (sec.)**					
M	C	GA	SG	GLS	DG	GD	LS	GA	SG	GLS	DG	GD	LS
Gaussian	20	96.4	99.8	93	99.8	78	99.7	0.8	0.0^*	600	0.1	1.4	0.3
	40	99.5	99.8	89.2	99.8	92.9	83.9	0.9	0.0^*	600	0.3	2	0.9
	80	99.9	99.7	91.6	99.7	99.3	69.9	0.9	0.3	600	1.3	3.8	0.9
	150	100	100	85.2	100	96.5	69.7	1	0.9	600	3.9	5.7	12.2
Poisson	20	95.6	99.4	92.3	99.4	83.2	99.9	0.8	0.0^*	600	0.0^*	1.5	0.3
	40	98.5	99.5	91.1	99.5	91.7	97.1	0.9	0.0^*	600	0.3	2.9	1
	80	99.5	99.7	89.8	99.7	99.4	83.9	0.9	0.3	600	1.3	4	2.6
	150	100	100	82	100	99.3	71.2	0.9	1	571	3.8	5	12.8
Non-IFR	20	97.1	99.2	92.1	99.2	79.8	99.5	0.8	0.0^*	600	0.0^*	1.5	0.3
	40	95.9	99	91.3	99	89.3	95.6	0.8	0.0^*	600	0.3	2.5	1.1
	80	99.9	98.9	89.8	98.9	97.7	84.4	0.9	0.2	600	1.2	4.8	1.7
	150	100	100	86.4	100	96.7	72.3	0.9	1.2	561	3.7	5	18.6

Here, GA designates our general approximation algorithm, SG is our selective-greedy approximation for the nested choice model, DG is the discrete-greedy algorithm, GD is the gradient-descent approach, LS corresponds to the local search heuristic, and GLS is the enumeration-based algorithm of Goyal et al. (2016). 0.0^**: running time is smaller than* 0.01 *sec.*

segment. As a result, the recursion depth is $O(\log K)$, thus leading to the abovementioned approximation ratio.

Newsvendor-like models. The problem formulation considered in this chapter incorporates a hard capacity constraint on the number of units stocked. A natural direction for future research is to study newsvendor-like models, where there is no capacity limitation, and instead, the salvage value of inventory has decreasing marginal gains. It would be interesting to investigate whether the technical ideas we developed can be leveraged to this setting.

Refined approximability results. Following the present work, one open question is that of determining whether the intervals model can be efficiently approximated within a constant factor. Although we were able to obtain a performance guarantee of $O(\log\log(P_{\max}/P_{\min}))$, even the simpler problem of evaluating the expected revenue of a given inventory vector in this model is still wide open. An interesting direction to consider would be to propose structural transformations, in the spirit of Section 4.4, in order to reveal certain submodularity properties. It is worth mentioning that, even with identical prices, the original revenue function is not submodular, as we demonstrate in Lemma B.1.4 (see Appendix B.1.3).

Chapter 5

Joint Assortment and Inventory Decisions: Multinomial Logit Model

5.1 Introduction

The Multinomial Logit model (MNL) has gained widespread popularity among practitioners, since it can be estimated efficiently, even from limited data (Ford 1957, McFadden 1973), and it yields tractable assortment optimization formulations (Talluri and van Ryzin 2004, Rusmevichientong et al. 2010, 2014). However, in practice, the assortment and inventory decision levers are inherently connected. Due to the proliferation of products, stock-out events are pervasive in modern-day retailing [1], and consequently, firms need to manage their inventory supply to ensure the availability of assortment products to their end customers.

Similar to Chapter 4, we study the joint assortment optimization and inventory management problem, referred to as *dynamic assortment planning*. In this chapter, we assume that the customers' dynamic substitution behavior is described by the Multinomial Logit choice model. Despite the centrality of the MNL model in revenue management literature, obtaining efficient algorithms with analytical guarantees for MNL-based dynamic assortment planning models is a long-standing open question, since the seminal work of Mahajan and van Ryzin (2001). In fact, even the efficient

[1] See IHL (2015).

evaluation of the expected revenue generated by given assortment and inventory decisions is a challenging computational problem under this choice model.

Model description. A formal description of the MNL-based dynamic assortment planning model is given in Section 5.1.2. The ingredients of this optimization model are identical to Chapter 4. To wit, the firm makes assortment and inventory decisions at the beginning of the selling-season, under a cardinality constraint on the total number of units stocked. The consumption process is modeled through a random sequence of arriving customers, each having random preferences over the products on stock upon arrival. From a computational standpoint, in addition to the underlying choice model, the problem formulation hinges on describing the distribution of the number of customer arrivals, named the *demand* hereafter.

5.1.1 Results and techniques

The main contribution of this chapter is to devise the first provably-good approximation algorithm for dynamic assortment planning under the Multinomial Logit choice model. Specifically, our approach guarantees a constant-factor worst-case approximation for a broad class of demand distributions (standing for the total number of arriving customers) commonly used in operations management, that satisfy the *increasing failure rate* (IFR) property. Moreover, we show that this algorithm has a superior empirical performance in comparison to existing heuristics on synthetic instances. Against existing state-of-the-art methods, our algorithm leads to substantial gains in the expected revenue, ranging from 6% to 16%, with better computational efficiency and robustness. Our algorithmic approach relies on a combination of greedy procedures, where stocking decisions are restricted to specific assortments, and the objective function takes a modified form. Our theoretical analysis along with the experimental results provide evidence that such restrictive policies could in fact be more effective than general-purpose methods, that consider stocking decisions across *all* products. Along the way, we develop a number of novel technical ideas that could very well contribute to studying additional combinatorial optimization problems and to assortment planning methodologies in particular.

Restricted-submodular maximization. At the core of our analysis, we develop new concepts of submodularity and monotonicity, called the *restricted-submodular* and *restricted-non-decreasing* properties, that are weaker than their standard counterparts. Specifically, when optimizing certain set functions under a cardinality constraint, the objective function could generally violate the submodularity property, while still having a submodular-like behavior within the feasible collection of sets, i.e., those satisfying the cardinality constraint. Thus, in the restricted-submodular setting, the structural inequalities defining submodularity and monotonicity are not required uniformly over all sets, and instead, we restrict attention to the feasible region only. We show that the classic analysis of greedy algorithms extends to this broader setting, and obtain a $(0.318 - \epsilon)$-approximation. Moreover, this worst-case guarantee holds with high probability even when the greedy procedure is given access to noisy estimates of the objective function at each step.

Algorithmic approach and performance guarantees. For ease of presentation, we describe our approach in an incremental way, where a simplified setting is first examined, prior to addressing the most general case, thereby establishing the following worst-case guarantees.

- *Core algorithm with evaluation oracle.* As previously mentioned, in dynamic substitution models, it is generally unknown how to efficiently compute the expected revenue generated by given initial inventory levels. To bypass this difficulty, we first operate under the *efficient oracle assumption*, where we temporarily assume that the expected revenue function can be efficiently evaluated with high probability by some (unspecified) oracle procedure. Under this assumption, we devise in Section 5.3 a polynomial-time algorithm with a constant-factor worst-case guarantee, for any demand distribution with increasing failure rates. Specifically, for any error parameters $\epsilon \in (0, 1/4)$ and $\delta > 0$, our randomized algorithm attains a $(0.139 - \epsilon)$-approximation with probability at least

$1-\delta$. Moreover, our methods are amenable to tighter analysis under more restrictive settings, allowing us to obtain an approximation guarantee of $0.179-\epsilon$ under the plausible assumption that the number of inventory units (or capacity) exceeds the number of products, and $0.632-\epsilon$ when only products within an optimal assortment (in the standard sense) can be stocked. The latter result holds for general demand distributions.

Technically speaking, for large enough capacity values, the algorithm concurrently runs two greedy procedures: each restricts attention to a specific class of products, and the inventory levels are chosen greedily over the residual set of products, using a modified objective function. Our analysis relies in large part on the restricted-submodular and restricted-non-decreasing properties mentioned earlier. Indeed, after we interpret one residual problem in terms of optimizing a set function, we show that, while the latter generally violates the standard properties of submodularity and monotonicity, it still satisfies their weaker (restricted) version. The proof is based on novel probabilistic coupling ideas, allowing us to compare the dynamic substitution patterns driven by the MNL model. A particularly interesting byproduct of our analysis is showing that a commonplace heuristic, which stocks the optimal assortment (in the standard sense) and scales inventory proportional to the expected sales, has a provable performance guarantee with respect to a restricted class of products.

- *General approximation algorithm.* In Appendix C.1, we bypass the efficient oracle assumption, and derive a general constant-factor approximation for dynamic assortment planning under the MNL choice model, with increasing failure rate demand distributions. For any $\epsilon \in (0, 1/4)$ and $\delta > 0$, we devise a randomized polynomial-time algorithm attaining a worst-case guarantee of $0.122-\epsilon$ with probability at least $1-\delta$, which is improved to $0.151-\epsilon$ when the capacity exceeds the number of products. Here, the key observation is that only a specific class of products, named *heavy* products, presents an optimization challenge, while the other *light* products are in fact easier to approximate, within any degree of accuracy, using a standard greedy procedure. On the other hand, the expected revenue due to heavy products can be efficiently evaluated using a sampling-based estimator, allowing us to apply the oracle-based algorithm described earlier. As explained in Appendix C.1, the notions of light and heavy relate to the preference weight of the respective products.

Empirical evaluation. While our theoretical worst-case guarantees might look unsatisfactory for practical purposes, we present in Section 5.4 extensive computational experiments, showing that the resulting algorithm largely outperforms existing heuristics in terms of performance and speed. These experiments employ our algorithm on randomly-generated instances, concurrently to the following heuristics: (i) a local-search heuristic based on greedily exchanging units between pairs of products, similar to Goyal et al. (2016); (ii) a gradient-descent approach based on a continuous extension of the revenue function, similar in spirit to the work of Mahajan and van Ryzin (2001); (iii) exact dynamic programming for two variants of the problem formulated by Topaloglu (2013), based on a Poisson and a normal approximation of the demand process; (iv) the deterministic relaxation heuristic proposed by Honhon et al. (2010), implemented using a commercial integer programming solver; (v) a discrete-greedy algorithm, where in each step a single unit is added to the product with the largest marginal expected revenue. Against these benchmarks, our algorithm attains expected revenues that are better by a factor ranging between 6% and 16%, and simultaneously dominates all methods in 66% of the instances tested. We also report that the proportional scaling heuristic, used as a subroutine in Section 5.3.3, is outperformed by the overall algorithm on average by 2.4%. In addition, the running time of our algorithm is significantly shorter than the above-mentioned heuristics, at the exception of the normal-based dynamic program.

5.1.2 Problem formulation

We are given n products, where each product $i \in [n]$ is associated with a preference weight w_i and a per-unit selling price r_i. In addition, there is a capacity bound of

C on the total number of units to be stocked. In the dynamic assortment planning problem, the retailer has to jointly decide on an assortment, i.e., a subset of products to be offered, as well as on the initial inventory levels of these products, which are not replenished later on. That is, a feasible solution specifies the initial inventory levels of all products, represented by an integer-valued vector $U = (u_1, \dots, u_n)$ that meets the capacity constraint, $\sum_{i=1}^{n} u_i \leq C$.

Stochastic MNL-based consumption process. We proceed by providing the additional model ingredients that describe the process according to which customers arrive and purchase products over time. A random number of customers M arrive sequentially, where the distribution of M is known to the decision-maker. Upon the arrival of a customer, suppose that $S \subseteq [n]$ is the subset of products that are currently available, due to being initially stocked, and not depleted until now. Then, this customer either:

- Picks a random product out of S and purchases a single unit, where the probability for choosing product $i \in S$ is $w_i/(1 + w(S))$. Here, $w(S)$ stands for the total weight of the products in S, i.e., $w(S) = \sum_{j \in S} w_j$.
- Leaves without purchasing any product, which happens with probability $1/(1 + w(S))$.

Objective function. When the sequence of customer arrivals ends, we use $\mathcal{R}(U)$ to denote the revenue resulting from an initial inventory vector U. This revenue is clearly random, due to the stochasticity in the number of customers and in their choice of products to purchase. The objective is to compute a feasible inventory vector, so that the expected revenue is maximized,

$$\max_{(u_1,\dots,u_n) \in \mathbb{Z}_+^n} \left\{ \mathbb{E}\left[\mathcal{R}(u_1, \dots, u_n)\right] : \sum_{i=1}^{n} u_i \leq C \right\} .$$

The IFR assumption. As mentioned in Section 5.1.1, the distribution of the number of customers M is assumed to have an increasing failure rate (IFR), meaning that the sequence $\Pr[M = k] / \Pr[M \geq k]$ is non-decreasing over the integer domain. For definitions of stochastic orders and stochastic monotonicity, we refer the reader to Shaked and Shanthikumar (1994). It is worth mentioning that the IFR property is satisfied by many distributions considered in operations management applications, including Normal, Exponential, Geometric, Poisson, and Beta (for certain parameters).

Remark: Static formulation. Recall that the *static* case corresponds to the situation where there is a single customer arrival. This setting is equivalent to relaxing the capacity constraint (i.e., $C = \infty$), and the problem reduces to the standard assortment optimization formulation. Indeed, since there are no stock-out events, we can always offer the optimal assortment to each arriving customer. The assortment computed in this setting is referred to as the *optimal static assortment*.

5.1.3 Related literature

The MNL choice model. The Multinomial Logit (MNL) model is arguably the most widespread approach for modeling choice among practitioners, as reflected by seminal studies in transportation (McFadden 1980, Ben-Akiva and Lerman 1985) and marketing (Guadagni and Little 1983, Grover and Vriens 2006, Chandukala et al. 2008). This model, proposed independently by Luce (1959) and Plackett (1975), is grounded in economic theory of utility maximization, and describes the probabilistic choice outcomes of a representative agent who maximizes his utility over different alternatives, through a noisy evaluation of the utility they procure. The popularity of this model was notably driven by its simple estimation procedures (McFadden 1973, Talluri and van Ryzin 2004, Maystre and Grossglauser 2015), even with limited data (Ford 1957, Negahban et al. 2012), as well as by its computational tractability in decision-making problems. Indeed, the standard assortment optimization problem is well-understood in the context of MNL choice preferences. For the uncapacitated variant, where any number of products can be offered, Talluri and van Ryzin (2004) showed that the optimal assortment consists of the k-highest price products, for some k, and can therefore be computed efficiently. Rusmevichientong et al. (2010) devised a polynomial-time algorithm for the capacitated variant, where an upper bound is imposed on the number of products offered. These results were further advanced to handle more general settings (Rusmevichientong and Topaloglu 2012, Rusmevichientong et al. 2014), including a linear programming approach proposed by Davis et al. (2013) and a local-ratio framework developed by Désir et al. (2015).

Challenges in dynamic assortment planning. Under multiple stochastic arrivals, the problem we study becomes considerably more challenging than its 'static' counterpart (single-arrival model), due to the additional 'dynamic' aspect. Indeed, the assortment is altered along the sequence of arrivals due to stock-out events, as customers purchase the most preferred product available according to a probabilistic choice model. Therefore, the substitution behavior of customers depends on each sample-path realization, and a large number of samples is generally needed to obtain accurate estimates of the expected revenue function. In addition, this function violates several well-behaved properties. For instance, under a general model of choice, for a continuous relaxation of the dynamic assortment problem, Mahajan and van Ryzin (2001) showed that the revenue function is not even quasiconcave. We demonstrated in Chapter 4, through various counter-examples, that this function (in modified form) is not submodular, even for very simple choice modeling approaches.

Existing methods. As a result, most of the work we are aware of in the context of dynamic assortment planning develops heuristics based on continuous relaxations and probabilistic assumptions (Smith and Agrawal 2000, Mahajan and van Ryzin 2001, Gaur and Honhon 2006, Nagarajan and Rajagopalan 2008, Honhon et al. 2010, Honhon and Seshadri 2013). These approaches either give rise to exponential-time algorithms, apply to more stylized models, or converge to local optima, such as the gradient-descent method proposed by Mahajan and van Ryzin (2001).

Similar to the present setting, Topaloglu (2013) studied a joint assortment and inventory management model with sequential customer arrivals, MNL preferences, and exogenous per-unit costs rather than capacities. This model was shown to admit an efficient approximate dynamic programming formulation, based on strong separability properties. However, this setting has several restrictions: the dynamic substitution effects are overlooked and the demand follows a Poisson process, while on the other hand, the retailer is allowed to utilize a mixed assortment strategy. Goyal et al. (2016) and Segev (2015) considered dynamic assortment planning models, with a fully stochastic consumption process, for which they devised polynomial-time algorithms with provable approximation guarantees. However, the choice models considered in these papers have simple combinatorial structures, that impose a very specific order by which products are consumed and depleted. This property is crucial to the design of low-dimensional dynamic programs for revenue evaluation and optimization. In contrast, the choice outcomes described by the MNL model do not impose any particular (deterministic) pattern on stock-out events. Consequently, dynamic optimization in this context appears to be significantly more challenging.

5.2 Preliminaries

In what follows, we establish a number of technical results that were briefly discussed in Section 5.1.1. These are instrumental for our algorithmic approach and its analysis.

5.2.1 Extensions of submodular maximization

The crux of our algorithm resides in exploiting new notions of submodularity and monotonicity, respectively termed as *restricted submodularity* and *restricted monotonicity*. Intuitively, these properties require that the structural inequalities defining submodularity and monotonicity are satisfied as long as the sets involved are within the feasible region, formed by a cardinality constraint. Although weaker than the standard notions, we show that these properties are sufficient for the design of constant worst-case approximations, even with noisy estimates of the objective function.

Restricted submodularity and monotonicity. We begin by defining the notion of restricted submodularity. A set function $f : 2^{[n]} \to \mathbb{R}$ is said to be restricted-s-submodular for some integer $s \in \mathbb{N}$ if, for any subset $S \subseteq [n]$ of cardinality at most $s - 2$ and elements $i \neq j \in [n] \setminus S$, we have

$$f(S \cup \{i, j\}) - f(S \cup \{j\}) \leq f(S \cup \{i\}) - f(S) \ .$$

By a similar extension of conventional definitions, we say that a set function f is restricted-s-non-decreasing if $f(S) \leq f(T)$ for any pair of subsets $S \subseteq T$ of cardinality at most s. In what follows, the parameter s is always equal to the capacity C, and therefore, we simply say that a set function is *restricted-submodular* or *restricted-non-decreasing.*

The efficient oracle assumption. A particularly useful extension of the standard submodular maximization setting is to assume that the objective function f cannot be evaluated exactly in an efficient way, and instead, we are given access to a noisy estimation oracle. Formally, the efficient oracle assumption states that, for any error parameter $\epsilon > 0$ and for any confidence level $\delta > 0$, there exists an efficient procedure that, given any subset $S \subseteq [n]$, computes a random estimate $\tilde{f}(S)$ of $f(S)$ such that

$$\Pr\left[(1 - \epsilon) \cdot f(S) \leq \tilde{f}(S) \leq (1 + \epsilon) \cdot f(S)\right] \geq 1 - \delta \ .$$

The running time of this procedure is assumed to be polynomial in the input size, $1/\epsilon$, and $1/\delta$.

By leveraging classic techniques for approximately maximizing monotone submodular functions (see, e.g., Nemhauser et al. (1978)), we derive a constant-factor approximation for non-negative restricted-non-decreasing and restricted-submodular functions, as stated in the following claim.

Lemma 5.2.1. *Under the efficient oracle assumption, for any $\epsilon \in (0, 1/4)$ and $\delta > 0$, the problem of maximizing a non-negative restricted-non-decreasing and restricted-submodular set function under a cardinality constraint can be approximated within* *factor* $0.318 - \epsilon$*, with probability at least* $1 - \delta$*. The running time of our algorithm is polynomial in the input size,* $n^{1/\epsilon}$*, and* $1/\delta$*.*

To avoid deviating from our general theme, we present the proof in Appendix C.3.1, and only provide a high-level description here. For a small cardinality parameter ($C < 1/\epsilon$), we simply enumerate over all possible subsets. Otherwise, the algorithmic idea behind Lemma 5.2.1 is a standard greedy procedure: starting with an empty set, we add at each step the element that guarantees the largest marginal increase in the objective function. To account for the boundary effects in the restricted-submodular setting, the traditional analysis of this algorithm needs to be refined with suitable lower bounds on the marginal increase at each iterative step. In addition, we control for the accumulated estimation error during these iterations, due to the noisy oracle. For this purpose, the oracle is executed with an appropriate choice for the error parameters, $\epsilon = \epsilon(n, C)$ and $\delta = \delta(n, C)$, while ensuring that its running time remains polynomial.

5.2.2 Subadditivity of the expected revenue function

The next lemma, whose proof appears in Appendix C.3.2, asserts that the expected revenue function in the MNL-based dynamic assortment model is subadditive.

Lemma 5.2.2 (Subbaditivity)**.** *For any inventory vectors* U_1 *and* U_2*, we have* $\mathbb{E}[\mathcal{R}(U_1 + U_2)] \leq \mathbb{E}[\mathcal{R}(U_1)] + \mathbb{E}[\mathcal{R}(U_2)]$*.*

To better understand the implications of this claim, let U^* be an optimal inventory vector. For any subset of products $S \subseteq [n]$, we use U^*_S to designate the projection of U^* on S, i.e., U^*_S is the vector obtained from U^* by setting the inventory levels of all products in $[n] \setminus S$ to zero. Now suppose that the collection of products $[n]$ is partitioned into the subsets $\mathcal{S}_1, \ldots, \mathcal{S}_K$. Consequently, since $U^* = \sum_{k \in [K]} U^*_{\mathcal{S}_k}$, and the expected revenue function is subadditive, it follows that

$$\sum_{k \in [K]} \mathbb{E}\left[\mathcal{R}(U^*_{\mathcal{S}_k})\right] \geq \mathbb{E}[\mathcal{R}(U^*)] \ . \tag{5.1}$$

From an algorithmic perspective, this bound can be utilized by treating each subset $\mathcal{S}_k$ as a separate subproblem for which a tailor-made algorithm is developed. Now, suppose we obtain a γ_k-approximation for each subproblem, i.e., an inventory vector $\tilde{U}_{\mathcal{S}_k}$ satisfying $\mathbb{E}[\mathcal{R}(\tilde{U}_{\mathcal{S}_k})] \geq \gamma_k \cdot \mathbb{E}\left[\mathcal{R}(U^*_{\mathcal{S}_k})\right]$. By picking the best solution (revenue-wise) out of the K resulting inventory vectors, for any $\alpha_1, \ldots, \alpha_K \geq 1$ with $\sum_{k \in [K]} \alpha_k = 1$, we obtain an expected revenue of

$$\max_{k \in [K]} \mathbb{E}\left[\mathcal{R}(\tilde{U}_{\mathcal{S}_k})\right] \geq \sum_{k \in [K]} \alpha_k \cdot \mathbb{E}\left[\mathcal{R}(\tilde{U}_{\mathcal{S}_k})\right] \geq \left(\min_{k \in [K]} \alpha_k \gamma_k\right) \cdot \sum_{k \in [K]} \mathbb{E}\left[\mathcal{R}(U^*_{\mathcal{S}_k})\right] \geq \left(\min_{k \in [K]} \alpha_k \gamma_k\right) \cdot \mathbb{E}\left[\mathcal{R}(U^*)\right] ,$$

where the last inequality holds by (5.1). As a result, we have just obtained an approximation ratio of $\min_{k \in [K]} \alpha_k \gamma_k$ for the original problem, which can be optimized by picking the best convex combination $\alpha_1, \ldots, \alpha_K$. This decomposition idea is exploited in Section 5.3 and in Appendix C.1.

5.3 Core Algorithm with Evaluation Oracle

In this section, we devise an efficient algorithm with a constant-factor worst-case guarantee, under IFR demand distributions. Since revenue evaluation is challenging by itself in the dynamic setting, we temporarily operate under the efficient oracle assumption described in Section 5.2.1. Specifically, we assume in the remainder of this section that, for any error parameter $\epsilon > 0$ and confidence level $\delta > 0$, there is an efficient procedure to estimate the expected revenue $\mathbb{E}[\mathcal{R}(U)]$ of any inventory vector U up to a multiplicative factor of $1 \pm \epsilon$, with probability at least $1 - \delta$.

In Appendix C.1, we explain how this assumption can be bypassed, losing a small constant factor in optimality, while utilizing the conditional approach developed here as a subroutine. The latter bears practical significance by itself, since simulation-based methods or surrogate models are commonly used to go around the computational difficulties of evaluating certain objective functions.

Theorem 5.3.1. *Under the efficient oracle assumption, for any $\epsilon \in (0, 1/4)$ and $\delta > 0$, the dynamic assortment planning problem under the Multinomial Logit choice* *model and IFR demand distribution can be approximated within a factor of* $0.139 - \epsilon$ *with probability at least* $1 - \delta$*, in time polynomial in the input size,* $n^{1/\epsilon}$*, and* $1/\delta$*. When* $C \geq n$*, this factor can be improved to* $0.179 - \epsilon$*.*

5.3.1 Overview of the algorithm

Preliminary step: price threshold. We begin by computing $\mathrm{OPT}_{\mathrm{static}}$, the optimal capacitated static revenue. There are several well-known polynomial-time algorithms (Megiddo 1979, Rusmevichientong et al. 2010, Davis et al. 2013) to solve the capacitated assortment optimization problem (with a single representative customer), still with an upper bound of C on the number of products to be stocked (rather than units). Hereafter, the corresponding optimal static assortment (which generates an expected revenue of $\mathrm{OPT}_{\mathrm{static}}$) is denoted by $\mathcal{A}^*$. Subsequently, we use $\mathrm{OPT}_{\mathrm{static}}$ as a price threshold to distinguish between *expensive* products, with price greater or equal to $\mathrm{OPT}_{\mathrm{static}}$, and *cheap* products, whose price is smaller than $\mathrm{OPT}_{\mathrm{static}}$. We let $\mathcal{E}$ and $\mathcal{C}$ designate the subsets of expensive and cheap products, respectively, thus forming a partition of the products $[n]$. In the sequel, our algorithm constructs inventory vectors that are exclusively composed of expensive products, whereas the optimal inventory vector could stock both cheap and expensive products.

Decomposition approach. Next, we utilize the decomposition idea described in Section 5.2.2. Specifically, we pick the most profitable among two candidate inventory vectors, denoted by $U_{\mathcal{E}}$ and $U_{\mathcal{C}}$. While both stocking only expensive products, these vectors are constructed to fulfill different purposes: $U_{\mathcal{E}}$ competes against the contributions of expensive products in the optimal expected revenue, while $U_{\mathcal{C}}$ competes against the revenue contributions of cheap products. By 'compete', we mean that $U_{\mathcal{E}}$ is guaranteed to generate a constant fraction of the expected revenue due to selling expensive products in the optimal solution, and an analogous property holds for $U_{\mathcal{C}}$ with respect to the cheap products. From this point on, we let U^* be a fixed optimal inventory vector, and recall that $U^*_{\mathcal{E}}$ designates the projection of U^* on the set of expensive products $\mathcal{E}$, i.e., the vector obtained from U^* by setting the inventory levels of all cheap products to zero. The vector $U^*_{\mathcal{C}}$ is defined in an analogous way. Our analysis relies on comparing the expected revenue of $U_{\mathcal{E}}$ and $U_{\mathcal{C}}$ with that of $U^*_{\mathcal{E}}$ and $U^*_{\mathcal{C}}$, respectively.

Competing against expensive products (Section 5.3.2). Since our analysis of the greedy algorithm for restricted-non-decreasing and restricted-submodular functions (see Section 5.2.1) results in an extra additive error that depends on $1/C$, we distinguish between two cases in order to construct $U_{\mathcal{E}}$. Specifically, when $C \geq 1/\epsilon$, the inventory levels of expensive products are determined by a greedy approach, where at each step a single unit of the product that generates the largest marginal increase in the expected revenue is picked until stocking C units. In this case, the above-mentioned additive error affects the multiplicative factor we obtain by a factor of only $O(\epsilon)$. In the opposite case, when $C < 1/\epsilon$, we resort to enumeration over all $O(n^{1/\epsilon})$ feasible inventory vectors. To analyze this approach, we prove that the restricted-non-decreasing and restricted-submodular properties are satisfied by the revenue function (in modified form), for the problem restricted to the collection of expensive products $\mathcal{E}$. By executing the evaluation oracle with the appropriate error and confidence parameters described in Lemma 5.2.1, it follows that the inventory vector $U_{\mathcal{E}}$ competes against the optimal expected revenue obtained from expensive products. Specifically, with probability at least $1-\delta$,

$$\mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{E}}\right)\right] \geq (0.318-\epsilon) \cdot \mathbb{E}\left[\mathcal{R}\left(U^*_{\mathcal{E}}\right)\right] \ . \tag{5.2}$$

When $C \geq n$, we obtain an improved approximation ratio of $(1-\epsilon) \cdot (1-1/e)$.

Competing against cheap products (Section 5.3.3). We compete against $U^*_{\mathcal{C}}$ by stocking expensive products, rather than cheap products. Specifically, $U_{\mathcal{C}}$ is computed through a greedy procedure, where stocking decisions are restricted to the optimal static assortment $\mathcal{A}^*$, and the expected revenue function is replaced by a simplified objective function. This alternative objective is formed by neglecting the revenue generated by stock-out substitution, namely, assuming that customers do not substitute to less preferred options once their most preferred product is depleted. The lower bound thus obtained can be interpreted as the objective function of a multi-item newsvendor problem, that can be optimized greedily. By exploiting the IFR property, we show that $U_{\mathcal{C}}$ guarantees at least 1/4 of the optimal expected revenue due to cheap products, i.e.,

$$\mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{C}}\right)\right] \geq \frac{1}{4} \cdot \mathbb{E}[\mathcal{R}\left(U_{\mathcal{C}}^{*}\right)] \ . \tag{5.3}$$

Concluding the proof of Theorem 5.3.1. Before providing additional details on the above-mentioned algorithms and their respective performance, we argue that inequalities (5.2) and (5.3) are sufficient to prove the worst-case guarantee stated in Theorem 5.3.1, using the decomposition ideas of Section 5.2.2. Recall that, since the expected revenue function is subadditive, we have

$$\mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{E}}^{*}\right)\right] + \mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{C}}^{*}\right)\right] \geq \mathbb{E}\left[\mathcal{R}\left(U^{*}\right)\right] \ . \tag{5.4}$$

Now, for any $\alpha \in [0,1]$, picking the better vector out of $U_{\mathcal{E}}$ and $U_{\mathcal{C}}$ guarantees, with probability at least $1-\delta$, an expected revenue of

$$\begin{aligned}
\max\left\{\mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{E}}\right)\right], \mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{C}}\right)\right]\right\} &\geq \alpha \cdot \mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{E}}\right)\right] + (1-\alpha) \cdot \mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{C}}\right)\right] \\
&\geq \alpha \cdot (0.318-\epsilon) \cdot \mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{E}}^{*}\right)\right] + \frac{1-\alpha}{4} \cdot \mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{C}}^{*}\right)\right] \\
&\geq (1-4\epsilon) \cdot \left(0.318 \cdot \alpha \cdot \mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{E}}^{*}\right)\right] + \frac{1-\alpha}{4} \cdot \mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{C}}^{*}\right)\right]\right) \ ,
\end{aligned}$$

where the second inequality is an immediate consequence of (5.2) and (5.3). Thus, by choosing $\alpha = 0.25/0.568$,

$$\begin{aligned}
\max\left\{\mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{E}}\right)\right], \mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{C}}\right)\right]\right\} &\geq (1-4\epsilon) \cdot 0.318 \cdot \alpha \cdot \left(\mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{E}}^{*}\right)\right] + \mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{C}}^{*}\right)\right]\right) \\
&\geq (1-4\epsilon) \cdot 0.139 \cdot \mathbb{E}\left[\mathcal{R}\left(U^{*}\right)\right] \ ,
\end{aligned}$$

where the last inequality follows from the upper bound (5.4). As previously mentioned, when $C \geq n$, the inventory vector $U_{\mathcal{E}}$ actually satisfies $\mathbb{E}[\mathcal{R}(U_{\mathcal{E}})] \geq (1 -$

$\epsilon)(1-1/e)\cdot\mathbb{E}[\mathcal{R}(U^*_{\mathcal{E}})]$. By plugging-in this inequality instead of (5.2), and by picking $\alpha = e/(5e-4)$, we derive an improved constant-factor guarantee of $0.179-\epsilon$.

5.3.2 Competing against expensive products

In this section, we consider the *expensive-products problem*, that is, a modified instance only comprised of products in $\mathcal{E}$. We show that the restricted-submodular and restricted-non-decreasing properties are satisfied by the expected revenue function (when reformulated appropriately), although their standard counterparts do not hold in this context. The proof mainly relies on probabilistic coupling ideas, that allow us to compare the consumption process under different initial inventory level decisions. As a result, Lemma 5.2.1 entails the following theorem.

Theorem 5.3.2. *Under the efficient oracle assumption, for any* $\epsilon \in (0,1/4)$ *and* $\delta > 0$*, the expensive-products problem can be approximated within factor* $0.318-\epsilon$ *with probability at least* $1-\delta$*. The running time of our algorithm is polynomial in the input size,* $n^{1/\epsilon}$*, and* $1/\delta$*.*

Set decision formulation. In order to establish the desired submodularity-like properties, the problem needs to be interpreted as the maximization of a set function under a cardinality constraint. To this end, each product is duplicated into C copies, each representing a distinct unit of that product. In the expensive-products problem, this transformation results in an extended set of $N = C\cdot|\mathcal{E}|$ distinct units. With this notation, the objective is to decide on a subset of the extended collection of units $S \subseteq [N]$, as a substitute to the inventory vector U.

Once an initial offer set $S \subseteq [N]$ is chosen, each arriving customer purchases one unit of her most preferred product available, according to the MNL choice model. Since units of the same product are identical, the realizations of the revenue random variable are invariant to the precise unit being purchased, which can thus be chosen arbitrarily (in the sequel, we often impose that a specific unit is purchased, for purposes of analysis). Finally, we define the objective function $f_Y(S)$ to be the expected revenue when initially stocking the subset of units S, where Y stands for the random number of arriving customers. Consequently, the original expensive-products problem translates to maximizing $f_M(S)$ over all subsets $S \subseteq [N]$ of cardinality at most C.

Simplified notation. In what follows, we allow mixed notation between products and their respective units. Specifically, w_i and r_i designate the preference weight and the selling price of the product corresponding to unit i. Unless specified otherwise, when the subset of units $S \subseteq [N]$ is fixed, the corresponding assortment of products is designated by $\mathcal{A} \subseteq \mathcal{E}$. We use the shorthand notation $\mathcal{A}^{+i}$ to denote the resulting assortment when a unit $i \in [N]$ is added to S, and $\mathcal{A}^{+ij}$ when two units $i, j \in [N]$ are added.

5.3.2.1 Probabilistic coupling

To establish the restricted-submodular and restricted-non-decreasing properties, we would have to compare the expected revenue of different subsets. For example, we wish to prove that, for any subset $S \subseteq [N]$ of cardinality at most $C-2$, and any units $i \neq j \in [N] \setminus S$, we have

$$f_M\left(S \cup \{i,j\}\right) - f_M\left(S \cup \{j\}\right) \leq f_M\left(S \cup \{i\}\right) - f_M\left(S\right) \ .$$

To derive such inequalities, we implicitly need to compare the consumption process for the initial subsets S, $S \cup \{i\}$, $S \cup \{j\}$, and $S \cup \{i,j\}$. To this end, our coupling construction will introduce useful relationships between the probabilistic outcomes generated by these subsets. By design, in the construction below, units i and j correspond to two distinct products, both not stocked in S. The construction remains identical even in other settings, where units i and j are arbitrary.

Purchase random variables. We focus on the first arriving customer, and introduce several random variables to describe her purchase decision, when facing each of the above-mentioned subsets. Specifically, denoting the no-purchase option by product 0, with preference weight $w_0 = 1$ and selling price $r_0 = 0$, we define:

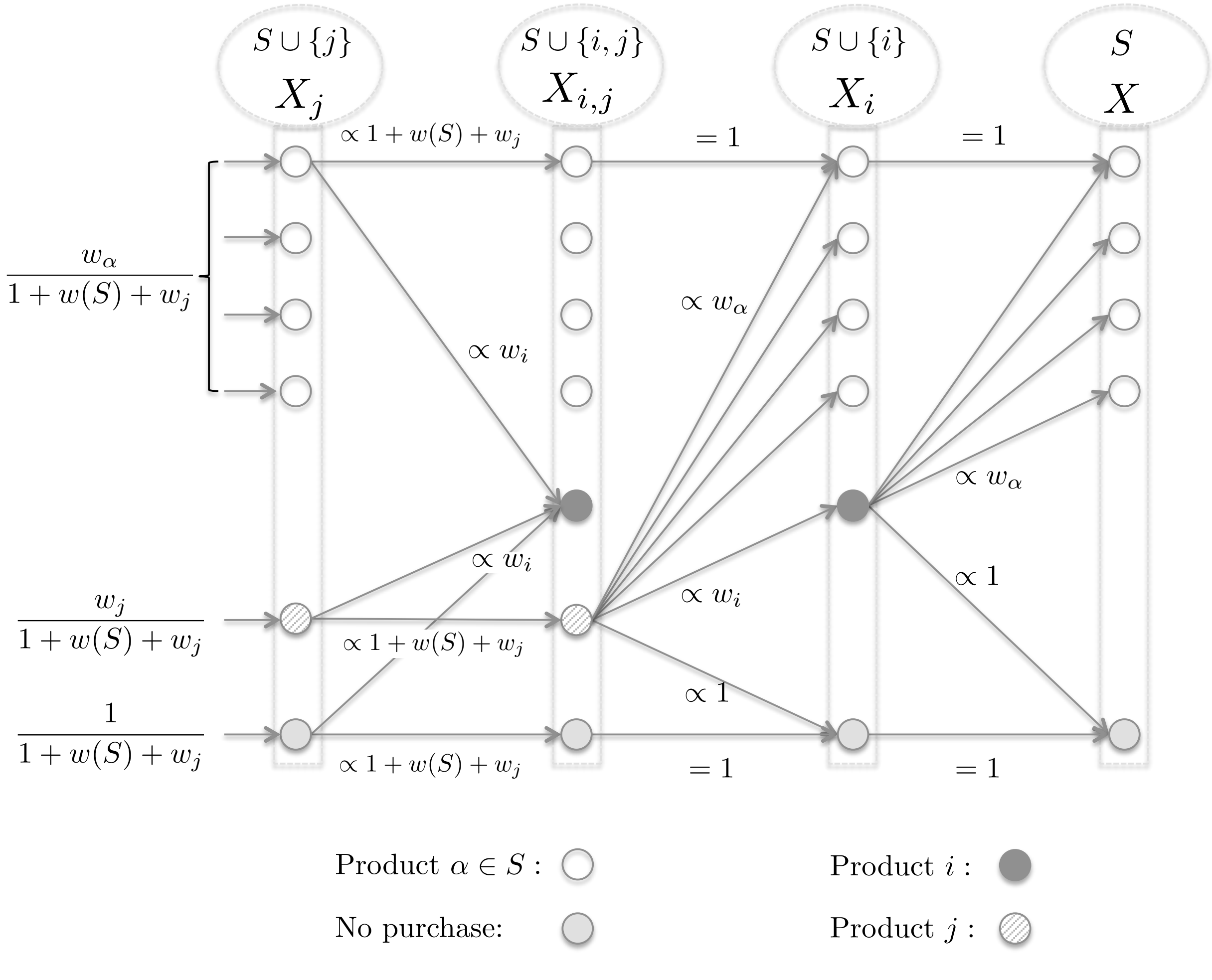
- P as the product purchased when the offered set is S, i.e., within the initial assortment $\mathcal{A} \cup \{0\}$.
- P_i as the product purchased when the offered set is $S \cup \{i\}$, i.e., within the initial assortment $\mathcal{A}^{+i} \cup \{0\}$.
- P_j as the product purchased when the offered set is $S \cup \{j\}$, i.e., within the initial assortment $\mathcal{A}^{+j} \cup \{0\}$.
- $P_{i,j}$ as the product purchased when the offered set is $S \cup \{i,j\}$, i.e., within the initial assortment $\mathcal{A}^{+ij} \cup \{0\}$.

Coupling construction. Rather than defining these random variables separately, in independent probabilistic spaces, we artificially correlate their random outcomes for purposes of analysis, while still preserving their MNL-based marginal probabilities. In other words, denoting by $Y \sim Z$ the equivalence in distribution between two random variables Y and Z, we construct a multivariate distribution for $(X_j, X_{i,j}, X_i, X)$, where $X_j \sim P_j$, $X_{i,j} \sim P_{i,j}$, $X_i \sim P_i$, and $X \sim P$. Our coupling approach relies on stipulating that the sequence $X_j, X_{i,j}, X_i, X$ forms a Markov chain, i.e., $X_i|(X_{i,j}, X_j) = X_i|X_{i,j}$ and $X|(X_i, X_{i,j}, X_j) = X|X_i$, whose transition probabilities are specified below, through the conditional random variables X_j, $X_{i,j}|X_j$, $X_i|X_{i,j}$, and $X|X_i$.

To illustrate the upcoming definitions, we provide in Figure 5-1 a schematic representation of the underlying transition graph, that can be used to derive useful probabilistic claims regarding the purchase random variables. For example, there is a single incoming edge to each white node of $X_{i,j}$, representing the purchase of a product $\alpha \in \mathcal{A}$. Given that this edge is horizontal, it describes the same product option for the variable X_j, and it follows that $\Pr[X_j = \alpha | X_{i,j} = \alpha] = 1$.

- *Defining* X_j. Here, we simply use the marginal probabilities prescribed by the MNL choice model for the purchases made by the first arriving customer under

Figure 5-1: Markov chain representation of the coupling between the random variables X_j,$X_{i,j}$,X_i, and X. Here, random purchase events (or states) are represented by nodes, and each arc corresponds to a transition with positive probability. These transition probabilities are specified either exactly or in proportions (e.g., if a node has two outgoing arcs, one with $\propto 3$ and the other with $\propto 5$, the transition probabilities are 3/8 and 5/8, respectively).



the initial assortment $\mathcal{A}^{+j}$. That is, for any product $\alpha \in \mathcal{A}^{+j} \cup \{0\}$,

$$\Pr\left[X_j = \alpha\right] = \frac{w_\alpha}{1 + w(\mathcal{A}) + w_j} \ .$$

- *Defining* $X_{i,j}|X_j$. The initial set $S \cup \{i,j\}$ contains one more purchase option than $S \cup \{j\}$, namely product i. Intuitively, the event $\{X_{i,j} = i\}$ is defined by 'rescaling' uniformly the purchase probabilities of all other products in $\mathcal{A}^{+j} \cup \{0\}$, which are captured by the variable X_j. Formally, for any product $\alpha \in \mathcal{A}^{+ij} \cup \{0\}$, we define:

$$\begin{aligned}
\Pr\left[X_{i,j} = i|X_j = \beta\right] &= \frac{w_i}{1 + w(\mathcal{A}) + w_j + w_i} \text{ for } \beta \in \mathcal{A}^{+j} \cup \{0\} \ , && (5.5)\\
\Pr\left[X_{i,j} = \alpha|X_j = \alpha\right] &= \frac{1 + w(\mathcal{A}) + w_j}{1 + w(\mathcal{A}) + w_j + w_i} \text{ if } \alpha \neq i \ , && (5.6)\\
\Pr\left[X_{i,j} = \alpha|X_j = \beta\right] &= 0 \text{ if } \alpha \neq i \text{ and } \beta \neq \alpha \ . && (5.7)
\end{aligned}$$

- *Defining* $X_i|X_{i,j}$. We relate the purchases made in the assortment $\mathcal{A}^{+i} \cup \{0\}$ with the purchases made in $\mathcal{A}^{+ij} \cup \{0\}$. In contrast to the previous case, we now need to 'eliminate' the purchase option relative to product j. This is done by 'reallocating' the probability of the event $\{X_{i,j} = j\}$ to the purchases of other products, proportionally to their MNL weights. That is, for any product $\alpha \in \mathcal{A}^{+i} \cup \{0\}$, we define:

$$\begin{aligned}
\Pr\left[X_i = \alpha|X_{i,j} = \alpha\right] &= 1 \ , && (5.8)\\
\Pr\left[X_i = \alpha|X_{i,j} = j\right] &= \frac{w_\alpha}{1 + w(\mathcal{A}) + w_i} && (5.9)\\
\Pr\left[X_i = \alpha|X_{i,j} = \beta\right] &= 0 \text{ for } \beta \neq \alpha, j \ . && (5.10)
\end{aligned}$$

- *Defining* $X|X_i$. Our construction is similar to the previous case, and for any product $\alpha \in \mathcal{A} \cup \{0\}$, we define:

$$
\begin{aligned}
\Pr\left[X = \alpha | X_i = \alpha\right] &= 1 \ , && (5.11)\\
\Pr\left[X = \alpha | X_i = i\right] &= \frac{w_\alpha}{1 + w(\mathcal{A})} \ . && (5.12)\\
\Pr\left[X = \alpha | X_i = \beta\right] &= 0 \text{ for } \beta \neq \alpha, i \ . && (5.13)
\end{aligned}
$$

The next lemma, whose proof is given in Appendix C.3.3, states that this coupling method indeed preserves the (marginal) MNL purchase probabilities for each initial offer set.

Claim 5.3.3. *$X_j \sim P_j$, $X_{i,j} \sim P_{i,j}$, $X_i \sim P_i$, and $X \sim P$.*

In addition, we establish several equivalence properties that will prove useful for the analysis, stating that the purchase random variables X, X_i, and X_j are invariant in distribution when conditioned on appropriate events of $X_{i,j}$. The proof of the next lemma appears in Appendix C.3.4.

Claim 5.3.4. *$(X_j | X_{i,j} = i) \sim X_j$, $(X_i | X_{i,j} = j) \sim X_i$, $(X | X_{i,j} = i) \sim X$, and $(X | X_{i,j} = j) \sim X$.*

5.3.2.2 Proving restricted monotonicity

We prove that the revenue function f_M is restricted-non-decreasing by an inductive argument. To better understand which sufficient properties come into play, the proof is broken down into two lemmas: we first examine the case of a single arriving customer, before extending our arguments to any random variable M. It is worth mentioning that this property holds regardless of how M is distributed, whether IFR or not.

Lemma 5.3.5. *In the expensive-products setting, the static expected revenue function f_1 is restricted-non-decreasing.*

Proof. It is easy to verify that the restricted-non-decreasing property is equivalent to having $f_1(S \cup \{i\}) \geq f_1(S)$ for any subset S of size at most $C - 1$ and any unit

i. Observe that, if the product corresponding to i is stocked by S, we clearly have $f_1(S \cup \{i\}) = f_1(S)$. When this product is not stocked in S,

$$\begin{aligned}
f_1\left(S \cup \{i\}\right) - f_1\left(S\right) &= \frac{r_i w_i}{1 + w(\mathcal{A}) + w_i} + \sum_{k \in S} r_k w_k \cdot \left(\frac{1}{1 + w(\mathcal{A}) + w_i} - \frac{1}{1 + w(\mathcal{A})} \right) \\
&= \frac{w_i}{1 + w(\mathcal{A}) + w_i} \cdot \left(r_i - \sum_{k \in S} \frac{r_k w_k}{1 + w(\mathcal{A})} \right) \\
&= \frac{w_i}{1 + w(\mathcal{A}) + w_i} \cdot \left(r_i - f_1\left(S\right)\right) \ . \qquad (5.14)
\end{aligned}$$

This proves the desired inequality, since $r_i \geq \mathrm{OPT}_{\mathrm{static}} \geq f_1(S)$, where the former inequality follows from i being an expensive product, and the latter holds since the assortment $\mathcal{A}$ stocked by S has fewer than C products (all expensive), implying that its static expected revenue is at most $\mathrm{OPT}_{\mathrm{static}}$, which stands for the maximum possible static revenue when we are allowed to stock at most C products (expensive and cheap). □

Lemma 5.3.6. *For any instance of the dynamic assortment planning problem where the static revenue function f_1 is restricted-non-decreasing, the revenue function f_M is restricted-non-decreasing as well, for any demand random variable M.*

Proof. The preliminary observation is that, by the formula of conditional expectation, it is sufficient to prove the desired property for a deterministic value of M. Also, as noted earlier, it is sufficient to prove that $f_M(S_1) \leq f_M(T_1)$ for any two initial offer sets $S_1 \subseteq T_1 \subseteq [N]$ with cardinality at most C, that differ by at most one unit, i.e., $|T_1 \setminus S_1| \leq 1$.

To this end, we leverage our coupling method for the purchase random variables, constructed in Section 5.3.2.1, to derive a coupling of the consumption processes under the initial subsets S_1 and T_1. Let $S_1, \ldots, S_M$ and $T_1, \ldots, T_M$ be the (random) residual subsets of inventory units facing customers $1, \ldots, M$, when respectively stocking the initial subsets S_1 and T_1. We denote by $\mathcal{A}_1, \ldots, \mathcal{A}_M$ and $\mathcal{B}_1, \ldots, \mathcal{B}_M$ the corresponding sequences of assortments.

We wish to define a coupling of these random variables such that $S_k \subseteq T_k$ and

$|T_k \setminus S_k| \leq 1$, at each arrival $k \in [M]$. This coupling is constructed inductively over the sequence of arrivals, by refining at each step our probabilistic space with respect to the next arriving customer. Since the desired properties are clearly satisfied for the base case $k = 1$ by definition of S_1 and T_1, suppose the inductive hypothesis holds until the k-th arrival. If $S_k = T_k$, it is easy to see that the inductive property propagates to the next arrivals, since the purchases made in the two consumption processes are identical. Otherwise, let i be the (single) unit contained in $T_k \setminus S_k$. For the next arriving customer, we distinguish between two cases:

- *Product i is contained in $\mathcal{A}_k$.* As a result, the k-th arriving customer is facing the same assortment under both offer sets S_k and T_k, i.e., $\mathcal{A}_k = \mathcal{B}_k$. Here, the purchase random variable X (with $S = S_k$) defines a trivial coupling of the purchases made by the first arriving customer in both cases, in the sense that the purchases are identical and described by the outcomes of X with respect to $\mathcal{A}_k$. Consequently, the random residual sets facing the next arriving customer satisfy $S_{k+1} \subseteq T_{k+1}$ since $S_k \subseteq T_k$ and the same unit of product X can be purchased in both cases. In addition, we have $|T_{k+1} \setminus S_{k+1}| = |T_k \setminus S_k| \leq 1$.

- *Product i is not contained in $\mathcal{A}_k$.* In this case, taking $S = S_k$ and $\mathcal{A} = \mathcal{A}_k$, we use the coupling of X and X_i as a joint distribution for the purchases made by the first arriving customer under the sets S_k and T_k, respectively. By definition of $X|X_i$, observe that if the customer faced with T_k purchases a product in $\mathcal{A} \cup \{0\}$, i.e., $X_i \in \mathcal{A} \cup \{0\}$, then the customer faced with S_k purchases the same product, i.e., $X = X_i$ (in Figure 5-1, there is a single horizontal edge going into each white node of X, describing the same purchase option in X and X_i). Indeed, our coupling entails that $\Pr[X = X_i | X_i \in \mathcal{A} \cup \{0\}] = 1$ due to equation (5.11). As a result, since the k-th customer purchases the same unit in both cases, the inductive hypothesis implies that $S_{k+1} \subseteq T_{k+1}$. On the other hand, the event $\{X_i = i\}$ means that the customer faced with T_k purchases the last unit of product i, and thus, conditional to this event, the remaining set of units is necessarily $T_{k+1} = T_k \setminus \{i\} = S_k$, which clearly leads to $S_{k+1} \subseteq T_{k+1}$.

In both cases, we have preserved the invariant $|T_{k+1} \setminus S_{k+1}| \leq 1$.

We have just obtained a coupling of the consumption processes such that $S_k \subseteq T_k$ and $|T_k \setminus S_k| \leq 1$ for every $k \in [M]$. By exploiting this inclusion property between subsets of units, we now prove that $f_M(S_1) \leq f_M(T_1)$. To this end, a natural transformation of the expected revenue function is

$$f_M(S_1) = \sum_{k=1}^{M} \mathbb{E}\left[f_1(S_k)\right] \ , \tag{5.15}$$

where the overall expected revenue breaks down into the sum of expected revenues generated by customers $1, \ldots, M$, faced by the random residual sets of units $S_1, \ldots, S_M$, respectively. Using a similar transformation for T_1, we have

$$f_M\left(T_1\right) - f_M\left(S_1\right) = \sum_{k=1}^{M} \mathbb{E}\left[f_1(T_k) - f_1(S_k)\right] \ .$$

Therefore, since f_1 is assumed to be restricted-non-decreasing, and $S_k \subseteq T_k$, the latter expression is non-negative, meaning that the restricted-non-decreasing property extends to f_M. □

5.3.2.3 Proving restricted submodularity

We now show that the transformed revenue function f_M is also restricted-submodular, regardless of how M is distributed, by exploiting the coupling method described in Section 5.3.2.1. We first examine the static case, before extending the desired property to any demand random variable.

Lemma 5.3.7. *For any instance of the dynamic assortment planning problem where the static expected revenue function f_1 is restricted-non-decreasing, this function is restricted-submodular.*

Proof. Let $S \subseteq [N]$ be a set with $|S| \leq C - 2$, $\mathcal{A}$ is the assortment stocked by S and let $i \neq j$ be two units in $[N] \setminus S$. In order to prove that $f_1(S \cup \{i, j\}) - f_1(S \cup \{j\}) \leq f_1(S \cup \{i\}) - f_1(S)$, we distinguish between four cases:

1. *Product i is contained in $\mathcal{A}$.* Adding unit i to any subset of units containing S leaves us with the same assortment $\mathcal{A}$, meaning that $f_1(S \cup \{i,j\}) - f_1(S \cup \{j\}) = f_1(S \cup \{i\}) - f_1(S) = 0$.

2. *Product j is contained in $\mathcal{A}$, product i is not.* In this case, adding unit j to any subset containing S leaves us with the same assortment, thus $f_1(S \cup \{i,j\}) - f_1(S \cup \{i\}) = f_1(S \cup \{j\}) - f_1(S)$.

3. *Units i and j are of the same product, not contained in $\mathcal{A}$.* Here, we observe that $f_1(S \cup \{i,j\}) - f_1(S \cup \{j\}) = 0$ while $f_1(S \cup \{i\}) - f_1(S) \geq 0$ since f_1 is restricted-non-decreasing.

4. *Products i and j are different, and both not contained in $\mathcal{A}$.* By calculations similar to those leading to equation (5.14), we get

$$\begin{aligned} f_1\left(S \cup \{i,j\}\right) - f_1\left(S \cup \{j\}\right) &= \frac{w_i}{1 + w(\mathcal{A}) + w_i + w_j} \cdot \left(r_i - f_1\left(S \cup \{j\}\right)\right) \\ &\leq \frac{w_i}{1 + w(\mathcal{A}) + w_i} \cdot \left(r_i - f_1\left(S\right)\right) \\ &= f_1\left(S \cup \{i\}\right) - f_1\left(S\right) \ , \end{aligned}$$

   where the inequality above holds since f_1 is restricted-non-decreasing, thus $f_1(S \cup \{j\}) \geq f_1(S)$.

□

Lemma 5.3.8. *For any instance of the dynamic assortment planning problem where the static expected revenue function f_1 is restricted-non-decreasing, the revenue function f_M is restricted-submodular as well, for any demand random variable M.*

Proof. By the formula of conditional expectations, we restrict attention to deterministic values of M without loss of generality, and prove the claim by induction on M. Since the case $M = 1$ corresponds to Lemma 5.3.7, suppose that restricted submodularity has been established for $M - 1$ arrivals. We show that, for any subset $S \subseteq [N]$

of cardinality at most $C-2$, and units $i \neq j \in [N] \setminus S$,

$$f_M\left(S \cup \{i,j\}\right) - f_M\left(S \cup \{j\}\right) \leq f_M\left(S \cup \{i\}\right) - f_M\left(S\right) \ . \tag{5.16}$$

The proof consists of the same case analysis made for the proof of Lemma 5.3.7. In what follows, we only discuss the most difficult case, where products i and j are different and not contained in the assortment $\mathcal{A}$ stocked by S. The additional cases have nearly-identical proofs and are not presented here to avoid redundancy.

A particularly instructive observation is that, for any subset of units $T \subseteq [N]$, the expected revenue function decomposes into the contribution due to the purchase made by the first arriving customer, and that associated with the residual subset of units and the remaining customers arrivals. Formally, letting $R_M(T)$ denote the random revenue obtained after M arriving customer facing an initial subset T, and using Y to designate the product purchased by the first arriving customer, we have

$$f_M(T) = \mathbb{E}\left[R_M(T)\right] = \mathbb{E}\left[r_Y\right] + \mathbb{E}\left[R_{M-1}\left(T \setminus \{Y\}\right)\right] \ . \tag{5.17}$$

In what follows, for ease of notation, we denote $S^{+i} = S \cup \{i\}$, $S^{+j} = S \cup \{j\}$, and $S^{+ij} = S \cup \{i,j\}$. In order to derive inequality (5.16), we make use of the revenue decomposition (5.17) under different initial subsets S, S^{+i}, S^{+j}, and S^{+ij}, along with their corresponding purchase random variables X, X_i, X_j, and $X_{i,j}$.

By conditioning the revenue random variable with respect to $X_{i,j}$, we leverage our coupling method of Section 5.3.2.1 to explicitly compare the random purchases made by the first arriving customer under different initial subsets, and establish the next claim. Finally, the desired inequality (5.16) comes at an immediate consequence of this claim, combined with the formula of conditional expectations.

Claim 5.3.9. $\mathbb{E}[R_M(S^{+ij}) - R_M(S^{+j})|X_{i,j} = \alpha] \leq \mathbb{E}[R_M(S^{+i}) - R_M(S)|X_{i,j} = \alpha]$

Proof of Claim 5.3.9. We present here the case where $\alpha \in \mathcal{A} \cup \{0\}$, the other cases being treated through similar arguments in Appendix C.3.5. We begin by observing that, conditional on the event $\{X_{i,j} = \alpha\}$ where $\alpha \in \mathcal{A} \cup \{0\}$, we necessarily have

$X_j = X_{i,j} = X_i = X = \alpha$. Indeed, in the transition graph of the Markov chain (Figure 5-1), observe that there is a single horizontal path going through each white node of $X_{i,j}$, that describes the same purchase option across all variables. Formally, by equation (5.8), observe that $\Pr[X_i = \alpha | X_{i,j} = \alpha] = 1$, while equation (5.11) implies that $\Pr[X = \alpha | X_i = \alpha] = 1$. Finally, $\Pr[X_j = \alpha | X_{i,j} = \alpha] = 1$ follows from Bayes rule, using equation (5.6) along with the marginal distributions of X_j and $X_{i,j}$, which are described by the MNL model (see Claim 5.3.3). Thus, we obtain:

$$\begin{aligned}
&\mathbb{E}\left[R_M\left(S^{+ij}\right) - R_M\left(S^{+j}\right) \middle| X_{i,j} = \alpha \right] \\
&\qquad = \mathbb{E}\left[r_{(X_{i,j}|X_{i,j}=\alpha)} + R_{M-1}\left(S^{+ij} \setminus \{\alpha\}\right) \right] - \mathbb{E}\left[r_{(X_j|X_{i,j}=\alpha)} + R_{M-1}\left(S^{+j} \setminus \{\alpha\}\right) \right] \\
&\qquad = \mathbb{E}\left[R_{M-1}\left(S^{+ij} \setminus \{\alpha\}\right) - R_{M-1}\left(S^{+j} \setminus \{\alpha\}\right) \right] \\
&\qquad \leq \mathbb{E}\left[R_{M-1}\left(S^{+i} \setminus \{\alpha\}\right) - R_{M-1}\left(S \setminus \{\alpha\}\right) \right] \\
&\qquad = \mathbb{E}\left[R_M\left(S^{+i}\right) - R_M\left(S\right) \middle| X_{i,j} = \alpha \right] \ , \qquad (5.18)
\end{aligned}$$

where the first equality proceeds from equation (5.17), the second equality holds since the terms $r_{(X_{i,j}|X_{i,j}=\alpha)} = r_{(X_j|X_{i,j}=\alpha)} = r_\alpha$ cancel out, the next inequality is due to the inductive hypothesis (5.16), and the last equality is analogous to the first two equalities (in reverse order). ■

□

5.3.2.4 Improved performance guarantees for special settings

A close examination of the statements of Lemmas 5.3.6, 5.3.7, and 5.3.8 reveals that, for any integer $s \in [N]$, the static expected revenue function f_1 being restricted-s-non-decreasing is a sufficient condition for f_M to be both restricted-s-non-decreasing and restricted-s-submodular. Hence, when f_1 is non-decreasing in the standard sense, it follows that the function f_M is non-decreasing and submodular. In such cases, the standard analysis of the greedy algorithm for monotone submodular maximization (Nemhauser et al. 1978), with appropriately-chosen error and confidence parameters for the evaluation oracle, provides an improved performance guarantee of

$(1 - \epsilon) \cdot (1 - 1/e)$.

It is worth mentioning that the expensive-products problem naturally satisfies this condition when $C \geq n$. Indeed, by Lemma 5.3.5, we know that f_1 is restricted-n-non-decreasing. Now, consider a transformation that maps each set $S \subseteq [N]$ to the subset $\tilde{S} \subseteq S$ obtained by keeping the lowest-index unit of each product stocked by S. Clearly, this transformation preserves the static expected revenue, i.e., $f_1(S) = f_1(\tilde{S})$, while ensuring that $\tilde{S}$ has at most n units. Consequently, for any subsets $S \subseteq T \subseteq [N]$, we infer that $f_1(S) = f_1(\tilde{S}) \leq f_1(\tilde{T}) = f_1(T)$ since f_1 is restricted-n-non-decreasing and $\tilde{S} \subseteq \tilde{T}$.

This condition is also satisfied when the assortment of products has been chosen in advance by solving the standard assortment optimization problem under the MNL model, and it remains to set their inventory levels. It is easy to verify that f_1 is non-decreasing when the entire collection of products forms an optimal static assortment.

5.3.3 Competing against cheap products

In this section, we construct an inventory vector $U_{\mathcal{C}}$ that guarantees a constant fraction of the expected revenue of $U^*_{\mathcal{C}}$, which stands for the projection of the optimal inventory U^* on the cheap products. We begin by presenting the algorithm before stating the performance guarantee obtained.

5.3.3.1 Algorithm

Step 1: Computing an optimal static assortment. As explained in Section 5.3.1, we begin by optimally solving the standard assortment planning problem, subject to a capacity constraint of C on the number of products offered (Megiddo 1979, Rusmevichientong et al. 2010, Davis et al. 2013). Recall that the corresponding optimal static assortment, that generates an expected revenue of $\text{OPT}_{\text{static}}$, is denoted by $\mathcal{A}^*$. We now highlight a basic property of optimal static assortments, claiming that only expensive products are being stocked.

Claim 5.3.10. *$\mathcal{A}^* \subseteq \mathcal{E}$.*

This property follows from existing work on the capacitated assortment optimization problem under the MNL model (see Proposition 1 of Talluri and van Ryzin (2004) and endnote 2 of Rusmevichientong et al. (2010)). For completeness, we provide a short proof in Appendix C.3.6.

Step 2: Deriving a newsvendor-like lower bound. To derive a lower bound, we will neglect the effects of dynamic substitution, and consider a setting where customers purchase their most preferred product until it stocks out. Specifically, suppose that U is an initial inventory vector stocking only units in the assortment $\mathcal{A}^*$. Then, for any product $i \in \mathcal{A}^*$, the probability that an arriving customer purchases that product is at least $\psi_i = w_i/(1 + w(\mathcal{A}^*))$, regardless of the inventory levels of the other products, as long as product i has not stocked out. Indeed, as the inventory vector is depleted due to previously-arriving customers, this can only increase the probability of each remaining product to be consumed by the next customer. To better understand the latter claim, consider two assortments $\tilde{\mathcal{A}} \subseteq \mathcal{A}$. The probability that an arriving customer purchases a unit of product $i \in \mathcal{A}$, when faced with assortment $\mathcal{A}$, is $w_i/(1+w(\mathcal{A}))$. By inclusion, this quantity is smaller or equal to $w_i/(1+w(\tilde{\mathcal{A}}))$, namely the probability of picking i among the assortment $\tilde{\mathcal{A}}$.

Consequently, the number of units purchased from i if this product had an unlimited number of units is stochastically larger than the binomial random variable $Y_i \sim B(M, \psi_i)$. However, since product i has only u_i units, we will actually be considering the truncated random variable $\bar{Y}_i(u_i) = \min\{Y_i, u_i\}$. Therefore, we obtain the following lower bound:

$$\mathbb{E}\left[\mathcal{R}\left(U\right)\right] \geq \sum_{i \in \mathcal{A}^*} r_i \cdot \mathbb{E}\left[\bar{Y}_i(u_i)\right] \ . \tag{5.19}$$

This lower bound can be interpreted as the objective function of a multi-item newsvendor problem, where the demand is separable across the products of $\mathcal{A}^*$. In what follows, this function is denoted by $\mathcal{L}(U) = \sum_{i \in \mathcal{A}^*} r_i \cdot \mathbb{E}\left[\bar{Y}_i(u_i)\right]$.

Step 3: Computing $U_{\mathcal{C}}$ by greedily optimizing the lower bound. Finally, the inventory vector $U_{\mathcal{C}}$, used to compete against cheap products, is constructed by solving the multi-item newsvendor instance defined above. That is, we compute U that maximizes $\mathcal{L}(U)$, subject to $\sum_{i \in \mathcal{A}^*} u_i \leq C$. This optimization problem can be solved exactly by a standard greedy procedure (see Muckstadt and Sapra (2010, Chap. 5)). Namely, starting from an empty inventory vector, units are added iteratively until reaching the capacity C, by picking at each step the unit with largest marginal contribution to the objective function. In contrast to the original expected revenue, for any inventory vector U the lower bound $\mathcal{L}(U)$ can be computed by a simple dynamic program in polynomial time.

5.3.3.2 Performance guarantee

The remainder of this section is devoted to proving the next theorem, showing that $U_{\mathcal{C}}$ competes against the expected revenue of cheap products $U^*_{\mathcal{C}}$.

Theorem 5.3.11. $\mathbb{E}[\mathcal{R}(U_{\mathcal{C}})] \geq (1/4) \cdot \mathbb{E}[\mathcal{R}(U^*_{\mathcal{C}})]$.

Our analysis proceeds by comparing an upper bound on $\mathbb{E}[\mathcal{R}(U^*_{\mathcal{C}})]$ with a lower bound on $\mathbb{E}[\mathcal{R}(U_{\mathcal{C}})]$, using the IFR property. It bears some resemblance to the analysis of Chapter 4, combined with additional structural properties of the MNL choice model.

Upper bound on the expected revenue of $U^*_{\mathcal{C}}$. The important observation is that, when initially stocking at most C units of cheap products, each arriving customer will generate an expected revenue of at most $\mathrm{OPT}_{\mathrm{static}}$. As a result, the expected revenue of $U^*_{\mathcal{C}}$ is upper bounded by $\mathbb{E}[M] \cdot \mathrm{OPT}_{\mathrm{static}}$. In addition, since all cheap products have by definition selling prices smaller than $\mathrm{OPT}_{\mathrm{static}}$, another upper bound on the expected revenue of $U^*_{\mathcal{C}}$ is $C \cdot \mathrm{OPT}_{\mathrm{static}}$. Therefore,

$$\mathbb{E}\left[\mathcal{R}\left(U^*_{\mathcal{C}}\right)\right] \leq \mathrm{OPT}_{\mathrm{static}} \cdot \min\left\{C, \mathbb{E}\left[M\right]\right\} \ . \tag{5.20}$$

Lower bound on the expected revenue of $U_{\mathcal{C}}$. To define our lower bound, we begin by introducing an inventory vector $U^{\propto}$, where the inventory levels are scaled proportionally to their revenue contribution toward $\mathrm{OPT}_{\mathrm{static}}$. Ideally, for each product $i \in \mathcal{A}^*$, the inventory level of i represents a fraction of $r_i\psi_i/\mathrm{OPT}_{\mathrm{static}}$ of the total capacity C (recalling that $\psi_i = w_i/(1 + w(\mathcal{A}^*))$). Hence, we would have liked to define the vector $\tilde{U}$, where $\tilde{u}_i = (r_i\psi_i/\mathrm{OPT}_{\mathrm{static}}) \cdot C$. However, this quantity may not be integral, and is therefore rounded up to the nearest integer, creating the vector $U^{\propto}$. That is, $u_i^{\propto} = \lceil \tilde{u}_i \rceil$ for every product $i \in \mathcal{A}^*$, and $u_i^{\propto} = 0$ otherwise. Due to our rounding procedure, the overall number of units stocked by $U^{\propto}$ exceeds the capacity C by a factor of at most 2 since

$$\sum_{i \in \mathcal{A}^*} u_i^{\propto} = \sum_{i \in \mathcal{A}^*} \lceil \tilde{u}_i \rceil \leq \sum_{i \in \mathcal{A}^*} \tilde{u}_i + |\mathcal{A}^*| \leq \frac{C}{\mathrm{OPT}_{\mathrm{static}}} \cdot \sum_{i \in \mathcal{A}^*} r_i\psi_i + |\mathcal{A}^*| \leq 2C \ . \tag{5.21}$$

Since the newsvendor-like objective function $\mathcal{L}$ has diminishing marginals (Muckstadt and Sapra 2010, Chap. 5), we infer that $\mathcal{L}(U^{\propto}) \leq 2 \cdot \mathcal{L}(U_{\mathcal{C}})$ by observing that $U_{\mathcal{C}}$ is an optimal inventory vector for $\mathcal{L}$ with C units, while $U^{\propto}$ has at most $2C$ units. Therefore, by the lower bound (5.19),

$$\mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{C}}\right)\right] \geq \mathcal{L}\left(U_{\mathcal{C}}\right) \geq \frac{\mathcal{L}\left(U^{\propto}\right)}{2} \ . \tag{5.22}$$

Comparing the upper bound (5.20) **with the lower bound** (5.22)**.** In the next claim, we leverage the structure of $U^{\propto}$ as well as the IFR property to obtain a lower bound on the marginal contribution of each product in $\mathcal{A}^*$ toward $\mathcal{L}(U^{\propto})$. The proof is given in Appendix C.3.7

Claim 5.3.12. *For every product $i \in \mathcal{A}^*$, $\mathbb{E}[\bar{Y}_i(u_i^{\propto})] \geq (1/2) \cdot \min\{u_i^{\propto}, \mathbb{E}[Y_i]\}$.*

By plugging Claim 5.3.12 into the lower bound stated in (5.22), we conclude that

$$\begin{aligned}
\mathbb{E}\left[R\left(U_{\mathcal{C}}\right)\right] &\geq \frac{1}{4} \cdot \sum_{i \in \mathcal{A}^*} r_i \cdot \min\left\{u_i^{\propto}, \mathbb{E}\left[Y_i\right]\right\} \\
&\geq \frac{1}{4} \cdot \sum_{i \in \mathcal{A}^*} r_i \cdot \min\left\{\tilde{u}_i, \mathbb{E}\left[M\right] \cdot \psi_i\right\} \\
&= \frac{1}{4} \cdot \sum_{i \in \mathcal{A}^*} r_i \tilde{u}_i \cdot \min\left\{1, \frac{\mathbb{E}\left[M\right] \cdot \mathrm{OPT}_{\mathrm{static}}}{C \cdot r_i}\right\} \\
&\geq \frac{\mathrm{OPT}_{\mathrm{static}} \cdot \min\left\{C, \mathbb{E}\left[M\right]\right\}}{4C} \cdot \sum_{i \in \mathcal{A}^*} \tilde{u}_i \\
&\geq \frac{\mathbb{E}\left[R\left(U_{\mathcal{C}}^*\right)\right]}{4C} \cdot \sum_{i \in \mathcal{A}^*} \tilde{u}_i \\
&= \frac{\mathbb{E}\left[R\left(U_{\mathcal{C}}^*\right)\right]}{4} \ .
\end{aligned}$$

Here, the second inequality holds since $u_i^{\propto} = \lceil \tilde{u}_i \rceil$, the next equality follows from the definition of $\tilde{u}_i$, the third inequality holds since all products in $\mathcal{A}^*$ are expensive by Claim 5.3.10, meaning that $r_i \geq \mathrm{OPT}_{\mathrm{static}}$, the fourth inequality is derived from the upper bound in (5.20), and the last equality holds since $\sum_{i \in \mathcal{A}^*} \tilde{u}_i = C$.

5.4 Computational Experiments

In this section, we show that our algorithmic approach has a superior empirical performance in comparison to existing heuristics on randomly-generated instances. In particular, substantial gains in the expected revenue are demonstrated against these heuristics, with better computational efficiency and robustness.

5.4.1 Generative models

Products and MNL parameters. Our simulations make use of $n = 20$ products and a capacity bound of C, taking values in $\{25, 50, 100\}$. Instances of the MNL model are constructed by considering two alternative settings, with different levels of heterogeneity in revenues and preferences.

- *Setting A:* The preference weights w_i are i.i.d. samples of a uniform distribution over the interval $[0, 1]$. The per-unit selling prices r_i are i.i.d. random samples of a standard log-normal random variable (with $\mu = 0$ and $\sigma = 1$).

- *Setting B:* Here, we create instances having a greater dispersion of weights and prices. Specifically, the weights are generated through i.i.d. samples of a standard log-normal distribution, rescaled by a factor of 1/2 to remain on average equivalent to setting A. The per-unit selling prices r_i are now sampled from a log-normal distribution with $\mu = 0$ and $\sigma = 2$.

The demand. The random number of arriving customers M is generated through two families of distributions with finite support $0, \ldots, 100 = \bar{M}$: a truncated Poisson distribution and randomly generated nonparametric distributions. The former uses a random variable $\mathcal{P} \sim \text{Poisson}(0.35 \cdot \bar{M})$, such that $M = \min\{\mathcal{P}, \bar{M}\}$. The latter nonparametric distributions are constructed as follows. To enforce the IFR property, we generate a decreasing sequence of $\bar{M}$ failure rates, each of at most 5%. To this end, we first draw $\bar{M}$ i.i.d. samples of the uniform distribution over the interval $[0, 0.04]$, which are next sorted by increasing values, to obtain a sequence $z_1 \leq \cdots \leq z_{\bar{M}}$. This sequence is used to specify the failure rate $\Pr[M = k | M \geq k] = z_{k+1}$ for every $k \in [0, \bar{M} - 1]$, and in addition, $\Pr[M = \bar{M} | M \geq \bar{M}] = 1$.

5.4.2 Tested heuristics

The performance of our algorithm is compared against five different heuristics, whose specifics are discussed in Appendix C.4: (i) a local search heuristic similar to that of Goyal et al. (2016); (ii) a gradient-descent approach based on a continuous extension of the revenue function, similar in spirit to the work of Mahajan and van Ryzin (2001); (iii) the dynamic programming formulations devised by Topaloglu (2013) for variants of our problem; (iv) the deterministic relaxation heuristic proposed by Honhon et al. (2010); (v) a discrete-greedy heuristic. The latter forms a natural benchmark since our approach relies primarily on greedy decisions, with the main difference of stocking products within a restricted set, possibly with modified objective functions. In addition, we report the revenue performance of the subroutine used in Section 5.3.3 to compete against cheap products, that scales the inventory levels proportional to the expected sales within an optimal static assortment.

5.4.3 Additional technical details

We implemented our algorithm, as well as the above-mentioned heuristics, using the Python programming language. The experiments described in this section were conducted on a standard laptop with 2.5GHz Intel Core i5 processor and 8GB of RAM.

The number of tested instances for each combination of parameters is 20. We impose a time limit of 1000 seconds (per instance) for every algorithm. In case this limit is reached before termination, we use an identical rounding procedure on the current solution. Specifically, letting U be the best inventory vector found after 1000 seconds, U is linearly scaled and rounded down to the nearest integral vector: $U'_i = \lfloor \frac{U_i}{\|U\|_1} \cdot C \rfloor$. Finally, units are greedily added to U' until an inventory vector of exactly C units is obtained.

To approximately evaluate the expected revenue function, each call to the random oracle results in 500 samples. Although the number of samples needed to derive our theoretical guarantee in Lemma C.1.3 could be significantly larger, we observed in preliminary experiments that a greater number of samples has negligible impact on the performance of the algorithms considered. However, the gradient-descent approach and discrete-greedy algorithm become rapidly impractical for the instances tested when the number of samples is increased.

Relative performance. For each instance tested, obtaining an estimate of the optimal expected revenue through brute-force enumeration is computationally prohibitive. Furthermore, we are not aware of any good empirical upper bound on the optimal expected revenue. For example, using a sample average approximation method, the resulting problem can be formulated as an integer program. However, using a state-of-the-art commercial solver (Gurobi Optimization 2015), this IP incurred running times greater than 1 hour, even for the simplest instances with $n = \bar{M} = C = 20$ and 500 samples. The latter approach can be made more tractable using a relaxation, where a custom solution is computed for each sample-path realization through a separate IP. However, this approach produces low quality approximations.

For these reasons, we do not estimate the exact optimality gap attained by each algorithm. Instead, the algorithms are compared on a relative basis where, for each instance, the benchmark is set as the expected revenue of the most profitable inventory vector obtained through all 4 algorithms. Then, the *relative performance* of each algorithm is defined as the ratio between its expected revenue and that of the benchmark. For example, if our algorithm attains for a particular instance an expected revenue of 1, while all tested heuristics generate an expected revenue of 0.9, the relative performance is 100% for our algorithm, and 90% for the others.

5.4.4 Results

Practical performance. As shown in Table 5.1, our algorithm exhibits moderate to substantial performance gains in comparison to the heuristics under consideration, for all configurations. Specifically, the average performance gains of our algorithm range from 0.2% to 35.3%. Overall, the expected revenues are increased by an average factor of 2.4% in comparison to the proportional scaling heuristic (a subroutine of our algorithm to compete against cheap products), 7.4% in comparison to the Poisson-based dynamic program, 6.8% in comparison to the normal approximation-based dynamic program, 10.3% in comparison to the deterministic relaxation, 16.4% against the local search algorithm, 8.5% in comparison to gradient descent, and 10.4% in comparison to discrete-greedy. In addition, our algorithm is robust, as it outperforms all heuristics for 66% of the instances.

Although discrete-greedy is closest in spirit to our algorithm, particularly for expensive products, the relative performance gap between the two approaches is significant. Since the discrete-greedy algorithm is given access to a larger space of incremental actions (augmenting the inventory level of any product) at each iteration

Table 5.1: Average revenue performance of the different algorithms tested.

Parameters			**Relative revenue performance (%)**							
M	Setting	C	ALG	PRO	PDP	NDP	DET	LS	GD	DG
Poisson	A	25	99.6	98.8	96.6	94.5	79.9	90.5	95.1	97.9
		50	99.7	98.6	93.8	88.2	75.7	84.4	96.2	96.4
		100	99.9	97.6	86	94	85.8	86.3	96.3	93.6
	B	25	99.3	97.5	97.7	97	88.3	90.7	95.6	96.3
		50	99.7	97.6	91.3	95.3	92.3	92.5	95.2	81.5
		100	99.3	96.9	78.8	94.6	94.8	92	92.4	80.6
Nonparametric	A	25	98.6	91.2	97.2	90.9	88.8	86.3	80.3	98.4
		50	99.5	97.3	97.9	89	90.8	68.8	91.3	94.1
		100	99.9	97.3	88.8	92	90.4	64.7	86.5	81.4
	B	25	98.5	96.1	95.6	92.4	96.2	87.2	92.2	93.1
		50	99.9	98.4	91.4	90	93.3	75	91.4	82.9
		100	100	97.5	90.5	94.8	94.2	79.3	80	73.4

Here, ALG designates our algorithm, PRO is the subroutine of our algorithm to compete against cheap products, PDP is the dynamic program under a Poisson process, NDP is the dynamic program under a normal approximation, DET is the deterministic relaxation, LS corresponds to the local search heuristic, GD designates the gradient-descent approach, and DG is the discrete-greedy algorithm. Recall that setting A describes uniformly generated preference weights with log-normal prices (scale $\sigma = 1$), and setting B corresponds to log-normal preference weights ($\sigma = 1$) with log-normal prices ($\sigma = 2$).

in comparison to how our greedy procedures operate (augmenting inventory levels within restrictive assortments), this result is somewhat surprising, as one could expect that a more constrained decision space would limit the flexibility in constructing the stocking policy. However, the numerical results reported here provide empirical evidence that the structural restrictions imposed on the stocking policy are in fact beneficial, not only for purposes of analysis, but also in practical settings.

It is worth noting that, as the price and weight variabilities increase from setting A to setting B, the relative performance of the discrete-greedy and the Poisson-based dynamic program is negatively affected. On the other hand, our algorithm along with the deterministic relaxation, the normal approximation, and the local search algorithm have better robustness in the face of heterogeneity. Intuitively, in such settings, it is expected that near-optimal inventory vectors are concentrated over fewer products. Thus, it is not surprising that the deterministic relaxation and normal approximation turn out to be more accurate, as further corroborated by Table 5.2 below.

Random stock-outs vs. other models. Our experiments demonstrate the benefits of using a realistic modeling approach, that captures the stochastic nature of stock-out events, even though the resulting model is not solved optimally. The proportional scaling heuristic, based on the optimal static assortment (and used as a subroutine to compete against cheap products), has a rather satisfactory performance, which is not entirely surprising in light of the guarantees established in Section 5.3.3. However, its average revenue loss can be as large as 7.3%, thus supporting the value of jointly studying the assortment and inventory dynamics. As shown in Table 5.2, the model proposed by Topaloglu (2013) tends to be less accurate for larger capacity values as well as for larger prices and weight variabilities. These trends are more pronounced for the Poisson-based dynamic program, while the normal-based algorithm tends to be more robust. This observation suggests that the flexibility to vary the assortment over time provides greater value to the retailer in such regimes. Interestingly, the accuracy of the deterministic relaxation tends to vary in the opposite direction, as a function of the different parameters. Indeed, the quality of the approximation and the optimality gap often improve as we scale-up the different parameters. One possible intuitive explanation is that fluid approximation models become more relevant asymptotically, as well as more tractable.

Table 5.2: Average absolute approximation errors and optimality gap of the mixed integer program.

Model	Setting	C	PDP	NDP	DET	GAP
Poisson	A	25	15.9%	31.9%	5.2%	30.3%
		50	29.0%	81.1%	6.6%	27.9%
		100	68.5%	101.6%	1.2%	22.6%
	B	25	23.8%	36.9%	2.8%	16.1%
		50	81.2%	127%	2.8%	10.3%
		100	1118.4%	318.8%	6.1%	14.0%
Nonparametric	A	25	17.8%	12.7%	3.8%	52.9%
		50	19.2%	34.3%	2.7%	15%
		100	72.4%	47.5%	4.3%	12.9%
	B	25	21%	26.2%	5.1%	8.8%
		50	37.7%	48.9%	4.3%	12.8%
		100	81.3%	54.3%	5%	9.5%

Here, GAP refers to the MIP optimality gap for the deterministic relaxation algorithm. The other entries are the average absolute errors when using the optimal value of the approximate models (DP-P, NDP, and DET) as a proxy for the expected revenue obtained in our stochastic dynamic substitution model (by the same inventory vector).

Running time. As shown in Table 5.3, the normal approximation dynamic program emerges as the fastest algorithm tested followed by our algorithm. In theory, the former algorithm has a running time of $O(n \cdot \log(\frac{p_{\max}}{p_{\min}}))$, given the bisection search, while our algorithm runs in time $O(nC)$. On average over all configurations, both algorithms cut the running time by more than 50% against all other heuristics. The relative efficiency of our algorithm is mainly due to the restrictive greedy rules, which limit the actions examined prior to each incremental decision, in comparison to the local search and discrete-greedy heuristics.

The gradient-descent algorithm is particularly inefficient from a computational perspective, presumably due to the likely existence of local minima, where the algorithm progresses at a slower rate towards the final solution. Furthermore, due to its parameter dependency (step size and stopping criterion), the gradient-descent algorithm poses several implementation challenges. Even though we used here the best parameter values found by trial and error, it is still possible that fine-tuned parameters for each configuration could improve the performance in terms of optimality gaps and running times. Interestingly, as mentioned above, the optimality gap of the deterministic MIP shrinks when we increase the capacity or the price and weights variability, possibly since the combinatorial aspects are likely to be mitigated in an asymptotic regime, and since the LP relaxation becomes tighter.

Table 5.3: Average running time of the algorithms tested.

Parameters			**Average running time (sec.)**[*]						
Model	Setting	C	ALG	PDP	NDP	DET	LS	GD	DG
Poisson	A	25	99	316	57	926	702	831	239
		50	205	394	103	929	630	1000	595
		100	357	711	170	801	673	1000	971
	B	25	52	292	58	652	320	637	239
		50	91	514	107	434	256	1000	520
		100	203	795	167	271	247	1000	981
Nonparametric	A	25	126	241	44	1000	572	897	332
		50	307	417	93	973	415	1000	807
		100	667	682	165	792	688	1000	1000
	B	25	57	280	49	707	154	1000	324
		50	164	437	80	664	285	1000	765
		100	348	772	160	622	351	1000	1000

[*] Recall that every algorithm is being run with a time limit of 1000 seconds.

5.5 Concluding Remarks

Applications of restricted properties. To derive our main result, the analysis in Section 5.3.2 unravels hidden submodularity-like properties satisfied by the expected revenue function, and utilizes new notions of monotonicity and submodularity. One interesting direction for future research is to investigate whether these weaker properties could be used for closely related models in dynamic assortment planning, such as the Markov chain choice model (Blanchet et al. 2016), which generalizes MNL, or a mixture of Multinomial Logits (Bront et al. 2009, Méndez-Díaz et al. 2014, Rusmevichientong et al. 2014, Feldman and Topaloglu 2015) with fixed number of customer types.

Approximation guarantee without IFR. It would be interesting to determine whether a constant-factor approximation for the MNL-based dynamic assortment planning problem can be obtained under general (non-IFR) demand distributions. Here, we mention in passing that the methods developed in this chapter allow us to obtain an $O(\log n)$-approximation in this general setting, using an appropriate decomposition of the underlying set of products, combined with greedy procedures. The specifics of the resulting algorithm and its analysis are given in Appendix C.2.

Open questions. A natural direction for future research is that of obtaining improved approximation guarantees, which seems particularly challenging through the techniques developed in this chapter, specifically due to the optimality loss incurred by subadditivity-based bounds. Another important theoretical question is to establish hardness of approximation results for dynamic substitution models. In fact, it might be NP-hard to evaluate (even approximately) the expected revenue function at a given inventory vector. That being said, due to the stochastic nature of this problem, any complexity results along these lines would be very interesting to obtain.

Finally, given their greedy nature and scalability, the algorithms we present are applicable in a broad range of settings. These include, for instance, product-specific per-unit costs, general knapsack constraints for storage or display, matroid/extendibility constraints on the assortment offered, etc. For such settings (and combinations thereof), the only requirement is being able to efficiently solve the corresponding static formulation. However, in its current form, our worst-case analysis holds under a cardinality constraint, similar to previous analytical work on approximation algorithms for dynamic assortment planning (Goyal et al. 2016, Segev 2015). An interesting open question is that of devising provably-good algorithms for more general constraint structures, which seem to require further technical developments.

Chapter 6

Future Research

We conclude this thesis by briefly outlining leads for future research in the area of choice-based revenue management. We begin by describing several open technical problems and model extensions that arise from our current work, focusing on assortment and inventory optimization models. Next, we discuss a broader set of practical questions, which we believe are critical to the adoption of nonparametric choice models by practitioners.

Optimization problems. We summarize in what follows the main open problems and technical challenges that stem from this thesis:

- *Assortment optimization under uniform distributions.* As shown in Chapter 2, when the distribution over ranked lists is uniform, i.e., $\lambda_1 = \lambda_2 = \cdots = \lambda_K = 1/K$, the assortment optimization problem is APX-hard. A natural question is whether we can obtain a constant-factor approximation in this setting. This encoding of the assortment optimization problem might be useful in practice, since the number of samples K needed to obtain an accurate estimates of the revenue function (in an additive sense) grows as $O(p^2_{\max}/p^2_{\min}(n \log(n)))$, by standard concentration inequalities (see Rusmevichientong et al. (2006)).

- *Assortment optimization under conjunctive consideration set models.* The conjunctive model, where customers form their consideration sets through a sequence of eliminations in the feature space, is the most common decision process observed in empirical settings (Gilbride and Allenby 2004, Parkinson and Reilly 1979, Belonax and Mittelstaedt 1978, Laroche et al. 2003). In addition, this model is of interest since any collection of consideration sets can be replicated by some conjunctive process, if products are embedded in a feature space of large enough dimension. As such, it would be interesting to investigate the parametric complexity of assortment optimization in small dimensions, similar to the result obtained for disjunctive consideration set models, in Chapter 3.

- *Extension of dynamic assortment optimization models.* There are three practical extensions of the inventory optimization models discussed in Chapters 4 and 5: (i) knapsack capacity constraint: in order to capture storage space or working capital limitations, it would be interesting to study a general setting, where each product has a distinct marginal consumption of the capacity; (ii) newsvendor-like objective: instead of imposing a hard constraint on inventory, it would be more realistic in certain settings to introduce overage and underage costs; (iii) general demand distribution: at the exception of the nested choice model, most of our results rely on the Increasing Failure Rate property.

Estimation and optimal learning. The estimation of parametric and nonparametric choice models remains a fundamental challenge, both in theory and in practice. One initial step is to study the identifiability of nonparametric models, that is, to establish under which sufficient (or necessary) probabilistic conditions on the choice data and the distribution over ranked lists, the ground truth model can be uniquely recovered. Furthermore, this class of models has not been studied through the lens of learn-and-earn tradeoffs, to balance preference learning with revenue generation. In particular, it would be interesting to investigate how the "optimal" distribution sparsity grows as a function of the learning horizon. Finally, to the best of our knowledge, for most choice models used in revenue management, it is unknown whether we can obtain finite-sample estimators.

Nonparametric choice models with context information. While this thesis is concerned with operationalizing nonparametric choice models in decision problems, another challenge that currently limits the practical relevance of nonparametric models in predictive tasks, is the ability to leverage context information (e.g., customer-level covariates and product attributes). Indeed, in most applications, choice data offers little assortment variability whereas the context information varies greatly in the customer population, so that leveraging this side information is essential in order to learn detailed distributions over ranked lists. Also, in most categories, the number of relevant attributes is significantly smaller than the number of products. Thus, under limited data, it is critical to reduce the estimation task to the space of attributes.

Appendix A

Appendix of Chapter 3

A.1 Modeling the Consideration Sets

Proof of Lemma 3.4.7. To ease the exposition, we define i as the minimal product in S, and let v designate a customer-type in T. By definition, there exists a customer-type $u \in T$ such that $i \in C_u$. Also, since $G[S,T]$ is a connected subgraph,give there exists a path between v and u. We now define $v^* \in T$ as the customer-type in T which satisfies $i \in C_{v^*}$ and has the shortest path with v. In other terms, v^* minimizes the length of a path between v and x over all $x \in T$ such that $i \in C_x$. This set is not empty because it contains customer-type u. We are going to prove that $C_v \cap [i] \subset C_{v^*} \cap [i]$.

Let j_1, j_2,... j_l be sequence of customer-type nodes in $G[S,T]$ corresponding to the shortest path between v and v^*:

$$\begin{cases} j_1 = v^* \text{ and } j_l = v \\ \forall r \in [l-1],\ \exists a \in S \ \text{ s.t. } (j_r, a, j_{r+1}) \text{ is a path of } G[S,T] \end{cases}$$

Let a_1, a_2...,a_{l-1} be the corresponding sequence of maximal intersections of the consideration set of each two subsequent customer-types along this path:

$$\forall 2 \leq r \leq l,\ a_r = \max[C_{j_r} \cap C_{j_{r-1}}]$$

By convention, we set $a_1 := i$. We now prove by induction over r, $2 \leq r \leq l$, that $a_r > a_{r-1}$ and $C_{j_r} \cap [a_r] \subset C_{j_{r-1}} \cap [a_r]$.

- *Base case ($r = 2$).* We first note that $a_1 < a_2$. Indeed, since i is the minimal element of S, we can infer that $a_2 \geq i$. These indices can not be equal otherwise $i \in C_{j_2}$ and we would obtain a strictly shorter path between v and j_2 by considering the path $(j_2, a_3, \ldots, a_l, j_l)$ and this contradicts the minimality of l. We now prove the inclusion. We infer from the definition of weakly laminar consideration sets that either $C_{j_1} \cap [a_2] \subset C_{j_2} \cap [a_2]$ or $C_{j_2} \cap [a_2] \subset C_{j_1} \cap [a_2]$. In addition, item i is contained in C_{j_1} and $i \notin C_{j_2}$, otherwise it would contradict the minimality of the path. Since $i \notin C_{j_2}$, we can infer that $C_{j_2} \cap [a_2] \subset C_{j_1} \cap [a_2]$, which leads to the desired result.

- Inductive step $r > 2$. We begin by assuming that $a_r > a_{r-1}$. Again, by definition, either $C_{j_r} \cap [a_r] \subset C_{j_{r-1}} \cap [a_r]$ or $C_{j_{r-1}} \cap [a_r] \subset C_{j_r} \cap [a_r]$. We assume that the latter is satisfied to prove a contradiction. Since we assume that $a_r > a_{r-1}$, the latter set inclusion leads to $a_{r-1} \in C_{j_r}$. Therefore, $C_{j_{r-2}}$ and C_{j_r} both contain product a_{r-1} and (j_{r-2}, a_{r-1}, j_r) is a path of $G[S, T]$. Thus, we can obtain a path between v^* and v of strictly smaller length using the shortcut (j_{r-2}, a_{r-1}, j_r) instead of $(j_{r-2}, a_{r-1}, j_{r-1}, a_r, j_r)$. However, this would contradict the minimality of l. Thus: $C_{j_r} \cap [a_r] \subset C_{j_{r-1}} \cap [a_r]$.

In order to prove the above assumption that $a_r > a_{r-1}$, we now assume that $a_r \leq a_{r-1}$ and prove that it leads to a contradiction. By the induction hypothesis, we know that $C_{j_{r-1}} \cap [a_{r-1}] \subset C_{j_{r-2}} \cap [a_{r-1}]$. Thus, if $a_r \leq a_{r-1}$, it follows that $a_r \in C_{j_{r-1}} \cap [a_{r-1}]$. From the above inclusion, we obtain that $a_r \in C_{j_{r-2}}$. Therefore, there is an edge between j_{r-2} and a_r and $(j_1, a_2, j_2, \ldots, j_{r-2}, a_r, j_r, \ldots, j_l)$ would form a path between v^* and v of strictly smaller length, which contradicts the minimality of l. We can thus obtain that $a_r > a_{r-1}$.

So far, for any given $v \in T$, we have proven the existence of $v^* \in T$ such that $C_v \cap [i] \subset C_{v^*} \cap [i]$ and $i \in C_{v^*}$. Defining $T(i)$ as the subset of customer-types in T that consider product i, we may verify that the collection of subsets $C_j \cap [i]$ where

$j \in T(i)$ is nested. As a consequence, this collection admits a maximal element, that corresponds to a customer-type $j^* \in T$. Thus, we conclude that $C_v \subseteq C_{j^*}$ for any $v \in T$. ■

Proof of Theorem 3.4.9. The proof is analogous to the previously considered models. We seek to upper-bound the quantity $|\Phi\langle\mathcal{S}\rangle|$. To this end, we let (S,T) be a subproblem of $\mathcal{S}$. We have

$$\begin{aligned}
\Phi(S,T) &= [\min(S)] \bigcap \left(\bigcup_{j\in T} C_j\right) \\
&= [\min(S)] \bigcap \left(\bigcup_{j\in T}\bigcup_{e\in[d]} \left\{i \in [n] : x_e^{(j)} \geq t_e^{(j)}\right\}\right) \\
&= [\min(S)] \bigcap \left(\bigcup_{e\in[d]}\bigcup_{j\in T} \left\{i \in [n] : x_e^{(i)} \geq t_e^{(j)}\right\}\right) \\
&= [\min(S)] \bigcap \left(\bigcup_{e\in[d]} \left\{i \in [n] : x_e^{(i)} \geq \min_{j\in T} t_e^{(j)}\right\}\right) ,
\end{aligned}$$

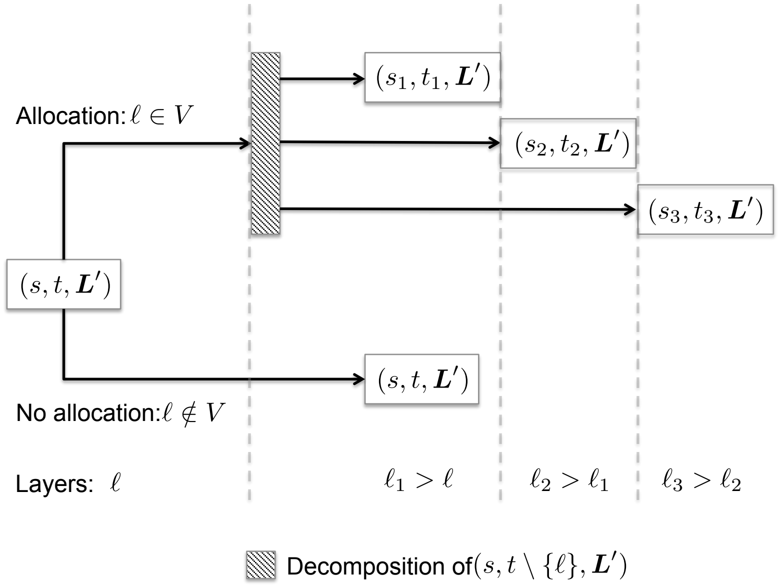
where the second equality follows from Definition 3.4.8, and the third equality proceeds by changing the union order. We conclude by observing that for each $e \in [d]$, the quantity $\min_{j\in T} t_e^{(j)}$ can take at most K distinct values. Therefore, we obtain that $|\Phi\langle\mathcal{S}\rangle| \leq n \cdot K^d$. ■

A.2 Marginalized Recursion

We give here the specifics of the marginalization algorithm described in Section 3.5.

Informal sketch. By constructing and updating an appropriate data-structure, denoted by $\mathcal{D}(S,T,\boldsymbol{L}) \sim \mathcal{D}$, we prevent the redundant exploration of the children subproblems appearing in equation (3.4). Specifically, we construct recursively a directed graph $\mathcal{D}$, illustrated by Figure (A-1). To this end, each node inserted in $\mathcal{D}$ is labelled by a combination of a child subproblem, and the index of the last customer-type in $T(i)$ that has been processed, termed the *layer* of the node. At each step, we consider all unmarked nodes, and process their next customer-type in $T(i)$ according to the increasing index order. The dynamic program decides whether the current customer-type is allocated to product i or not. Each decision entails a graph decomposition into children subproblems according to Lemma 3.5.1. The corresponding nodes, with the respective customer-type layer, are inserted in $\mathcal{D}$ as unmarked nodes. Also, we add directed edges connecting the father node to its respective children nodes. The procedure terminates when it attains the maximal layer index.

Figure A-1: Recursion step of the marginalized dynamic program.


Allocation: $\ell \in V$
$(s_1, t_1, \boldsymbol{L}')$
$(s_2, t_2, \boldsymbol{L}')$
$(s_3, t_3, \boldsymbol{L}')$
$(s, t, \boldsymbol{L}')$
$(s, t, \boldsymbol{L}')$
No allocation: $\ell \notin V$
Layers: ℓ
$\ell_1 > \ell$
$\ell_2 > \ell_1$
$\ell_3 > \ell_2$
▨ Decomposition of $(s, t \setminus \{\ell\}, \boldsymbol{L}')$

Generation of the computational tree. More formally, we assume that the customer-types $T(i)$ are reindexed in an arbitrary order $T(i) \sim [l]$ where $l = |T(i)|$. We introduce a directed graph data-structure $\mathcal{D}(S, T, \boldsymbol{L})$, initially set empty. (In the following, unless ambiguity arises, it is simply denoted $\mathcal{D}$ for ease of exposition.) Each node we add to $\mathcal{D}$ is uniquely labelled by a tuple $(j, s, t) \in [l] \times \mathcal{P}(S) \times \mathcal{P}(T)$ where $(s, t, \boldsymbol{L}')$ is a child subproblem appearing in equation (3.4) and generated by an allocation contained in $[j]$. The nodes are generated by an iterative procedure described below:

- *Base case.* We start with an empty graph $\mathcal{D} \leftarrow \emptyset$. The first nodes that we add correspond to the empty allocation $V = \emptyset$. Namely, for each connected components $G_{\boldsymbol{L}'}[S_u^{(\emptyset)}, T_u^{(\emptyset)}]$, we insert a node in $\mathcal{D}$ labelled $(S_u^{(\emptyset)} T_u^{(\emptyset)}, 0)$. We refer to them as the roots of $\mathcal{D}$.

- *Recursive step.* Assume that a node with label (s, t, j) has been added to $\mathcal{D}$. The next customer-type we consider, denoted j', is the minimum of $t \cap [j+1, l]$. The decision made at this stage is whether customer-type j' gets allocated to product i or not. In the latter case, a node (s, t, j') is inserted in $\mathcal{D}$ unless it already belongs to the data-structure. Also, we create a directed edge between the parent node labelled (s, t, j) and its descendant (s, t, j'). Conversely, in case j' is allocated to product i, we derive the residual graph $G_{\boldsymbol{L}'}[s, t \setminus \{j'\}]$ and compute its connected components. Each connected component $G_{\boldsymbol{L}'}[s_u, t_u]$ leads to the insertion of a new node (s_u, t_u, j') unless it already belongs to $\mathcal{D}$. Also, directed edges are added between the parent node and its descendants in $\mathcal{D}$.

The graph $\mathcal{D}$ built via this recursive procedure is a directed forest – a cycle-free directed graph. Indeed, the only edges are between father nodes and their offspring. Because the customer-type index in the node label is monotonic ($j' > j$), there cannot be any cycle. Finally, we observe that the leafs of $\mathcal{D}$ uniquely represent all subproblems generated by the allocations $V \subseteq T(i)$. Indeed, any V corresponds to a sequence of binary decisions in $[l]$. This sequence of decisions defines a collection of paths in $\mathcal{D}$ starting from the root nodes. By construction, the subproblems described by the labels of the terminating leafs are exactly the subproblems generated by V.

In terms of running time, each distinct subproblem shows up in at most l nodes of $\mathcal{D}$ (and l is smaller than K). Therefore, the total running time to generate the DP computational tree is upper bounded by $O(nK^2 \cdot |\mathcal{S}|)$.

Solving equation (3.4)**.** Once the DP computational tree has been drawn, the subproblems are solved backwards using the recursive formula (3.4). By exploiting the data-structure $\mathcal{D}(S, T, \boldsymbol{L})$, we show in this paragraph that the maximization problem (3.4) can be recast as a low dimensional dynamic program that can be solved efficiently. That is, at each recursive step of the master dynamic program, we solve a separate dynamic program, termed the *marginalized dynamic program.*

We consider a fixed instance $(S, T, \boldsymbol{L})$. Suppose that all subsequent subproblems have been solved as we move backwards over the computational tree. For ease of exposition, the reference to the parameters $(S, T, \boldsymbol{L})$ is omitted when introducing the marginalized dynamic program, and the notations $i, T(i), \boldsymbol{L}', l$ and $\mathcal{D}$ are consistent with the previous definitions.

By construction, we note that for each node of $\mathcal{D}$, with label $q = (s, t, j)$, the corresponding subproblem $(s, t, \boldsymbol{L}')$ has been generated by at least one allocation $V \subseteq [j]$, that we designate as $V(q)$. We define the value function $F(q)$ as the optimal expected revenue from customer-types t in the subproblem $(S, T, \boldsymbol{L})$ under the constraints that *(i)* product i is stocked and *(ii)* the allocation of this product $V \subseteq T(i)$ satisfies the constraint $V \cap [j] = V(q)$, i.e., the projection of V on $[j]$ is $V(q)$. Let j' be the next customer-type for which a decision is made when examining node q, i.e., $j' = \min([j+1, l] \cap T)$. Letting $\mathcal{N}(q)$ denote the children nodes of q if j' is allocated to product i and q' be the child node of q otherwise, we obtain:

$$F(q) = \max \left(F\left(q'\right) \ , \ \lambda_{j'} \cdot P_i + \sum_{u \in \mathcal{N}(q)} F\left(u\right) \right)$$

Indeed, if customer-type j' is allocated to product i, it generate a revenue of $\lambda_{j'} \cdot P_i$ and the residual graph decomposes into the connected subgraphs described by $\mathcal{N}(q)$. Conversely, if j' is not allocated to product i, the connected subgraph is not modified further and the revenue is that of $F(q')$. This is consistent with the constraint *(ii)* as $V(q') = V(q)$ when j' is not added to the allocation.

By applying this formula inductively from the leafs of $\mathcal{D}$, we compute $F(q_1)$,...,$F(q_{r(\emptyset)})$ where q_1,...,$q_{r(\emptyset)}$ are the root nodes of $\mathcal{D}$. Conditional on the fact that i is stocked, we conclude that:

$$J(S,T,\boldsymbol{L}) = \sum_{u=1}^{r(\emptyset)} F(q_u)$$

Therefore, equation (3.4) is equivalent to:

$$J(S,T,\boldsymbol{L}) \;=\; \max\left(\sum_{u=1}^{r(-)} J(S_u^-, T_u^-, \boldsymbol{L}) \;,\; \max\left[\sum_{u=1}^{r(\emptyset)} F(q_u)\right]\right)$$

Example with the in-tree model. To flesh out our marginalized algorithm through a concrete model, we argue now that this approach allows to solve efficiently the *in-tree* model proposed by Honhon et al. (2012). Here, each product is represented by a node in a rooted tree $\mathcal{T}$. Each consideration set in $\mathcal{C}$ corresponds to a path from the root to a given node – we will denote by C_v the consideration set formed by the path from the root to node $v \in \mathcal{T}$. We further assume that such directed paths define the increasing preference order, namely, the farther from the root, the more preferred is a product, thus leading to some (non-unique) ranking function σ. The processing order is chosen as the reverse permutation $\bar{\sigma}$, that is, products are processed from the root to the descendant nodes. To argue that the marginalization is efficient, it is sufficient to show that, for any product $i \in [n]$, we can restrict attention to allocations $V \subseteq T(i)$ corresponding to subtrees of product nodes (here, without ambiguity, we can mix each customer-type with his corresponding consideration set and its left-most product node in $\mathcal{T}$). To arrive at a contradiction, suppose we have an optimal allocation $V \subseteq T(i)$ with $C_i, C_j \in V$, where j is a descendent of i, and k is on the path from i to j although $C_k \notin V$. Since product i has been allocated to the customer-type C_j, who prefers product k over i according to σ, it follows that product k is not contained in $S^{(V)}$. Consequently, all product nodes between j and k have been eliminated from the residual graph, and therefore the customer-type node of C_k is disconnected from any (non-trivial) connected component. Thus, we can assume without loss of generality that $C_k \in V$.

Complexity Analysis. We now derive a general upper bound on the running time.

Proof of Proposition 3.5.2. The proof of the first claim follows from our previous observations. At each node $(S, T, \boldsymbol{L})$ of the computational tree, the running time for generating the graph $\mathcal{D}(S, T, \boldsymbol{L})$ along with the running time for solving the marginalized DP is at most $O(n \cdot K^2 \cdot |\mathcal{S}|)$. Summing over all the nodes of the DP computational tree, we obtain a total running time of $O(n \cdot K^2 \cdot |\mathcal{S}|^2)$.

We now derive an upper-bound on the state space size. To this end, we construct a function Φ that maps any subproblem generated along the recursion to a subset of products as well as a product within this set. By definition, any subproblem $(S, T, \boldsymbol{L}) \in \mathcal{S}$ has been generated by a sequence of decisions whereby some products in $\{1, \ldots, \min(S) - 1\}$ have been stocked. We define $\mathcal{A}(S, T, \boldsymbol{L})$ as a partial assortment of products corresponding to a sequence of decisions prior to generating subproblem $(S, T, \boldsymbol{L})$. The mapping is described as follows:

$$\begin{aligned}\Phi : \mathcal{S} &\to \mathcal{P}([n]) \times [n] \\ (S, T, \boldsymbol{L}) &\mapsto (\mathcal{A}(S, T, \boldsymbol{L}) \cup S, \min(S))\end{aligned}$$

It is sufficient to show that this function is injective to obtain the desired result. Assume that two subproblems satisfy $\Phi(\mathcal{I}_1) = \Phi(\mathcal{I}_2)$ where $\mathcal{I}_1 = (S_1, T_1, \boldsymbol{L}_1)$ and $\mathcal{I}_2 = (S_2, T_2, \boldsymbol{L}')$. Then, by definition:

$$\begin{aligned}S_1 &= \Phi^{(1)}(\mathcal{I}_1) \cap [\Phi^{(2)}(\mathcal{I}_1), n] \\ &= \Phi^{(1)}(\mathcal{I}_2) \cap [\Phi^{(2)}(\mathcal{I}_2), n] \\ &= S_2 \ .\end{aligned}$$

This proves that the two subproblems have the same subsets of products. By similar observations, we can claim that both $\mathcal{I}_1$ and $\mathcal{I}_2$ are generated by the same sequence of decisions, or equivalently the same assortment $\mathcal{A} \subseteq \Phi^{(1)}(\mathcal{I}_1) \cap [\Phi^{(2)}(\mathcal{I}_1) - 1]$. We also know that $\boldsymbol{L} = \boldsymbol{L}'$ because the truncation vector is determined by the previous stocking decisions. As a result, the only difference between the two subproblems could only be caused by a different sequence of allocations. Therefore, it is sufficient to prove that the set of customer-types remaining in the two connected subgraphs are exactly the same in order to obtain that $\mathcal{I}_1 = \mathcal{I}_2$. Ad absurdum, assume $j \in T_1 \setminus T_2$. Because $S_1 = S_2$, this means that customer-type j is still unsatisfied in $\mathcal{I}_1$ whereas it was allocated to a product in the sequence of decisions that generates the subproblem $\mathcal{I}_2$. Since j has been satisfied along the generation of the subproblem $\mathcal{I}_2$, there exists a product i in $\mathcal{A}$ that belongs to C_j. In addition, since $G_{\boldsymbol{L}}[S_1, T_1]$ is a connected subgraph, this means there exists an edge between node j and a product node $i' \in S_1$. Along the generation of subproblem $\mathcal{I}_1$, i has been made available to j but it was not allocated to customer-type j - as a result its consideration set has been truncated to only account for products more preferred than i: $L_j < \sigma_j(i)$. Thus customer j necessarily prefers i' over product i. On the other hand, as product i was allocated to customer-type j when generating $\mathcal{I}_2$, the product i' has been deleted because he prefers it over i. Thus $i' \notin S_2$ and since $S_1 = S_2$, we obtain a contradiction: $i' \notin S_1$. ■

A.3 Proof of Theorem 3.6.1

We construct a function Ψ that maps any subproblem generated along the recursion to a tuple that lies in a space of size $2^{2d-2}h$. By showing that Ψ is injective, we obtain the upper bound on the size of the state space.

Specifically, assuming that $(S, T, \boldsymbol{L}) \in \mathcal{S}$, we define i as the next product to be processed in S and $\mathcal{A}$ corresponds to the assortment decisions in $[i-1]$ which generate this subproblem. The image of $(S, T, \boldsymbol{L})$ by Ψ is defined as the tuple $(S_0, T_0, \boldsymbol{x}, \boldsymbol{y})$ where:

- (S_0, T_0) is the subproblem of the unique-ranking dynamic program generated by the sequence of stocking decisions $\mathcal{A} \cap [i - 2d]$ over $[i-1]$ and such that $T \cap T_0 \neq \emptyset$,
- $\boldsymbol{x} = \mathcal{A} \cap [i - 2d + 1, i - 1]$.
- and $\boldsymbol{y} = S \cap [i + 1, i + 2d - 1]$.

To prove that this mapping is injective, we show that each subproblem $(S, T, \boldsymbol{L})$ is uniquely determined by the tuple $(S_0, T_0, \boldsymbol{x}, \boldsymbol{y})$.

We begin by remarking that all preference lists j in T do not intersect with $\mathcal{A} \cap [i - 2d]$. Otherwise, we define α as an arbitrary product of the intersection of $\mathcal{A} \cap [i - 2d]$ with C_j. Then, by construction, $L_j \leq \sigma_j(\alpha)$. In addition, given that $\sigma_j \in B(\sigma, d)$, any product in $[i - 2d]$ is preferred over products in S (recall that the products are numbered according to the central permutation σ, meaning that σ is the identity). As a result, $\sigma_j(\alpha) < \sigma_j(\beta)$ for all product $\beta \in S$, meaning that $C_j(L_j)$ does not intersect with S which contradicts the connectivity of the subgraph $G_{\boldsymbol{L}}[S, T]$.

Uniqueness of $\boldsymbol{L}$. We now argue that $\boldsymbol{L}$ is uniquely determined by $\boldsymbol{x}$. Indeed, if $j \in T$, then by the above remark, C_j does not intersect with the projection of $\mathcal{A}$ on $[i - 2d]$ and as result L_j is a deterministic function of $\boldsymbol{x}$:

$$L_j = \min\{\sigma_j(\alpha) : \alpha \in \boldsymbol{x}\}$$

Uniqueness of T. We first show that $T \subseteq T_0$. Assume that $j \in T$. Using the above remark again, we infer that j is not satisfied and eliminated by the decisions of stocking $\mathcal{A} \cap [i - 2d]$ in the unique-ranking DP. As a result, T is included in the set of customer-type nodes of the residual graph obtained in the unique-ranking DP after performing the graph operations associated with the sequence of stocking decisions $\mathcal{A} \cap [i - 2d]$. Thus, it is sufficient to verify that T lies in a connected component of the residual graph in order to prove that $T \subseteq T_0$. The key observation is that the residual graph obtained by the stocking decisions of $\mathcal{A}$ under the general DP is a subgraph of the residual graph generated by the decisions of stocking $\mathcal{A} \cap [i - 2d]$ in the unique-ranking DP. Indeed, the graph operations performed by the unique-ranking algorithm are also performed at some step of the recursion of the general DP:

- Customer-type node deletions: by the above remark, any preference list that is discarded as a result of the stocking decisions $\mathcal{A} \cap [i - 2d]$ in the unique-ranking DP is also discarded at some point of the decision sequence associated with $\mathcal{A}$

in the general DP.

- Product node deletions: in the unique-ranking case, a product node is deleted when it is processed. Given that the two algorithms follow the same processing order, any product deleted in the unique-ranking case has also been deleted in the general DP.

Therefore, because $(S, T, \boldsymbol{L})$ is connected in the residual graph of the general DP, it is also connected in the residual graph of the unique-ranking DP. Thus, $T \subseteq T_0$ and what remains to be proven is the uniqueness of $T_0 \setminus T$ conditional to $\Psi(S, T, \boldsymbol{L})$. In fact, it is immediate that $T_0 \setminus T$ corresponds to the subset of preference lists nodes that get deleted (or disconnected) due to the allocation of products $\boldsymbol{x} = \mathcal{A} \cap [i-2d+1, i-1]$. Let j be a preference list of T_0 that satisfies $\boldsymbol{x} \cap C_j \neq \emptyset$ while customer-type j still belongs to T, meaning that j has not been satisfied with any product of $\boldsymbol{x}$. Let $\alpha \in \boldsymbol{x}$ be the most preferred product of customer-type j in $\boldsymbol{x}$. Then, there necessarily exists at least one product in $S \cap C_j$ preferred over product α otherwise $C_j(L_j) \cap S = \emptyset$ and the subgraph would not be connected. In fact, because the preference rankings σ_j are contained in $B(\sigma, d)$, this product is at distance at most $2d$ from α, meaning that it belongs to $\boldsymbol{y}$. Reciprocally, if there exists a product in $\boldsymbol{y} \cap C_j$ preferred over α, then it follows that j has not been allocated to any product of $\boldsymbol{x}$. In particular, since $C_j(L_j) \cap S \neq \emptyset$, the customer-type node j is connected with S and thus $j \in T$. This shows that $T_0 \setminus T$ is uniquely determined as a function of the subsets $\boldsymbol{x}$ and $\boldsymbol{y}$, which proves the desired result.

Uniqueness of S. In general, we have:

$$S = \bigcup_{u \in T} C_j(L_j) \ .$$

It immediately follows that the uniqueness of S can be inferred from the uniqueness of T and $\boldsymbol{L}$.

A.4 Quasi-convex Preference Lists

Claim A.4.1. *For a fixed central permutation, there exists* $2^{n+1} - n - 2$ *quasi-convex preference lists.*

Proof. Let $\Sigma(n)$ be the set of quasi-convex preference lists over n products. The preference lists are uniquely defined by their consideration set and the quasi-convex ranking function. For any fixed interval of length $\ell \in [n]$, the ranking function can be *viewed* as a permutation over ℓ elements: $[\ell] \to [\ell]$. We now construct a mapping ϕ from any subset $S \subset [2, \ell]$ to a quasi-convex permutation over the interval $[\ell]$. $\phi(S)$ is defined as follows:

$$\begin{cases} \phi(S) \text{ is decreasing over } [|S|] & \text{with } \ \phi(S)\langle|[|S|]\rangle = S \\ \phi(S) \text{ is increasing over } [|S|+1, n] & \text{with } \ \Phi(S)\langle[|S|+1, n]\rangle = [\ell] \setminus S \end{cases}$$

Indeed, the quasi-convex permutation $\phi(S)$ is uniquely defined given its monotonicity on each interval. It can be verified that this mapping is surjective (by taking S equal to the set of image values of the quasi-convex permutation on its decreasing interval excluding the minimal value 1). Finally, it is injective by observing that if $\phi(S_1)$ and $\phi(S_2)$ are equal, in particular they share the same decreasing segments and $S_1 = S_2$. Therefore, the cardinality of quasi-convex ranking functions over an interval of length ℓ is $2^{\ell-1}$. By remarking that there exists $n - \ell + 1$ distinct intervals of length ℓ, we obtain:

$$\begin{aligned} |\Sigma(n)| &= \sum_{\ell=1}^{n} (n - \ell + 1) \cdot 2^{\ell-1} \\ &= (n+1) \cdot \sum_{\ell=0}^{n-1} 2^{\ell} - \sum_{\ell=1}^{n} \ell \cdot 2^{\ell-1} \\ &= (n+1) \cdot (2^n - 1) - (n-1) \cdot 2^n - 1 \\ &= 2^{n+1} - n - 2 \ . \end{aligned}$$

□

A.5 Proof of Theorem 3.6.5

The processing order can be chosen as an arbitrary permutation - but to fix ideas, we process the products in the increasing index order.

We incorporate to the model a 'dummy' preference list denoted by index 0 associated with the null utility vector $\boldsymbol{u}_0 = \vec{0}$ as well as a 'dummy' product corresponding to $\boldsymbol{x}_0 = \vec{0}$. Also, note that in what follows, by abuse of language, a product may refer to the corresponding graph node or its representation in the feature space.

Inductive hypothesis. We prove that each DP subproblem is characterized by a polytope in the feature space defined by a constant number of facets, chosen among a polynomial set of affine constraints. Specifically, we prove the following property inductively. Suppose that $(S, T, \boldsymbol{L})$ is generated along the recursion. Then, $G_{\boldsymbol{L}}[S, T]$ is the connected component of $G_{\boldsymbol{L}}[S', T']$ that contains product node $i = \min(S)$, where:

- There exists $(a, b, c, d) \in [n]^2 \times [K]^2$ such that S' is defined as follows:

$$\begin{cases} \boldsymbol{z} = Rot\left(\frac{\pi}{2}, \boldsymbol{x}_b - \boldsymbol{x}_a\right) \ , \\ H(a,b,c,d) = \left\{\boldsymbol{x} \in \mathbb{R}^2 : \boldsymbol{x} \cdot \boldsymbol{u}_c \leq \boldsymbol{u}_c \cdot \boldsymbol{x}_a, \ \boldsymbol{x} \cdot \boldsymbol{u}_d \leq \boldsymbol{u}_d \cdot \boldsymbol{x}_b, \ \boldsymbol{x} \cdot \boldsymbol{z} \geq \boldsymbol{x}_a \cdot \boldsymbol{z}\right\} \ , \\ S' = \{j \in [i, n] : x_j \in H(a,b,c,d) \setminus \partial H(a,b,c,d)\} \ . \end{cases}$$

- The set T' is formed by all customer-types whose utility vector belongs to the cone with extreme rays $(\boldsymbol{u}_c, \boldsymbol{u}_d)$ (where the rays are ordered in the anti-trigonometric order):

$$T' = \left\{j \in [K] : \exists \lambda_1, \lambda_2 > 0 \text{ s.t. } \boldsymbol{u}^{(j)} = \lambda_1 \boldsymbol{u}_c + \lambda_2 \boldsymbol{u}_d\right\} \ .$$

- The truncation vector $\boldsymbol{L}$ is given by:

$$\forall j \in T', \ L_j = \min(\sigma_j(a), \sigma_j(b)) \ .$$

Before proving this result, observe that this property implies that there exists an injective mapping from the state space $\mathcal{S}$ to the space of 5-tuples, described by a 3-tuple of products and a pair of customer-types. As a result, we conclude that $|\mathcal{S}| = O(K^2 n^3)$.

Base case. If $(S, T, \boldsymbol{L})$ is one of the roots of the DP computational tree, no products has been examined yet and $G_{\boldsymbol{L}}[S, T]$ is a connected component of G. Then, we set $a = b = c = d = 0$ and $i = \min(S)$. The polyhedron $H(a, b, c, d)$ describes to the entire space $\mathbb{R}^2$, $S' = [n]$ and $T' = [K]$. The above property is immediately satisfied.

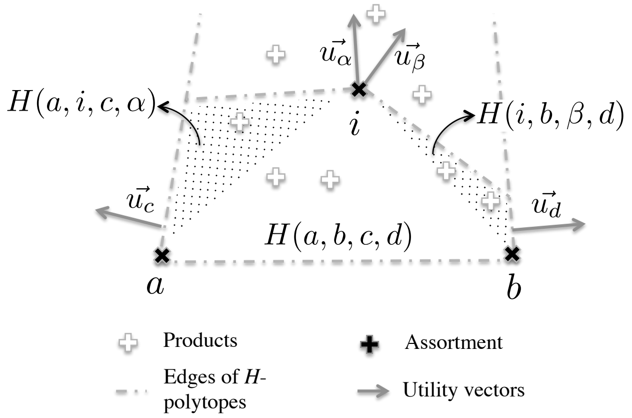
Recursive step. We assume that $(S, T, \boldsymbol{L})$ satisfies the above properties with respect to the tuple (a, b, c, d) and $(S', T', \boldsymbol{L})$. Let $i = \min(S)$ be the next item to be examined.

If i is not stocked in the assortment, we only need to discard product node i from the graph and compute the connected components of $G_{\boldsymbol{L}}[S \setminus \{i\}, T]$ to obtain the children subproblems. Equivalently, by the induction hypothesis, any child subproblem is a connected component of $G_{\boldsymbol{L}}[S' \setminus \{i\}, T'])$ and the above property is satisfied.

On the other hand, if the product i is allocated to a subset of customer-types $V \subseteq T(i)$. We define α, β as the indices corresponding to the extreme lines of the cone $conv\{\boldsymbol{u}_h : h \in V\}$ - such that $\boldsymbol{u}_\alpha, \boldsymbol{u}_\beta$ are ordered in the anti-trigonometric order. Using previous notation, any subproblem generated by the allocation V is a connected component of $(S^{(V)}, T^{(V)}, \boldsymbol{L}')$. We prove that it satisfies the desired inductive property, either with respect to the parameters (a, i, c, α), or with respect to (i, b, β, d), as illustrated by Figure A-2. In what follows, note that, by abuse of language, utility vectors refer indifferently to vectors in the feature space, and to the corresponding customer-types.

We begin by proving that the allocation V necessarily corresponds to the cone of utility vectors $(\boldsymbol{u}_\alpha, \boldsymbol{u}_\beta)$, meaning that $T^{(V)}$ is either contained in the cone $(\boldsymbol{u}_c, \boldsymbol{u}_\alpha)$ or in the cone $(\boldsymbol{u}_\beta, \boldsymbol{u}_d)$. To this end, assume that a preference list $j \in T'$ has its utility vector included in the cone $(\boldsymbol{u}_\alpha, \boldsymbol{u}_\beta)$, we seek to prove that $j \notin T^{(V)}$. This preference

Figure A-2: Recursive step: the allocation of product i to the cone $(\boldsymbol{u}_\alpha, \boldsymbol{u}_\beta)$ gives rise to independent subproblems, either contained in the polyhedra $H(a, i, c, \alpha)$, or $H(i, b, \beta, d)$.



list would only pick a product whose scalar product is greater than $\boldsymbol{x}_i \cdot \boldsymbol{u}^{(j)}$. Indeed, either product i lies the consideration set C_j, thus customer-type j would only pick a product preferred over i, or it does not belong to C_j and any product it chooses has its scalar product larger than the customer's cut-off level. Given that customer-types α, β are both satisfied with product i, then, all products which satisfy $\boldsymbol{x} \cdot \boldsymbol{u}_\alpha \geq \boldsymbol{x}_i \cdot \boldsymbol{u}_\alpha$ or $\boldsymbol{x} \cdot \boldsymbol{u}_\beta \geq \boldsymbol{x}_i \cdot \boldsymbol{u}_\beta$ have been removed. By Farkas lemma, since $\boldsymbol{u}^{(j)}$ is contained in the cone $(\boldsymbol{u}_\alpha, \boldsymbol{u}_\beta)$, then any product x which satisfies $\boldsymbol{x} \cdot \boldsymbol{u}^{(j)} \geq \boldsymbol{x}_i \cdot \boldsymbol{u}^{(j)}$ would satisfy either $\boldsymbol{x} \cdot \boldsymbol{u}_\alpha \geq \boldsymbol{x}_i \cdot \boldsymbol{u}_\alpha$ or $\boldsymbol{x} \cdot \boldsymbol{u}_\beta \geq \boldsymbol{x}_i \cdot \boldsymbol{u}_\beta$. As a result, customer-type j does not prefer any product of $S^{(V)}$ over i, meaning that customer-type node j is disconnected from $S^{(V)}$. Hence, without loss of generality, we may assume that $j \notin T^{(V)}$.

We now prove that the products of $S^{(V)}$ either lie in the polyhedron $H(a, i, c, \alpha)$, or within $H(i, b, \beta, d)$. Since all products that the customer-types α and β prefer over i have been discarded, we already know that $S^{(V)}$ is contained in the set $\bar{H}$:

$$\bar{H} = H(a, b, c, d) \setminus \left(\left\{ \boldsymbol{x} \in \mathbb{R}^2 : \boldsymbol{x} \cdot \boldsymbol{u}_\alpha \geq \boldsymbol{x}_i \cdot \boldsymbol{u}_\alpha \right\} \bigcup \left\{ \boldsymbol{x} \in \mathbb{R}^2 : \boldsymbol{x} \cdot \boldsymbol{u}_\beta \geq \boldsymbol{x}_i \cdot \boldsymbol{u}_\beta \right\} \right)$$

Let $j \in T^{(V)}$ designate a preference list in the residual graph. Since a, b and i are stocked in the assortment, the products contained in the truncated consideration set $C_j(L'_j)$ necessarily have a scalar product with $\boldsymbol{u}^{(j)}$ greater than the following quantities: $\boldsymbol{x}_a \cdot \boldsymbol{u}^{(j)}$, $\boldsymbol{x}_i \cdot \boldsymbol{u}^{(j)}$ and $\boldsymbol{x}_b \cdot \boldsymbol{u}^{(j)}$. Equivalently, they lie in the affine half-space defined by $\boldsymbol{x} \cdot \boldsymbol{u}^{(j)} \geq y$, where:

$$y = \max\left(\left\{\boldsymbol{x}_a \cdot \boldsymbol{u}^{(j)}, \boldsymbol{x}_b \cdot \boldsymbol{u}^{(j)}, \boldsymbol{x}_i \cdot \boldsymbol{u}^{(j)}\right\}\right) \ .$$

By construction of the polyhedron $H(a, b, c, d)$, if $\boldsymbol{u}^{(j)} \in (\boldsymbol{u}_c, \boldsymbol{u}_\alpha)$, then customer j's most preferred product among $\{a, i, b\}$ is either a or i. Also, the intersection of $\bar{H}$ with the half-space $\{\boldsymbol{x} : \boldsymbol{x} \cdot \boldsymbol{u}^{(j)} \geq y\}$ is included in $H(a, i, c, \alpha)$ and it has no product in common with $H(i, b, \beta, d)$. Therefore $C_j(L'_j)$ is included in $H(a, i, c, \alpha)$ and it does not contain any product in $H(i, b, \beta, d)$. Conversely, if $\boldsymbol{u}^{(j)} \in (\boldsymbol{u}_\beta, \boldsymbol{u}_d)$, then customer j's most preferred product among $\{a, i, b\}$ is either b or i. In this case, $C_j(L'_j)$ is included in the intersection of the affine half-space $\{\boldsymbol{x} : \boldsymbol{x} \cdot \boldsymbol{u}^{(j)} \geq y\}$ with $\bar{H}$, thus it is contained in $H(i, b, \beta, d)$. Also, it does not contain any product in $H(a, i, c, \alpha)$. Combining the above two observation, since $S^{(V)}$ is equal to the union of $C_j(L'_j)$ over $j \in T^{(V)}$, we infer that any connected component of $(S^{(V)}, T^{(V)}, \boldsymbol{L}')$ has its products either included in $H(i, b, \beta, d)$ or contained in $H(a, i, c, \alpha)$.

To prove the desired inductive property, it remains to show that the allocation information is fully captured by each polyhedron. By symmetry, we may focus on $H(i, b, \beta, d)$. Observe that the customer-type deletions give rise to the cone $(\boldsymbol{u}_\beta, \boldsymbol{u}_d)$, such that each customer-type whose utility vector lies in this cone is not discarded at the moment. Similarly, the constraints of the polyhedra $H(i, b, \beta, d)$ with left-hand side $x \cdot u_\beta$ and $x \cdot u_d$ capture all the product deletions active in the polyhedron $H(i, b, \beta, d)$. Finally, since b and i are the most preferred products among all products stocked prior for any customer-type j in the cone $(\boldsymbol{u}_\beta, \boldsymbol{u}_d)$, the truncation of its consideration sets is captured by the equality $L_j = \min(\sigma_j(i), \sigma_j(b))$.

A.6 Capacitated Optimization

The approach that we have described extends to the *capacitated* variant of the problem. Specifically, we consider the assortment optimization problem wherein at most B products can be stocked. This constraint represents storage or display space constraints, or the limited number of spots of a web page in the context of e-retail and online advertising.

The complexity performance for the different model specifications analyzed in Sections 3.4 and 3.6 carries over to the constrained setting, up to a polynomial factor. Specifically, the problem is solved by an extension of our dynamic program. We add a single state variable that encodes the remaining 'capacity' budget for each subproblem, i.e., subproblems are duplicated to account for all possible budget values $[B]$. The new computational tree is inferred by adding edges between any pair of duplicated subproblems that were previously linked by the recursive formula, as long as the budget of the child subproblem is smaller than that of the father subproblem. The recursive formula decides on how to spread the remaining capacity budget across the children subproblems. We prove that, at each step of the recursion, the optimal capacity allocation is determined by solving a shortest path problem that we explicitly describe below. For sake of clarity we will only consider the unique-ranking case wherein $(S, T, \boldsymbol{L}) \sim (S, T)$, but the reasoning is similar for the general algorithm.

State space. The state space is described by the 3-tuple (S, T, b) where b is a new variable that encodes the maximal capacity budget. In this notation, $J(S, T, b)$ designates the maximal expected revenue garnered from customer-types T with an assortment of at most b products in S. The graph and subproblem notations remain unchanged.

Recursion formula. The recursion formula should be generalized to account for all potential different budget allocations. Hence, we introduce $B(b, r)$ the set of all feasible allocations of a capacity of b products between r classes of customers: $B(b, r) = \{\boldsymbol{b} \in \mathbb{N}^r \mid \sum_{j=1}^{r} b_j = b\}$. The recursive formula between subproblems becomes:

$$J(S,T) = \max \Bigg[P_i \cdot \sum_{j \in T(i)} \lambda_j + \max_{\mathbf{b} \in B(b-1,r(+))} \sum_{u=1}^{r(+)} J(S_u^+, T_u^+, b_u) \ , \tag{A.1}$$

$$\max_{\mathbf{b} \in B(b,r(-))} \sum_{u=1}^{r(-)} J(S_u^-, T_u^-, b_u) \Bigg] \tag{A.2}$$

Resource allocation problem. We observe that finding the optimal budget allocations in each max-expression (A.2) and (A.1) boils down to solving a simple allocation problem of the form

$$\max_{\sum_{i=1}^{k} b_i \leq b} \sum_{i=1}^{k} f(i, b_i) \ ,$$

where the integral non-negative decision variables b_i are coupled by a single constraint. It is well known that this problem can be efficiently solved by means of dynamic programming; see for instance Katoh and Ibaraki (1998).

A.7 Synthetic Computational Experiments

Generative models. The prices of products are sampled independently and identically from a log-normal distribution. The scale parameter is calibrated to reflect realistically the variability of prices in the Shampoo product category. The probability vector is drawn uniformly at random from the unit simplex. To generate instances of the quasi-convex model, the collection of preference lists is formed by independent and uniformly-distributed samples over the class of quasi-convex permutations. To construct instances with arbitrary consideration sets, we use a random Bernoulli generator, as explained in Section 3.7.2. The ranking function is given by the increasing price order.

Implementation of our algorithm. We use a ‘plain’ implementation of our algorithm which follows the two-pass approach explained in Section 3.3. First, we generate the computational tree using the recursive equations. Next, we compute the value function by solving a maximum flow problem. In the quasi-convex case, each subproblem is simply encoded by the latest three dynamic programming decisions, leading to an implementation in time $O(n^3K)$.

A.8 State Space Collapse in Experiments

Table A.1: Relative size of the collapsed state space in comparison to naive enumeration.

n	K	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$
20	1000	17.8%	4.8%	1.7%
20	2000	22.5%	8.0%	3.6%
100	20	–	$< 0.1\%$	$< 0.1\%$

.

Appendix B

Appendix of Chapter 4

B.1 Additional Proofs

B.1.1 Proof of Lemma 4.2.6

In what follows, we prove a sufficient condition for subadditivity, stating that with respect to any inventory vector, the deletion of units can only increase the consumption probability of any remaining unit. Formally, any inventory vector U is viewed as a collection of units, each of which is a separate copy of a given product; within units of the same product, a fixed (arbitrary) order is set, according to which they are consumed by customers. We denote by U^{-i} the inventory vector obtained from U by deleting the first unit of product i, and use $[\mathcal{E}_v|U]$ to designate the event where unit v is purchased during the consumption process with the initial inventory vector U.

Lemma B.1.1. *For any inventory vector U, any product $i \in [n]$ with $U_i \geq 1$, and any unit $v \in U^{-i}$,*

$$\Pr\left[\mathcal{E}_v \left| U^{-i}\right.\right] \geq \Pr\left[\mathcal{E}_v | U\right] \ .$$

Proof. Given an initial inventory vector U, we designate by $U_{(1)}, \ldots, U_{(M)}$ the random sequence of residual inventory levels facing each customer arrival, i.e., $U_{(m)}$ is the inventory vector facing the m-th arriving customer. To prove the desired claim, it suffices to show that $U_{(m)}(w) \geq U^{-i}_{(m)}(w)$ for any realization w of the consumption process. Here, any such realization corresponds to the specific outcomes of the number of arriving customers M and their choices of preference lists.

We now focus on a fixed realization w, and prove the latter claim by induction on the arrival rank m. For $m = 1$, we have by definition $U_{(1)}(w) = U > U^{-i} = U^{-i}_{(1)}(w)$. For the general case, by the induction hypothesis, we have $U_{(m-1)}(w) \geq U^{-i}_{(m-1)}(w)$. Now let k be the first product on the preference list of customer $m-1$ (picked according to w) that is stocked by $U_{(m-1)}(w)$; we define $k = \infty$ when not such product exists. As any product stocked by $U^{-i}_{(m-1)}(w)$ is necessarily stocked by $U_{(m-1)}(w)$, there are three cases:

1. $k = \infty$: Here, customer $m-1$ does not purchase any unit with respect to both $U_{(m-1)}(w)$ and $U^{-i}_{(m-1)}(w)$, meaning that $U_{(m)}(w) = U_{(m-1)}(w) \geq U^{-i}_{(m-1)}(w) = U^{-i}_{(m)}(w)$.

2. *$k < \infty$ and product k is stocked by both $U_{(m-1)}(w)$ and $U^{-i}_{(m-1)}(w)$*: In this case, customer $m-1$ purchases a single unit of product k in $U_{(m-1)}(w)$ and $U^{-i}_{(m-1)}(w)$, implying that $U_{(m)}(w) \geq U^{-i}_{(m)}(w)$.

3. *$k < \infty$ and product k is stocked by $U_{(m-1)}(w)$ but not by $U^{-i}_{(m-1)}(w)$*: Even though customer $m-1$ purchases a single unit of product k in $U_{(m-1)}(w)$, the remaining number of units stocked of this product is still greater or equal to the same quantity with respect to the residual vector after purchasing from $U^{-i}_{(m-1)}(w)$. For the latter, customer $m-1$ purchases a single unit of a product different than k or does not purchase at all; in either case, we have $U_{(m)}(w) \geq U^{-i}_{(m)}(w)$.

□

B.1.2 Proof of Lemma 4.2.4

The claim follows by observing that

$$\begin{aligned}
\mathbb{E}\left[X\right] &= \mathbb{E}\left[\min\left\{X, C\right\} + \left[X - C\right]^{+}\right] \\
&= \mathbb{E}\left[\bar{X}\right] + \Pr\left[X \geq C\right] \cdot \mathbb{E}\left[X - C | X \geq C\right] \\
&\leq \mathbb{E}\left[\bar{X}\right] + \delta \cdot \mathbb{E}\left[X\right] \ .
\end{aligned}$$

The last inequality holds since X is IFR and since

$$\delta C \geq \mathbb{E}\left[\bar{X}\right] = \mathbb{E}\left[\min\{X, C\}\right] \geq C \cdot \Pr\left[X \geq C\right] \ .$$

B.1.3 Counter-Examples

Lemma B.1.2. *Under the nested choice model and a single arrival, the set function* $f : \ \{0,1\}^N \rightarrow \mathbb{R}^+$ *is neither monotone nor submodular .*

Proof. Consider the following instance: there are three products denoted by $\{1,2,3\}$ with respective prices $P_1 = 1, P_2 = 2$ and $P_3 = 3$, while the total capacity is $C = 1$. We model a single customer arrival, where the list $(1,2,3)$ occurs with probability 1. With a slight abuse of notation, where sets are used instead of binary sequences, it is easy to verify that for $S_1 = \{3\}$ and $S_2 = \{1,3\}$, we have $f(S_1) > f(S_2)$, and $f(S_1 \cup \{2\}) - f(S_1) = -1$ while $f(S_2 \cup \{2\}) - f(S_2) = 0$. □

Lemma B.1.3. *Under the nested choice model and IFR demand distributions, the expected revenue function is not concave.*

Proof. Consider the following instance: there are two products denoted by $\{1,2\}$, with prices $P_1 = 0$ and $P_2 = 1$. There are two arriving customers, each of which draws the preference list $(1,2)$ with probability $1/2$ and the empty list with probability $1/2$. We consider the inventory vectors $(2,0)$, $(0,2)$ and $(1,1)$. We observe that $\mathbb{E}\left[\mathcal{R}(2,0)\right] = 0$,

$\mathbb{E}\left[\mathcal{R}(0,2)\right] = 1$ and $\mathbb{E}\left[\mathcal{R}(1,1)\right] = 1/4$. As a result, we obtain that

$$\frac{1}{2} \cdot \mathbb{E}\left[\mathcal{R}(2,0)\right] + \frac{1}{2} \cdot \mathbb{E}\left[\mathcal{R}(0,2)\right] > \mathbb{E}\left[\mathcal{R}(1,1)\right] = \mathbb{E}\left[\mathcal{R}\left(\frac{1}{2} \cdot (2,0) + \frac{1}{2} \cdot (0,2)\right)\right] .$$

□

Lemma B.1.4. *Under the interval choice model, the revenue function is not submodular, even with deterministic arrivals and uniform prices.*

Proof. To construct a counterexample, we consider the collection of products $\{1,2,3\}$ such that $P_1 = P_2 = P_3 = 1$, and define the consumption process where there are exactly two arrivals. Each of these customers samples a preference list according to the following distribution: with probability $1-\epsilon$, the customer chooses the list (2), and with probability ϵ she chooses the list $(1,2,3)$, where $\epsilon < 1/2$.

For this instance, focusing on the sets of products $S_1 = \{3\}$ and $S_2 = \{2,3\}$, we have $\mathbb{E}[\mathcal{R}(S_1 \cup \{1\})] - \mathbb{E}[\mathcal{R}(S_1)] = \epsilon^2$ whereas $\mathbb{E}[\mathcal{R}(S_2 \cup \{1\})] - \mathbb{E}[\mathcal{R}(S_2)] = \epsilon \cdot (1-\epsilon)$. Note that $\epsilon \cdot (1-\epsilon) > \epsilon^2$ since $\epsilon < 1/2$, meaning that

$$\mathbb{E}\left[\mathcal{R}(S_2 \cup \{1\})\right] - \mathbb{E}\left[\mathcal{R}(S_2)\right] > \mathbb{E}\left[\mathcal{R}(S_1 \cup \{1\})\right] - \mathbb{E}\left[\mathcal{R}(S_1)\right] .$$

□

Appendix C

Appendix of Chapter 5

C.1 General Constant-Factor Approximation

The approximation algorithm proposed in Section 5.3 relies on the efficient oracle assumption in order to compute the expected revenue generated by any given inventory vector. However, for the Multinomial Logit choice model, whether or not the expected revenue function can be evaluated in polynomial time (even approximately) is still an open question. We work around this difficulty by decomposing the set of products beforehand, and arguing that the terms requiring more effort from an optimization standpoint admit a sampling-based evaluation oracle, compatible with the algorithm developed in Section 5.3. Consequently, we establish the following theorem.

Theorem C.1.1. *For any $\epsilon \in (0, 1/4)$ and $\delta > 0$, the dynamic assortment planning problem under the Multinomial Logit choice model with IFR demand distribution can be approximated within a factor of $0.122 - \epsilon$, with probability at least $1 - \delta$. The running time of our algorithm is polynomial in the input size, $n^{1/\epsilon}$, and $1/\delta$. When $C \geq n$, this factor can be improved to $0.151 - \epsilon$.*

High-level overview of the algorithm. To work around the estimation obstacle, we make use of the decomposition idea explained in Section 5.2.2. Somewhat informally, sampling procedures fail to estimate the expected revenue accurately when there are very low probability purchase events, that require an exponential number of samples to be observed. Such rare events are possible when there is large variability between the preference weights of different products. Thus motivated, as a preliminary step, we partition the set of products into two classes, *light* and *heavy*, based on their respective MNL preference weights.

As a result, our decomposition generates two subproblems: one instance exclusively formed by heavy products, and another instance comprised of light products. Using an appropriate estimator, we show that the expected revenue in the heavy products instance can be efficiently approximated through sampling. Consequently, the methods developed in Section 5.3 provide a polynomial-time randomized algorithm, with a constant-factor worst-case guarantee. On the other hand, we show that a relatively simple approximation scheme can be derived for the light products instance.

Partition of products. To formalize this approach, the collection of products $[n]$ is decomposed into two sets:

- The set $\mathcal{L}$ of light products, consisting of those with $w_i \in (0, \epsilon/n]$.
- The set $\mathcal{H}$ of heavy products, with $w_i \in (\epsilon/n, \infty)$.

Product elimination. We further restrict attention to a smaller subset of heavy products, by eliminating in advance certain products whose revenue contribution toward $\mathbb{E}[\mathcal{R}(U^*)]$ is negligible. Specifically, let $i_{\max}$ be the heavy product that maximizes the quantity $r_i w_i/(1+w_i)$ over all products $i \in \mathcal{H}$ stocked by $U^*_{\mathcal{H}}$. From an algorithmic perspective, $i_{\max}$ can be guessed by considering $|\mathcal{H}|$ options, and we can now define the residual collection of heavy products $\tilde{\mathcal{H}} = \{i \in \mathcal{H} : \frac{\epsilon^2 r_{i_{\max}}}{2n^2 C} \leq r_i \leq \frac{2n^2 C \cdot r_{i_{\max}}}{\epsilon^3}\}$.

Upper bound on the optimal expected revenue. We now argue that the classes of products $\mathcal{L}$ and $\tilde{\mathcal{H}}$ are sufficient to compete against U^*. Recall that $U^*_{\mathcal{L}}$ denotes the projection of the optimal inventory vector U^* on light products, i.e., the vector obtained from U^* by setting the inventory levels of $[n] \setminus \mathcal{L}$ to zero. The vector $U^*_{\tilde{\mathcal{H}}}$ is defined in an analogous way. By exploiting the subadditive nature of the expected revenue function (see Lemma 5.2.2), we derive an upper bound on $\mathbb{E}[\mathcal{R}(U^*)]$ in the next lemma, whose proof is given in Appendix C.3.8.

Lemma C.1.2. $\mathbb{E}[\mathcal{R}(U^*_{\mathcal{L}})] + \mathbb{E}[\mathcal{R}(U^*_{\tilde{\mathcal{H}}})] \geq (1-2\epsilon) \cdot \mathbb{E}[\mathcal{R}(U^*)]$.

C.1.1 Efficient oracle for heavy products

Here, we show that the subproblem restricted to the heavy products $\tilde{\mathcal{H}}$ admits an efficient oracle. That is, for any error parameter $\epsilon > 0$ and confidence level $\delta > 0$, we devise a procedure to evaluate the expected revenue within a multiplicative factor of $1 \pm \epsilon$, running in time polynomial in the input size, $1/\epsilon$, and $1/\delta$.

For this purpose, suppose we are given an inventory vector U that stocks at most C units of products in $\tilde{\mathcal{H}}$, and wish to estimate $\mathbb{E}[\mathcal{R}(U)]$. Our evaluation procedure samples $L = \lceil 64C^6 n^{10}/(\epsilon^{12}\delta) \rceil$ independent realizations $R_1, \dots, R_L$ of the random variable $\mathcal{R}(U)$ conditional on $M \geq 1$. These conditional realizations are obtained by sampling from a modified instance, where the number of arrivals M is replaced by $M|M \geq 1$. Next, the expected revenue $\mathbb{E}[\mathcal{R}(U)]$ is estimated by the unbiased estimator

$$\tilde{R} = \Pr\left[M \geq 1\right] \cdot \frac{1}{L} \cdot \sum_{\ell=1}^{L} R_\ell \ . \tag{C.1}$$

Lemma C.1.3. *The estimator $\tilde{R}$ provides an efficient oracle for the expected revenue function, i.e.,*

$$\Pr\left[\left|\frac{\tilde{R}}{\mathbb{E}\left[\mathcal{R}\left(U\right)\right]} - 1\right| \geq \epsilon\right] \leq \delta \ .$$

Proof. The proof relies on bounding the variance of the conditional revenue relative to its expected value, before applying Chebyshev's inequality. Since U stocks at most C units, the random variable $\mathcal{R}(U)|M \geq 1$ is upper bounded by $C \cdot r_{i_1}$ for any realization, where $i_1 \in \tilde{\mathcal{H}}$ is the most expensive product stocked by U. Also, letting i_2 be the maximal preference weight product stocked by U, an immediate lower bound on the expectation of this random variable is given by

$$\mathbb{E}\left[\mathcal{R}\left(U\right) \middle| M \geq 1\right] \geq \frac{r_{i_2} w_{i_2}}{1 + |\tilde{\mathcal{H}}| \cdot w_{i_2}} \geq \frac{\epsilon r_{i_2}}{2n} \ , \tag{C.2}$$

where the first inequality accounts for the expected revenue due to the first arriving customer, who purchases a unit of product i_2 with probability at least $w_{i_2}/(1 + |\tilde{\mathcal{H}}|\cdot w_{i_2})$, given that i_2 the has maximal preference weight among all products stocked by U, and the second inequality holds since $w_{i_2} \geq \epsilon/n$. Note that, since $\mathbb{E}[\mathcal{R}(U)|M \geq 1] = \mathbb{E}[\mathcal{R}(U)]/\Pr[M \geq 1]$, we have:

$$\Pr\left[\left|\tilde{R} - \mathbb{E}\left[\mathcal{R}\left(U\right)\right]\right| \geq \epsilon \cdot \mathbb{E}\left[\mathcal{R}\left(U\right)\right]\right] = \Pr\left[\left|\frac{1}{L} \cdot \sum_{\ell=1}^{L} R_\ell - \mathbb{E}\left[\mathcal{R}\left(U\right)\right| M \geq 1]\right| \geq \epsilon \cdot \mathbb{E}\left[\mathcal{R}\left(U\right)\right| M \geq 1]\right] .$$

Hence, by Chebyshev's inequality,

$$\Pr\left[\left|\tilde{R} - \mathbb{E}\left[\mathcal{R}\left(U\right)\right]\right| \geq \epsilon \cdot \mathbb{E}\left[\mathcal{R}\left(U\right)\right]\right] \leq \frac{\mathrm{var}((1/L) \cdot \sum_{\ell=1}^{L} R_\ell)}{\epsilon^2 \cdot (\mathbb{E}\left[\mathcal{R}\left(U\right)\right| M \geq 1])^2} \leq \frac{4C^2 n^2}{\epsilon^2 L} \cdot \frac{r_{i_1}^2}{r_{i_2}^2} \leq \delta ,$$

where the second inequality follows from (C.2), along with the upper bound of $C^2 \cdot r_{i_1}^2$ on the second moment of each sample R_ℓ, and the last inequality holds since $L = \lceil 64 C^6 n^{10}/(\epsilon^{12}\delta) \rceil$ while $r_{i_1}/r_{i_2} \leq 4C^2 n^4/\epsilon^5$, by definition of $\tilde{\mathcal{H}}$. □

C.1.2 Approximation scheme for light products

The approach for handling light products $\mathcal{L}$ relies on identifying a newsvendor-like lower bound, in the spirit of Section 5.3.3. The important observation is that, when we are restricted to stocking only light products, each arriving customer faces a random assortment $S \subseteq \mathcal{L}$ with total weight $w(S) \leq |S|\cdot\epsilon/n \leq \epsilon$. Thus, as long as product $i \in \mathcal{L}$ is available, it is purchased by an arriving customer with probability at least $w_i/(1+\epsilon) \geq (1-\epsilon) \cdot w_i$, regardless of what the other available products are. Hence, at least intuitively, at the cost of losing a negligible factor in optimality, one could view the contribution of each product to the expected revenue as if it depends only on the initial number of units stocked.

Algorithm. To turn this intuition into a concrete argument, suppose that U is an inventory vector that stocks only light products. Then, the number of units purchased from each product $i \in \mathcal{L}$ is stochastically larger than the random variable

$\bar{Y}_i(u_i) = \min\{Y_i, u_i\}$, where $Y_i \sim B(M, (1-\epsilon) \cdot w_i)$. Therefore,

$$\mathbb{E}\left[\mathcal{R}(U)\right] \geq \sum_{i \in \mathcal{L}} r_i \cdot \mathbb{E}\left[\bar{Y}_i(u_i)\right] \ . \tag{C.3}$$

Our algorithm optimizes the latter newsvendor-like lower bound, by computing an optimal solution to the following problem:

$$\max_{U} \left\{ \sum_{i \in \mathcal{L}} r_i \cdot \mathbb{E}\left[\bar{Y}_i(u_i)\right] : \sum_{i \in \mathcal{L}} u_i \leq C \right\} \ . \tag{C.4}$$

As explained in Section 5.3.3, an optimal solution to this problem can be computed efficiently by means of a greedy procedure. Note that the expectation $\mathbb{E}[\bar{Y}_i(u_i)]$ can be computed in polynomial time with respect to C and the maximum value of M, using a simple dynamic program.

The next lemma shows that the inventory vector $U_{\mathcal{L}}$, obtained by solving problem (C.4), guarantees a $(1-\epsilon)$-approximation with respect to the inventory vector $U^*_{\mathcal{L}}$.

Lemma C.1.4. $\mathbb{E}\left[\mathcal{R}(U_{\mathcal{L}})\right] \geq (1-\epsilon) \cdot \mathbb{E}\left[\mathcal{R}(U^*_{\mathcal{L}})\right]$.

Proof. First, observe that the lower bound (C.3) can be complemented by an upper bound on the expected revenue of $U^*_{\mathcal{L}}$. Specifically, letting $Y^*_i \sim B(M, w_i)$ and $\bar{Y}^*_i = \min\{Y^*_i, u^*_i\}$, we have

$$\mathbb{E}\left[\mathcal{R}(U^*_{\mathcal{L}})\right] \leq \sum_{i \in \mathcal{L}} r_i \cdot \mathbb{E}\left[\bar{Y}^*_i\right] \ . \tag{C.5}$$

Based on inequalities (C.3) and (C.5), since $U_{\mathcal{L}}$ is an optimal solution to problem (C.4), it remains to show that the objective value of $U^*_{\mathcal{L}}$ with respect to the latter problem is at least $(1-\epsilon) \cdot \sum_{i \in \mathcal{L}} r_i \cdot \mathbb{E}[\bar{Y}^*_i]$. This follows by observing that $\mathbb{E}[\bar{Y}_i(u^*_i)] \geq (1-\epsilon) \cdot \mathbb{E}[\bar{Y}^*_i]$ for any product $i \in \mathcal{L}$, where the latter inequality is an immediate consequence of the next claim (proven in Appendix C.3.9), specialized for $\theta = 1 - \epsilon$.

Claim C.1.5. *Let M be a non-negative integer-valued random variable, and suppose that $X \sim B(M, \alpha)$ and $Y \sim B(M, \theta\alpha)$, where $\alpha \in [0,1]$ and $\theta \in [0,1]$. For some* *integer C, let $\bar{X} = \min\{X, C\}$ and $\bar{Y} = \min\{Y, C\}$. Then, $\mathbb{E}[\bar{Y}] \geq \theta \cdot \mathbb{E}[\bar{X}]$.*

□

C.1.3 Conclusion

To summarize, our algorithm computes two approximate inventory vectors, corresponding to the weight classes $\mathcal{L}$ and $\tilde{\mathcal{H}}$, and eventually picks the one with maximal expected revenue.

- *Heavy products.* We employ the algorithm described in Section 5.3, for the subproblem restricted to the heavy products $\tilde{\mathcal{H}}$. This algorithm relies on the efficient oracle assumption, and therefore, we utilize the efficient sampling-based procedure described in Appendix C.1.1, running in time polynomial in the input size, $n^{1/\epsilon}$, and $1/\delta$. By Theorem 5.3.1, the random vector $U_{\tilde{\mathcal{H}}}$ returned by this algorithm satisfies $\mathbb{E}[\mathcal{R}(U_{\tilde{\mathcal{H}}})] \geq (0.139 - \epsilon) \cdot \mathbb{E}[\mathcal{R}(U^*_{\tilde{\mathcal{H}}})]$, with probability at least $1 - \delta$.

- *Light products.* The vector $U_{\mathcal{L}}$, returned by the algorithm described in Section C.1.2, is a $(1 - \epsilon)$-approximation with respect to the expected revenue of $U^*_{\mathcal{L}}$, i.e., $\mathbb{E}[\mathcal{R}(U_{\mathcal{L}}] \geq (1 - \epsilon) \cdot \mathbb{E}[\mathcal{R}(U^*_{\mathcal{L}}]$. In addition, this guarantee applies to a lower bound, that can be efficiently computed through dynamic programming.

Establishing Theorem C.1.1. Since we pick the best vector out of $U_{\mathcal{L}}$ and $U_{\tilde{\mathcal{H}}}$, with probability at least $1 - \delta$, for any $\alpha \in [0, 1]$ we obtain an expected revenue of at least

$$\begin{aligned}
&\max \left\{ \mathbb{E}\left[\mathcal{R}(U_{\mathcal{L}})\right], \mathbb{E}\left[\mathcal{R}(U_{\tilde{\mathcal{H}}})\right] \right\} \\
&\qquad \geq \alpha \cdot \mathbb{E}\left[\mathcal{R}(U_{\mathcal{L}})\right] + (1 - \alpha) \cdot \mathbb{E}\left[\mathcal{R}(U_{\tilde{\mathcal{H}}})\right] \\
&\qquad \geq (1 - 8\epsilon) \cdot \left(\alpha \cdot \mathbb{E}\left[\mathcal{R}(U^*_{\mathcal{L}})\right] + 0.139 \cdot (1 - \alpha) \mathbb{E}\left[\mathcal{R}(U^*_{\tilde{\mathcal{H}}})\right] \right) \ .
\end{aligned}$$

By choosing $\alpha = 0.139/1.139 \approx 0.122$, we have

$$\begin{aligned}
&\max \left\{ \mathbb{E}\left[\mathcal{R}(U_{\mathcal{L}})\right], \mathbb{E}\left[\mathcal{R}(U_{\tilde{\mathcal{H}}})\right] \right\} \\
&\qquad \geq 0.122 \cdot (1 - 8\epsilon) \cdot \left(\mathbb{E}\left[\mathcal{R}(U^*_{\mathcal{L}})\right] + \mathbb{E}\left[\mathcal{R}(U^*_{\tilde{\mathcal{H}}})\right] \right) \\
&\qquad \geq (0.122 - 2\epsilon) \cdot \mathbb{E}\left[\mathcal{R}(U^*)\right] \ ,
\end{aligned}$$

where the last inequality is due to Lemma C.1.2. In the special case where $C \geq n$, an improved guarantee of $0.151 - \epsilon$ is derived by plugging the refined approximation ratio of $0.179 - \epsilon$ given by Theorem 5.3.1 for the heavy products vector $U_{\tilde{\mathcal{H}}}$.

C.2 Logarithmic approximation for non-IFR demand distributions

In what follows, recall that U^* is an optimal inventory vector, and for any subset of products $S \subseteq [n]$ we use U^*_S to denote the projection of U^* on S.

Step 1: Decomposition. Similar to the algorithm described in Appendix C.1, we begin by partitioning the set of products into the weight classes $\mathcal{L}$ and $\mathcal{H}$, by specifically choosing $\epsilon = 1/4$. Since our approximation algorithm for light products (Section C.1.2) does not rely on the IFR property, the resulting inventory vector $U_{\mathcal{L}}$ still attains a performance guarantee of $3/4$ with respect to $U^*_{\mathcal{L}}$.

Now, let $i_{\max} \in \mathcal{H}$ be the most expensive heavy product. From an algorithmic perspective, this product can be guessed by considering $|\mathcal{H}|$ options. With this definition at hand, we construct the subset of products $\mathcal{H}^+ \subseteq \mathcal{H}$ whose selling price is at least $r_{i_{\max}}/(8n)$, and designate by $\mathcal{H}^-$ the remaining heavy products.

Step 2: Competing against cheap heavy products. Let $U_{\mathcal{H}^-}$ be the inventory vector that stocks C units of product $i_{\max}$. In the next claim, whose proof is deferred to the end of this section, we argue that $U_{\mathcal{H}^-}$ is at least as good revenue-wise as $U^*_{\mathcal{H}^-}$.

Claim C.2.1. $\mathbb{E}\left[\mathcal{R}(U_{\mathcal{H}^-})\right] \geq \mathbb{E}\left[\mathcal{R}(U^*_{\mathcal{H}^-})\right]$.

Step 3: Competing against expensive heavy products. We further decompose $\mathcal{H}^+$ into $K = \lceil \log(8n) \rceil$ nearly-uniform price classes $\mathcal{H}_1^+, \dots, \mathcal{H}_K^+$, where $\mathcal{H}_k^+ = \{i \in \mathcal{H}^+ : \frac{r_{i_{\max}}}{2^k} < r_i \leq \frac{r_{i_{\max}}}{2^{k-1}}\}$. Next, for every $k \in [K]$, our algorithm computes an inventory vector $U_{\mathcal{H}_k^+}$ that compete against $U^*_{\mathcal{H}_k^+}$. To this end, consider the subproblem where only products in $\mathcal{H}_k^+$ can be stocked, and let f^k be the corresponding expected revenue set function, that specifies the expected revenue associated with subsets of the extended collection of units $\mathcal{H}_k^+ \times [C]$ (see Section 5.3.2). By rounding up the selling prices of products in $\mathcal{H}_k^+$ to $r_{i_{\max}}/2^{k-1}$, the resulting expected revenue set function $\tilde{f}^k$ clearly satisfies, for any subset of units S,

$$\frac{1}{2} \cdot \tilde{f}^k(S) \leq f^k(S) \leq \tilde{f}^k(S) \ . \tag{C.6}$$

On the other hand, it is easy to verify that, when all selling prices are equal, the static expected revenue function associated with an instance of the MNL model is non-decreasing, implying in particular that $\tilde{f}_1^k$ is non-decreasing. As a result, the problem of maximizing $\tilde{f}^k(S)$ over subsets S of at most C units falls within the special setting discussed in Section 5.3.2.4. Therefore, the standard greedy algorithm, combined with the sampling-based oracle of Appendix C.1 (with appropriate error and confidence parameters), computes an inventory vector $U_{\mathcal{H}_k^+}$ such that, with probability at least $1 - \delta/K$,

$$\mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{H}_k^+}\right)\right] \geq \left(\frac{1}{2} \cdot \left(1 - \frac{1}{e}\right) - \epsilon\right) \cdot \mathbb{E}\left[\mathcal{R}\left(U^*_{\mathcal{H}_k^+}\right)\right] \ , \tag{C.7}$$

where the latter performance guarantee follows from the approximation ratio of Section 5.3.2.4 combined with (C.6).

Step 4: Picking the most profitable inventory vector. Finally, the algorithm selects the most profitable inventory vector out of $U_{\mathcal{L}}, U_{\mathcal{H}^-}, U_{\mathcal{H}_1^+}, \dots, U_{\mathcal{H}_K^+}$. Since the corresponding expected revenues are unknown, these vectors are compared using the randomized oracle (for $U_{\mathcal{H}^-}, U_{\mathcal{H}_1^+}, \dots, U_{\mathcal{H}_K^+}$), and the previously-mentioned lower bound for $U_{\mathcal{L}}$. Given the subadditivity of the expected revenue function (see

Lemma 5.2.2), since $\mathcal{L}, \mathcal{H}^-, \mathcal{H}_1^+, \ldots, \mathcal{H}_K^+$ form a partition of $[n]$, it follows that

$$\mathbb{E}\left[\mathcal{R}\left(U^*_{\mathcal{L}}\right)\right] + \mathbb{E}\left[\mathcal{R}\left(U^*_{\mathcal{H}^-}\right)\right] + \sum_{k \in [K]} \mathbb{E}\left[\mathcal{R}\left(U^*_{\mathcal{H}_k^+}\right)\right] \geq \mathbb{E}\left[\mathcal{R}\left(U^*\right)\right] \ .$$

Therefore, by the union bound, with probability at least $1-\delta$ we obtain that

$$\begin{aligned}
\max \ & \left\{\mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{L}}\right)\right], \mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{H}^-}\right)\right], \mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{H}_1^+}\right)\right], \ldots, \mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{H}_K^+}\right)\right]\right\} \\
& \geq \frac{(1/2)\cdot(1-1/e)-\epsilon}{K+2} \cdot \left(\mathbb{E}\left[\mathcal{R}\left(U^*_{\mathcal{L}}\right)\right] + \mathbb{E}\left[\mathcal{R}\left(U^*_{\mathcal{H}^-}\right)\right] + \sum_{k \in [K]} \mathbb{E}\left[\mathcal{R}\left(U^*_{\mathcal{H}_k^+}\right)\right]\right) \\
& \geq \frac{1}{4(K+2)} \cdot \mathbb{E}\left[\mathcal{R}\left(U^*\right)\right] \\
& = \Omega\left(\frac{1}{\log n}\right) \cdot \mathbb{E}\left[\mathcal{R}\left(U^*\right)\right] \ .
\end{aligned}$$

where the first inequality holds due to the performance guarantees stated in Lemma C.1.4, Claim C.2.1, and inequality (C.7).

Proof of Claim C.2.1.. Since the selling price of every product in $\mathcal{H}^-$ is at most $r_{i_{\max}}/(8n)$, an upper bound on the expected revenue of $U^*_{\mathcal{H}^-}$ is given by

$$\mathbb{E}\left[\mathcal{R}\left(U^*_{\mathcal{H}^-}\right)\right] \leq \frac{r_{i_{\max}}}{8n} \cdot \mathbb{E}\left[\min\{M, C\}\right] \ .$$

On the other hand, when initially stocking the inventory vector $U_{\mathcal{H}^-}$, until product $i_{\max}$ stocks out, each arriving customer purchases a unit of $i_{\max}$ with probability $w_{i_{\max}}/(1+w_{i_{\max}}) \geq \epsilon/(2n) = 1/(8n)$, where the latter inequality holds since $w_{i_{\max}} \geq \epsilon/n$, given that $i_{\max}$ is a heavy product. Consequently, letting $Y \sim B(M, 1/(8n))$, we have

$$\mathbb{E}\left[\mathcal{R}\left(U_{\mathcal{H}^-}\right)\right] \geq r_{i_{\max}} \cdot \mathbb{E}\left[\min\{Y, C\}\right] \geq \frac{r_{i_{\max}}}{8n} \cdot \mathbb{E}\left[\min\{M, C\}\right] \geq \mathbb{E}\left[\mathcal{R}\left(U^*_{\mathcal{H}^-}\right)\right] \ .$$

where the second inequality follows from Claim C.1.5 specialized with $\theta = 1/(8n)$. ■

C.3 Additional Proofs

C.3.1 Proof of Lemma 5.2.1

Algorithm. For a small cardinality value, i.e., $C < 1/\epsilon$, one could simply enumerate over all $O(n^{1/\epsilon})$ possible subsets, and pick the one with largest estimated objective value, according to an $(\epsilon/2, \delta/n^{1/\epsilon})$-oracle. It is not difficult to verify that, by the union bound, this enumerative procedure returns a $1-\epsilon$-approximate solution with probability at least $1-\delta$. For large cardinality values ($C \geq 1/\epsilon$), the algorithm is a standard greedy procedure. Starting with the empty set $S_0 = \emptyset$, we add in each step the element that generates the largest increase in the objective function, among all unpicked elements. At each step, we call the random $(\epsilon/(2C), \delta/(nC))$-oracle to evaluate the objective value associated with each unpicked element. Let $S_0, S_1, \ldots, S_C$ be the sequence of subsets corresponding to the different steps in the algorithm, and let S^* be a fixed optimal subset. We assume without loss of generality that $|S^*|= C$, as f is restricted-non-decreasing.

Analysis. Since the greedy algorithm makes at most nC calls to the randomized oracle, by the union bound the relative error associated with all estimates returned by the $(\epsilon/(2C), \delta/(nC))$-oracle is upper bounded by $\epsilon/(2C)$ with probability at least $1-\delta$. From this point on, we establish the desired approximation guarantee under this condition. Below, for any subsets S and T, we use $f_S(T)$ to denote the marginal variation in f when S is augmented by T, i.e., $f_S(T) = f(S \cup \{T\}) - f(S)$.

Claim C.3.1. *Let S and T be disjoint subsets, with $|S|+|T|\leq C$ and $T \neq \emptyset$. Then, for every $0 \leq k \leq |T|$, there exists $T_k \subseteq T$ with $|T_k|= k$ and $f_S(T_k) \geq (k/|T|) \cdot f_S(T)$.*

Proof. The proof follows by an inductive argument over k. The base case $k = 0$ is clearly satisfied by $T_0 = \emptyset$. For the general case, by the induction hypothesis, there exists $T_k \subseteq T$ with $|T_k|= k$ and $f_S(T_k) \geq (k/|T|) \cdot f_S(T)$. Letting $T \setminus T_k =$

$\{e_1, \ldots, e_{|T|-k}\}$, we have

$$f_{S \cup T_k}(T \setminus T_k) = \sum_{j=0}^{|T|-k-1} f_{S \cup T_k \cup \{e_1,\ldots,e_j\}}(e_{j+1}) \leq \sum_{j=0}^{|T|-k-1} f_{S \cup T_k}(e_{j+1}) \ ,$$

where the latter inequality holds since f is restricted-submodular, by observing that $|S \cup T_k \cup \{e_1, \ldots, e_{|T|-k-1}\}|= |S|+|T|-1 \leq C-1$. Consequently, there exists $1 \leq j \leq |T|-k$ such that $f_{S \cup T_k}(e_j) \geq f_{S \cup T_k}(T \setminus T_k)/(|T|-k)$. As a result, by defining $T_{k+1} = T_k \cup \{e_j\}$, we obtain

$$\begin{aligned} f_S(T_{k+1}) &= f_S(T_k) + f_{S \cup T_k}(e_j) \\ &\geq f_S(T_k) + \frac{1}{|T|-k} \cdot f_{S \cup T_k}(T \setminus T_k) \\ &= \left(1 - \frac{1}{|T|-k}\right) \cdot f_S(T_k) + \frac{1}{|T|-k} \cdot f_S(T) \\ &\geq \left(\frac{k}{|T|} \cdot \left(1 - \frac{1}{|T|-k}\right) + \frac{1}{|T|-k}\right) \cdot f_S(T) \\ &= \frac{k+1}{|T|} \cdot f_S(T) \ , \end{aligned}$$

where the second equality holds since $f_{S \cup T_k}(T \setminus T_k) = f_S(T) - f_S(T_k)$, and the second inequality proceeds from the inductive hypothesis. □

It immediately follows from Claim C.3.1 that, for every $0 \leq k \leq C$, there exists a subset $S^*_{C-k} \subseteq S^*$ such that $|S^*_{C-k}|= C-k$, and

$$f_\emptyset\left(S^*_{C-k}\right) \geq \frac{C-k}{C} \cdot f_\emptyset\left(S^*\right) \ . \tag{C.8}$$

We can now analyze the sequence of subsets $S_0, \ldots, S_C$ produced by our (random) greedy procedure, where we use e_{k+1} to denote the unique element of $S_{k+1} \setminus S_k$, for every $0 \leq k \leq C-1$. To establish lower bounds on $f_{S_k}(e_{k+1})$ for every $0 \leq k \leq C-1$, we make the following case disjunction:

- *Case A:* $S^*_{C-k} \setminus S_k \neq \emptyset$. Observe that S_k and $S^*_{C-k} \setminus S_k$ are disjoint, and $|S_k|+|S^*_{C-k} \setminus S_k| \leq C$. By Claim C.3.1, since $S^*_{C-k} \setminus S_k \neq \emptyset$ by the case hypothesis, it follows that there exists an element $e \in S^*_{C-k} \setminus S_k$ such that

$$\begin{aligned} f_{S_k}(e) &\geq \frac{1}{|S^*_{C-k} \setminus S_k|} \cdot f_{S_k}\left(S^*_{C-k} \setminus S_k\right) \\ &\geq \frac{1}{C-k} \cdot \left(f_\emptyset\left(S_k \cup S^*_{C-k}\right) - f_\emptyset(S_k)\right) \\ &\geq \frac{1}{C-k} \cdot \left(f_\emptyset\left(S^*_{C-k}\right) - f_\emptyset(S_k)\right) \\ &\geq \frac{1}{C} \cdot f_\emptyset(S^*) - \frac{1}{C-k} \cdot f_\emptyset(S_k) \ , \end{aligned} \tag{C.9}$$

where the third inequality holds since f is restricted-non-decreasing and $|S_k \cup S^*_{C-k}| \leq C$, while the last inequality proceeds from (C.8).

- *Case B: $S^*_{C-k} \setminus S_k = \emptyset$.* In this case, $S^*_{C-k} \subseteq S_k$, and therefore

$$f_\emptyset(S_k) \geq f_\emptyset\left(S^*_{C-k}\right) \geq \frac{C-k}{C} \cdot f_\emptyset(S^*) \ , \tag{C.10}$$

where the first inequality holds since f is restricted-non-decreasing, and the last inequality follows from (C.8).

Concluding the analysis. Let $\mu \in [0,1]$ be a parameter that will be optimized later on, and let $L = \lfloor (1-\mu) \cdot C \rfloor$. When $f(S_C) \geq \mu \cdot f(S^*)$, our algorithm attains a μ-approximation. When $f(S_C) < \mu \cdot f(S^*)$, for every $k \leq (1-\mu) \cdot C$ we necessarily have $S^*_{C-k} \setminus S_k \neq \emptyset$, or otherwise

$$f_\emptyset(S_C) \geq f_\emptyset(S_k) \geq \frac{C-k}{C} \cdot f_\emptyset(S^*) \geq \mu \cdot f_\emptyset(S^*) \ ,$$

where the first inequality is due to f being restricted-non-decreasing, and the second inequality follows from (C.10); since $f(\emptyset) \geq 0$, the latter observation would imply that $f(S_C) \geq \mu \cdot f(S^*)$. As a result, in this setting we have

$$\begin{aligned}
f_{\emptyset}\left(S_C\right) &\geq f_{\emptyset}\left(S_L\right) \\
&= \sum_{k=0}^{L-1} f_{S_k}\left(e_{k+1}\right) \\
&\geq \sum_{k=0}^{L-1}\left(\frac{1}{C}\cdot f_{\emptyset}\left(S^*\right) - \frac{1}{C-k}\cdot f_{\emptyset}\left(S_k\right) - \frac{\epsilon}{C}\cdot f\left(S^*\right)\right) \\
&\geq \frac{L}{C}\cdot f_{\emptyset}\left(S^*\right) - \mu\cdot f_{\emptyset}\left(S^*\right)\cdot\sum_{k=0}^{L-1}\frac{1}{C-k} - \epsilon\cdot f\left(S^*\right) \\
&= \left(\frac{L}{C} - \mu\cdot\left(H_C - H_{C-L}\right)\right)\cdot f_{\emptyset}\left(S^*\right) - \epsilon\cdot f\left(S^*\right)\ ,
\end{aligned}$$

where $H_m = \sum_{k=1}^{m}\frac{1}{k}$ is the m-th harmonic number. Here, the first inequality holds since f is restricted-non-decreasing. The second inequality follows from (C.9), given that all estimates of the evaluation oracle are accurate up to a relative error of $\epsilon/(2C)$. The next inequality holds since $f(S_k) < \mu \cdot f(S^*)$ by hypothesis and since $f(\emptyset) \geq 0$.

Claim C.3.2. *$L/C - \mu\cdot(H_C - H_{C-L}) \geq 1 - \mu - \mu\ln\mu - O(\epsilon)$.*

Proof. Since $|H_n - \ln n| = \gamma + O(1/n)$, where γ is the EulerâĂŞMascheroni constant, we have

$$\begin{aligned}
\frac{L}{C} - \mu\cdot\left(H_C - H_{C-L}\right) &\geq \frac{L}{C} - \mu\cdot\ln\frac{C}{C-L} - \mu\cdot O\left(\frac{1}{C-L}\right) \\
&= \frac{\lfloor(1-\mu)\cdot C\rfloor}{C} + \mu\cdot\ln\left(1 - \frac{\lfloor(1-\mu)\cdot C\rfloor}{C}\right) - \mu\cdot O\left(\frac{1}{C - \lfloor(1-\mu)\cdot C\rfloor}\right) \\
&\geq 1 - \mu - \frac{1}{C} + \mu\ln\mu - O\left(\frac{1}{C}\right) \\
&= 1 - \mu + \mu\ln\mu - O\left(\epsilon\right)\ ,
\end{aligned}$$

where the first equality is obtained by substituting $L = \lfloor(1-\mu)\cdot C\rfloor$, and the last equality holds since $C \geq 1/\epsilon$. □

Using the above claim, it follows that $f_{\emptyset}(S_C) \geq (1-\mu-\mu\ln\mu - O(\epsilon))\cdot f_{\emptyset}(S^*) - \epsilon\cdot f(S^*)$, and since $f(\emptyset) \geq 0$, we have $f(S_C) \geq (1-\mu-\mu\ln\mu-O(\epsilon))\cdot f(S^*)$. Therefore, our algorithm attains an overall approximation ratio of $\min\{\mu, 1-\mu-\mu\ln\mu\} - O(\epsilon)$.

The latter constant is optimized by picking $\mu^* \approx 0.318$, in which case we obtain a performance guarantee of $0.318 - O(\epsilon)$.

C.3.2 Proof of Lemma 5.2.2

To show that the expected revenue function is subadditive, it is sufficient to prove that, in a given inventory vector, the deletion of any unit can only increase the probability of every other unit to be purchased. Indeed, if $U = U_1 + U_2$, starting from U, we can iteratively delete units to obtain U_1 or U_2 while increasing the consumption probabilities of all remaining units at each step. This immediately implies that the expected revenue generated by remaining units may only increase as well. Finally, by combining the units of U_1 and U_2, the total expected revenue should be at least as large as that of U.

To formalize the above statement, for any unit v stocked by some inventory vector, we use $\mathcal{C}_v$ to denote the event "unit v is consumed". With this definition, it remains to establish the following claim.

Claim C.3.3. *Let U be some inventory vector, and let U^- be a vector obtained by deleting a single unit from U. Then, for any remaining unit v stocked by U^-, we have* $\Pr[\mathcal{C}_v|U^-] \geq \Pr[\mathcal{C}_v|U]$.

Let x be the product of which one unit was deleted in order to obtain U^- from U. There are three cases:

1. The unit v belongs to product x.
2. The unit v does not belong to product x, and no additional units of x are stocked (i.e., $U_x = 1$ and $U_x^- = 0$).
3. The unit v does not belong to product x, and at least one additional unit of x is stocked (i.e., $U_x \geq 2$ and $U_x^- = U_x - 1$).

In what follows, we prove the claim for case 2, noting that the remaining cases can be proven in a nearly-identical way. Moreover, by the formula of conditional expectations, it is sufficient to establish the claim for a deterministic demand variable M. For any event E, we use $\Pr_M\left[E|U\right]$ to denote the probability of E with M arriving customers and the initial inventory vector U. Finally, to simplify the notation, we make use of product 0 to designate the no-purchase option, with preference weight $w_0 = 0$.

The proof is by induction on $\sum_{i=1}^n u_i + M$. The base case, corresponding to $\sum_{i=1}^n u_i + M = 1$, implies that $M = 0$. Hence, $\Pr_M\left[\mathcal{C}_v|U\right] = \Pr_M\left[\mathcal{C}_v|U^-\right] = 0$.

In the general case, consider the random product X picked by the first arriving customer, including the no-purchase option 0. Then,

$$\begin{aligned}
\Pr_M\left[\mathcal{C}_v|U\right] &= \Pr_M\left[X=0|U\right]\cdot\underbrace{\Pr_M\left[\mathcal{C}_v|X=0,U\right]}_{\text{(I)}}\\
&\quad+\Pr_M\left[X=x|U\right]\cdot\Pr_M\left[\mathcal{C}_v|X=x,U\right]\\
&\quad+\sum_{i\in S(U)\setminus\{x\}}\Pr_M\left[X=i|U\right]\cdot\underbrace{\Pr_M\left[\mathcal{C}_v|X=i,U\right]}_{\text{(II)}}\\
&\leq \Pr_M\left[X=0|U\right]\cdot\Pr_M\left[\mathcal{C}_v|X=0,U^-\right]\\
&\quad+\Pr_M\left[X=x|U\right]\cdot\Pr_M\left[\mathcal{C}_v|X=x,U\right]\\
&\quad+\sum_{i\in S(U)\setminus\{x\}}\Pr_M\left[X=i|U\right]\cdot\Pr_M\left[\mathcal{C}_v|X=i,U^-\right]\ . \qquad \text{(C.11)}
\end{aligned}$$

Here, we use $S(U)$ to denote the set of products stocked by the vector U, i.e., $S(U) = \{i \in [n] : u_i > 0\}$. The inequality above hold since by the induction hypothesis,

$$\text{(I)} = \Pr_M\left[\mathcal{C}_v|X=0,U\right] = \Pr_{M-1}\left[\mathcal{C}_v|U\right] \leq \Pr_{M-1}\left[\mathcal{C}_v|U^-\right] = \Pr_M\left[\mathcal{C}_v|X=0,U^-\right]\ .$$

In addition, if v is the first available unit of product i to be purchased,

$$\text{(II)} = \Pr_M\left[\mathcal{C}_v|X=i,U\right] = 1 = \Pr_M\left[\mathcal{C}_v|X=i,U^-\right]\ ,$$

and otherwise,

$$\text{(II)} = \Pr_M\left[\mathcal{C}_v|X=i,U\right] = \Pr_{M-1}\left[\mathcal{C}_v|U_{-i}\right] \leq \Pr_{M-1}\left[\mathcal{C}_v|U_{-i}^-\right] = \Pr_M\left[\mathcal{C}_v|X=i,U^-\right]\ ,$$

where U_{-i} and U^-_{-i} stand for the residual inventory vectors after a unit of product i is consumed in U and U^-, respectively. On the other hand,

$$\begin{aligned} \Pr_M\left[\mathcal{C}_v|U^-\right] &= \Pr_M\left[X=0|U^-\right]\cdot\Pr_M\left[\mathcal{C}_v|X=0,U^-\right] \\ &\quad+\sum_{i\in S(U^-)}\Pr_M\left[X=i|U^-\right]\cdot\Pr_M\left[\mathcal{C}_v|X=i,U^-\right] \ . \end{aligned} \tag{C.12}$$

To conclude the proof, note that since $S(U)=S(U^-)\uplus\{x\}$, by equation (C.11) and (C.12), we have

$$\begin{aligned} &\Pr_M\left[\mathcal{C}_v|U^-\right]-\Pr_M\left[\mathcal{C}_v|U\right] \\ &\quad\geq\left(\Pr_M\left[X=0|U^-\right]-\Pr_M\left[X=0|U\right]\right)\cdot\Pr_M\left[\mathcal{C}_v|X=0,U^-\right] \\ &\quad\;+\sum_{i\in S(U^-)}\left(\Pr_M\left[X=i|U^-\right]-\Pr_M\left[X=i|U\right]\right)\cdot\Pr_M\left[\mathcal{C}_v|X=i,U^-\right] \\ &\quad\;-\Pr_M\left[X=x|U\right]\cdot\Pr_M\left[\mathcal{C}_v|X=x,U\right] \\ &\quad\geq\Pr_{M-1}\left[\mathcal{C}_v|U^-\right]\cdot\left(\sum_{i\in S(U^-)\cup\{0\}}\Pr_M\left[X=i|U^-\right]-\sum_{i\in S(U)\cup\{0\}}\Pr_M\left[X=i|U\right]\right) \\ &\quad=0\ , \end{aligned}$$

where the first inequality holds since $\Pr_M\left[X=0|U^-\right]\geq\Pr_M\left[X=0|U\right]$ and $\Pr_M\left[X=i|U^-\right]\geq\Pr_M\left[X=i|U\right]$ by the choice probabilities of the MNL model, combined with the fact that $\Pr_M\left[\mathcal{C}_v|X=0,U^-\right]=\Pr_{M-1}\left[\mathcal{C}_v|U^-\right]$ and $\Pr_M\left[\mathcal{C}_v|X=x,U\right]=\Pr_{M-1}\left[\mathcal{C}_v|U^-\right]$, while $\Pr_M\left[\mathcal{C}_v|X=i,U^-\right]=\Pr_{M-1}\left[\mathcal{C}_v|U^-_{-i}\right]\geq\Pr_{M-1}\left[\mathcal{C}_v|U^-\right]$ due to the inductive hypothesis. The last equality proceeds from observing that the two sums of probabilities are both equal to 1.

C.3.3 Proof of Claim 5.3.3

By construction, the marginal purchase probabilities of the random variable X_j coincide with the MNL probabilities given by P_j. It remains to show that this property propagates to the random variables $X_{i,j}$, X_i, and X through the chain of conditional distributions $X_{i,j}|X_j$, $X_i|X_{i,j}$, and $X|X_i$. To avoid redundancy, we only present the proof for the variable $X_{i,j}$; those of X_i and X are based on similar ideas.

Recall that $P_{i,j}$ is the product purchased by the first arriving customer in the assortment stocked by $S \cup \{i,j\}$. Thus, we need to show that $\Pr[X_{i,j} = \alpha] = w_\alpha/(1 + w(\mathcal{A}) + w_i + w_j)$ for any product $\alpha \in \mathcal{A}^{+ij} \cup \{0\}$. For any product $\alpha \in \mathcal{A}^{+j} \cup \{0\}$, we have

$$\begin{aligned}
\Pr\left[X_{i,j} = \alpha\right] &= \sum_{\beta \in \mathcal{A}^{+j} \cup \{0\}} \Pr\left[X_j = \beta\right] \cdot \Pr\left[X_{i,j} = \alpha | X_j = \beta\right] \\
&= \Pr\left[X_j = \alpha\right] \cdot \Pr\left[X_{i,j} = \alpha | X_j = \alpha\right] \\
&= \frac{w_\alpha}{1 + w(\mathcal{A}) + w_j} \cdot \frac{1 + w(\mathcal{A}) + w_j}{1 + w(\mathcal{A}) + w_j + w_i} \\
&= \frac{w_\alpha}{1 + w(\mathcal{A}) + w_j + w_i}
\end{aligned}$$

where the second equality proceeds from equation (5.7), that guarantees $\Pr[X_{i,j} = \alpha | X_j = \beta] = 0$ for $\alpha \neq i$ and $\beta \neq \alpha$, and the next equality holds since the distribution of X_j is given by the MNL model with respect to products $\mathcal{A}^{+j} \cup \{0\}$, combined with equation (5.6). In addition,

$$\begin{aligned}
\Pr\left[X_{i,j} = i\right] &= \sum_{\beta \in \mathcal{A}^{+j} \cup \{0\}} \Pr\left[X_j = \beta\right] \cdot \Pr\left[X_{i,j} = i | X_j = \beta\right] \\
&= \frac{w_i}{1 + w(\mathcal{A}) + w_j + w_i} \cdot \sum_{\beta \in \mathcal{A}^{+j} \cup \{0\}} \Pr\left[X_j = \beta\right] \\
&= \frac{w_i}{1 + w(\mathcal{A}) + w_j + w_i}
\end{aligned}$$

where the second equality is due to our definition of $X_{i,j} | X_j$ (equation (5.5)).

C.3.4 Proof of Claim 5.3.4

To see why $(X_j | X_{i,j} = i) \sim X_j$, observe that the event $\{X_{i,j} = i\}$ is independent of the outcomes of X_j as stated by equation (5.5). Similarly, given equation (5.9) along with the equivalence $X_i \sim P_i$ shown in Claim 5.3.3, we infer that $(X_i | X_{i,j} = j) \sim X_i$.

To establish the next equivalence, $(X|X_{i,j} = i) \sim X$, observe that

$$X \sim (X|X_i = i) \sim (X|X_i = X_{i,j} = i) \sim (X|X_{i,j} = i) \ ,$$

where the first equivalence holds since the distributions of X and $X|X_i = i$ are both prescribed by the MNL model with respect to $\mathcal{A}$ (see equation (5.12) and Claim 5.3.3), and the second equivalence proceeds from the Markov property satisfied by the coupling $(X|X_i, X_{i,j}) \sim (X|X_i)$. Finally, the last equivalence follows from observing that the event $\{X_{i,j} = i\}$ is contained in $\{X_i = i\}$ due equation (5.8).

Finally, to show the equivalence $(X|X_{i,j} = j) \sim X$, we have

$$\begin{aligned}
\Pr\left[X = \alpha | X_{i,j} = j\right] &= \sum_{\beta \in \mathcal{A}^{+i} \cup \{0\}} \Pr\left[X_i = \beta | X_{i,j} = j\right] \cdot \Pr\left[X = \alpha | X_i = \beta, X_{i,j} = j\right] \\
&= \sum_{\beta \in \mathcal{A}^{+i} \cup \{0\}} \Pr\left[X_i = \beta\right] \cdot \Pr\left[X = \alpha | X_i = \beta\right] \\
&= \Pr\left[X = \alpha\right] \ ,
\end{aligned}$$

where the second equality is due to the equivalence $(X_i|X_{i,j} = j) \sim X_i$ and the Markov property.

C.3.5 Proof of Claim 5.3.9 (continued)

We begin by establishing a technical claim, useful for the upcoming analysis, whose proof is deferred to the end of this section.

Claim C.3.4. *For any subset $S \subseteq [N]$ of cardinality at most $C - 1$ and any unit $i \in [N]$,*

$$f_{M-1}(S \cup \{i\}) - f_{M-1}(S) \leq f_M(S \cup \{i\}) - f_M(S) \ .$$

We proceed with the remaining two cases: $X_{i,j} = \alpha$ where $\alpha \in \{i, j\}$.

Conditional on the event $\{X_{i,j} = i\}$. When $X_{i,j} = i$, our coupling method entails that $X_i = i$ as well due to equation (5.8). As a result,

$$\begin{aligned}
\mathbb{E}\left[R_M\left(S^{+ij}\right) - R_M\left(S^{+i}\right) \middle| X_{i,j} = i \right] &= \mathbb{E}\left[R_{M-1}\left(S^{+j}\right) - R_{M-1}\left(S\right) \right] \\
&\leq \mathbb{E}\left[R_M\left(S^{+j}\right) - R_M\left(S\right) \right] \\
&= \mathbb{E}\left[R_M\left(S^{+j}\right) - R_M\left(S\right) \middle| X_{i,j} = i \right] \ ,
\end{aligned}$$

where the first equality follows from the decomposition (5.17) by observing that the terms $r_{(X_{i,j}|X_{i,j}=i)} = r_{(X_i|X_{i,j}=i)} = r_i$ cancel out, and the next inequality holds due to Claim C.3.4. The last equality holds since $X_j|X_{i,j} = i$ and $X|X_{i,j} = i$ have the same distribution as X_j and X, respectively, as shown in Claim 5.3.4. Now, by reordering the terms in the above inequality,

$$\mathbb{E}\left[R_M\left(S^{+ij}\right) - R_M\left(S^{+j}\right) | X_{i,j} = i \right] \leq \mathbb{E}\left[R_M\left(S^{+i}\right) - R_M\left(S\right) | X_{i,j} = i \right] \ . \tag{C.13}$$

Conditional on the event $\{X_{i,j} = j\}$. In this case, our coupling method entails that $X_j = j$ as well. Indeed, using Bayes rule, equation (5.6) along with the marginal distributions of X_j and $X_{i,j}$ (see Claim 5.3.3), imply that $\Pr\left[X_j = j | X_{i,j} = j\right] = 1$. Therefore,

$$\begin{aligned}
\mathbb{E}\left[R_M\left(S^{+ij}\right) - R_M\left(S^{+j}\right) \middle| X_{i,j} = j \right] &= \mathbb{E}\left[R_{M-1}\left(S^{+i}\right) - R_{M-1}\left(S\right) \right] \\
&\leq \mathbb{E}\left[R_M\left(S^{+i}\right) - R_M\left(S\right) \right] \\
&= \mathbb{E}\left[R_M\left(S^{+i}\right) - R_M\left(S\right) \middle| X_{i,j} = j \right] , \qquad \text{(C.14)}
\end{aligned}$$

where the first equality is a consequence of (5.17) by observing that the terms $r_{(X_{i,j}|X_{i,j}=j)} = r_{(X_j|X_{i,j}=j)} = r_j$ cancel out, the next inequality follows from Claim C.3.4, and the last equality holds since $X_i|X_{i,j} = j$ and $X|X_{i,j} = j$ have the same distribution as X_i and X, respectively, by Claim 5.3.4.

Proof of Claim C.3.4. To establish the desired claim, recall that the random residual subsets of units at the k-th arrival, obtained in the proof of Lemma 5.3.6, respectively denoted by S_k and T_k when initially stocking S_1 and T_1 with $S_1 \subseteq T_1$ and $|T_1 \setminus S_1| \leq 1$, satisfy $S_k \subseteq T_k$ for every realization. In addition, using a transformation similar to that of equation (5.15), with $S_1 = S$ and $T_1 = S \cup \{i\}$, we have

$$(f_M(S \cup \{i\}) - f_M(S)) - (f_{M-1}(S \cup \{i\}) - f_{M-1}(S)) = \mathbb{E}[f_1(T_M) - f_1(S_M)] \geq 0 \ .$$

To understand the latter inequality, note that since $S_M \subseteq T_M$ for every realization, and since these subsets have cardinality at most C, we have $\mathbb{E}[f_1(T_M) - f_1(S_M)] \geq 0$ due to f_1 being restricted-non-decreasing. ■

C.3.6 Proof of Claim 5.3.10

Suppose on the contrary that there exists a product $i \in \mathcal{A}^*$ with a selling price of $r_i < \mathrm{OPT}_{\mathrm{static}} = \mathbb{E}[\mathcal{R}_1(\mathcal{A}^*)]$, where $\mathcal{R}_1(\mathcal{A})$ stands for the random revenue generated by a single customer, when the set of stocked products is $\mathcal{A}$. By calculations identical to those leading to equation (5.14),

$$\mathbb{E}[\mathcal{R}_1(\mathcal{A}^*)] = \frac{w_i}{1 + w(\mathcal{A}^*)} \cdot r_i + \left(1 - \frac{w_i}{1 + w(\mathcal{A}^*)}\right) \cdot \mathbb{E}[\mathcal{R}_1(\mathcal{A}^* \setminus \{i\})] \ .$$

In other words, $\mathbb{E}[\mathcal{R}_1(\mathcal{A}^*)]$ can be written as a convex combination of r_i and $\mathbb{E}[\mathcal{R}_1(\mathcal{A}^* \setminus \{i\})]$. Since $r_i < \mathbb{E}[\mathcal{R}_1(\mathcal{A}^*)]$, it follows that $\mathbb{E}[\mathcal{R}_1(\mathcal{A}^* \setminus \{i\})] > \mathbb{E}[\mathcal{R}_1(\mathcal{A}^*)]$, contradicting the optimality of $\mathcal{A}^*$.

C.3.7 Proof of Claim 5.3.12

The proof relies on the following technical claims regarding IFR distributions.

Lemma C.3.5 (Goyal et al. (2016)). *Let M be a non-negative integer-valued IFR random variable. For any $\alpha \in [0, 1]$, the random variable $X \sim B(M, \alpha)$ also follows an IFR distribution.*

Lemma C.3.6 (Chapter 4). *Let X be a non-negative IFR random variable, and for some constant C let $\bar{X} = \min\{X, C\}$. Suppose that $\mathbb{E}[\bar{X}] \leq \delta C$ for $\delta \in [0, 1]$. Then,*

$\mathbb{E}[\bar{X}] \geq (1-\delta) \cdot \mathbb{E}[X]$.

We argue that $\mathbb{E}[\bar{Y}_i(u_i^\propto)] \geq \mathbb{E}[Y_i]/2$ whenever $\mathbb{E}[\bar{Y}_i(u_i^\propto)] \leq u_i^\propto/2$. For this purpose, based on Lemma C.3.5, since the number of customers M is assumed to be IFR distributed, we know that $Y_i \sim B(M, \psi_i)$ follows an IFR distribution as well. As a result, by specializing Lemma C.3.6 with $\delta = 1/2$ and $C = u_i^\propto$, which is equivalent to assuming that $\mathbb{E}[\bar{Y}_i(u_i^\propto)] \leq u_i^\propto/2$, we infer that $\mathbb{E}[\bar{Y}_i(u_i^\propto)] \geq \mathbb{E}[Y_i]/2$. Therefore,

$$\mathbb{E}\left[\bar{Y}_i\left(u_i^\propto\right)\right] \quad \geq \quad \frac{1}{2} \cdot \min\left\{u_i^\propto, \mathbb{E}\left[Y_i\right]\right\} \ .$$

C.3.8 Proof of Lemma C.1.2

Let $\mathcal{H}^-$ be the set of heavy products whose selling price is less than $\epsilon^2 r_{i_{\max}}/(2n^2C)$, and $\mathcal{H}^+$ those with a selling price greater than $2n^2C \cdot r_{i_{\max}}/\epsilon^3$. Following the approach of Section 5.2.2, since the expected revenue function is subadditive (see Lemma 5.2.2), we have

$$\mathbb{E}\left[\mathcal{R}\left(U^*_{\mathcal{L}}\right)\right] + \mathbb{E}\left[\mathcal{R}\left(U^*_{\mathcal{H}^-}\right)\right] + \mathbb{E}\left[\mathcal{R}\left(U^*_{\tilde{\mathcal{H}}}\right)\right] + \mathbb{E}\left[\mathcal{R}\left(U^*_{\mathcal{H}^+}\right)\right] \geq \mathbb{E}\left[\mathcal{R}\left(U^*_{\mathcal{L}}\right)\right] + \mathbb{E}\left[\mathcal{R}\left(U^*_{\mathcal{H}}\right)\right] \geq \mathbb{E}\left[\mathcal{R}\left(U^*\right)\right] \ . \tag{C.15}$$

First, we observe that the contribution of any product $i \in \mathcal{H}^-$ toward the expected revenue of $U^*_{\mathcal{H}^-}$ is at most

$$\Pr\left[M \geq 1\right] \cdot C \cdot r_i \leq \Pr\left[M \geq 1\right] \cdot \frac{\epsilon^2 r_{i_{\max}}}{2n^2} \leq \frac{\epsilon}{n} \cdot \Pr\left[M \geq 1\right] \cdot \frac{r_{i_{\max}} w_{i_{\max}}}{1 + w_{i_{\max}}} \leq \frac{\epsilon}{n} \cdot \mathbb{E}[\mathcal{R}(U^*)] \ ,$$

where the first inequality holds by definition of $\mathcal{H}^-$, and the second inequality holds since $i_{\max}$ is a heavy product. The last inequality is obtained by observing that the optimal expected revenue $\mathbb{E}[\mathcal{R}(U^*)]$ is lower bounded by the corresponding quantity with respect to the inventory vector that stocks a single unit of product $i_{\max}$ and nothing more, which is at least $\Pr[M \geq 1] \cdot r_{i_{\max}} w_{i_{\max}}/(1 + w_{i_{\max}})$. Consequently, by summing over all products $i \in \mathcal{H}^-$, we infer that

$$\mathbb{E}\left[\mathcal{R}\left(U^*_{\mathcal{H}^-}\right)\right] \leq \epsilon \cdot \mathbb{E}\left[\mathcal{R}\left(U^*\right)\right] \ .$$

Hence, when $\mathcal{H}^+ = \emptyset$, by inequality (C.15), it follows that $\mathbb{E}[\mathcal{R}(U^*_{\mathcal{L}})] + \mathbb{E}[\mathcal{R}(U^*_{\mathcal{H}})] \geq (1-\epsilon) \cdot \mathbb{E}[\mathcal{R}(U^*)]$.

In the opposite case, when $\mathcal{H}^+ \neq \emptyset$, consider some product $i \in \mathcal{H}^+$. As before, the optimal expected revenue $\mathbb{E}[\mathcal{R}(U^*)]$ is lower bounded by the expected revenue when stocking a single unit of product i, thus we obtain

$$\begin{aligned}
\mathbb{E}\left[\mathcal{R}\left(U^*\right)\right] &\geq \Pr\left[M \geq 1\right] \cdot \frac{r_i w_i}{1+w_i} \\
&\geq \Pr\left[M \geq 1\right] \cdot \frac{nC}{\epsilon^2} \cdot r_{i_{\max}} \\
&\geq \Pr\left[M \geq 1\right] \cdot \frac{nC \cdot r_{i_1}}{\epsilon^2} \cdot \frac{w_{i_1}}{1+w_{i_1}} \cdot \frac{1+w_{i_{\max}}}{w_{i_{\max}}} \\
&\geq \Pr\left[M \geq 1\right] \cdot \frac{C \cdot r_{i_1}}{2\epsilon} \\
&\geq \frac{1}{2\epsilon} \cdot \mathbb{E}\left[\mathcal{R}\left(U^*_{\mathcal{H}}\right)\right] \ , \qquad \text{(C.16)}
\end{aligned}$$

where i_1 is the most expensive product stocked by $U^*_{\mathcal{H}}$. Here, the second inequality holds since $r_i \geq 2n^2 C \cdot r_{i_{\max}}/\epsilon^3$ and $w_i \geq \epsilon/n$, the third inequality follows by definition of $i_{\max}$ given that $r_{i_{\max}} w_{i_{\max}}/(1+w_{i_{\max}}) \geq r_{i_1} w_{i_1}/(1+w_{i_1})$, the fourth inequality holds since $w_{i_1} \geq \epsilon/n$, and the last inequality is due to the fact that r_{i_1} is the most expensive product on stock in $U^*_{\mathcal{H}}$. By combining inequality (C.15) with (C.16), we conclude that $\mathbb{E}[\mathcal{R}(U^*_{\mathcal{L}})] \geq (1-2\epsilon) \cdot \mathbb{E}[\mathcal{R}(U^*)]$.

C.3.9 Proof of Claim C.1.5

We first observe that using the formula of conditional expectation (relative to the value of M), we can restrict attention to a deterministic M. The desired inequality is proven inductively over C. For $C = 0$, we clearly have $\mathbb{E}[\bar{Y}] = \mathbb{E}[\bar{X}] = 0$.

For $C \geq 1$, by the induction hypothesis, $\bar{X}' = \min\{X, C-1\}$ and $\bar{Y}' = \min\{Y, C-1\}$ satisfy $\mathbb{E}[\bar{Y}'] \geq \theta \cdot \mathbb{E}[\bar{X}']$, and we wish to prove an analogous inequality between the expectations of $\bar{X} = \min\{X, C\}$ and $\bar{Y} = \min\{Y, C\}$. Each of the Binomial variables X and Y can be viewed as the terminating value of a Binomial process, counting the number of successes among M independent Bernoulli trials, with respective parameters α and $\theta\alpha$. We begin by defining the stopping time τ_X that corresponds to the first trial in which the Binomial process underlying the variable X, denoted by $X_1, \ldots X_M$, attains the value $C-1$. If there are fewer than $C-1$ successes among the M trials, then $\tau_X = M$. Next, observe that the expected value of X decomposes as follows:

$$\begin{aligned}
\mathbb{E}\left[\bar{X}\right] &= \mathbb{E}\left[\min\{X, C-1\} + \mathbb{I}\left[X > C-1\right]\right] \\
&= \mathbb{E}\left[\min\left\{X, C-1\right\}\right] + \Pr\left[X > C-1\right] \\
&= \mathbb{E}\left[\min\left\{X, C-1\right\}\right] + \sum_{\tau=0}^{M} \Pr\left[\tau_X = \tau\right] \cdot \Pr\left[X - X_\tau \geq 1 | \tau_X = \tau\right] \\
&= \mathbb{E}\left[\min\left\{X, C-1\right\}\right] + \sum_{\tau=0}^{M} \Pr\left[\tau_X = \tau\right] \cdot \Pr\left[X - X_\tau \geq 1\right] \\
&= \mathbb{E}\left[\min\left\{X, C-1\right\}\right] + \sum_{\tau=0}^{M} \Pr\left[\tau_X = \tau\right] \cdot \left(1 - (1-\alpha)^{M-\tau}\right) \ . \qquad \text{(C.17)}
\end{aligned}$$

The fourth equality follows from the independence of the Bernoulli trials, and the last equality holds since $X - X_\tau \sim B(M-\tau, \alpha)$. In an analogous way, τ_Y is defined as the first trial in which the Binomial process underlying the variable Y attains the value $C-1$, with $\tau_Y = M$ when $Y < C-1$. Based on the sequence of equations leading to (C.17),

$$\mathbb{E}\left[\bar{Y}\right] = \mathbb{E}\left[\min\left\{Y, C-1\right\}\right] + \sum_{\tau=0}^{M} \Pr\left[\tau_Y = \tau\right] \cdot \left(1 - (1-\theta\alpha)^{M-\tau}\right) \ . \qquad \text{(C.18)}$$

By the induction hypothesis, we already know that $\mathbb{E}[\min\left\{Y, C-1\right\}] \geq \theta \cdot \mathbb{E}[\min\left\{X, C-1\right\}]$. Thus, given (C.17) and (C.18) it remains to show that

$$\sum_{\tau=0}^{M} \Pr\left[\tau_Y = \tau\right] \cdot \left(1 - (1-\theta\alpha)^{M-\tau}\right) \geq \theta \cdot \sum_{\tau=0}^{M} \Pr\left[\tau_X = \tau\right] \cdot \left(1 - (1-\alpha)^{M-\tau}\right) \ . \quad \text{(C.19)}$$

Note that since the function $\varphi_k : x \mapsto 1 - (1-x)^k$ is concave over the interval $[0,1]$ for any $k \in \mathbb{N}$, we infer that $\varphi_k(\theta\alpha) \geq \theta \cdot \varphi_k(\alpha) + (1-\theta) \cdot \varphi_k(0) = \theta \cdot \varphi_k(\alpha)$, and therefore

$$1 - (1-\theta\alpha)^{M-\tau} \geq \theta \cdot \left(1 - (1-\alpha)^{M-\tau}\right) .$$

Hence, by observing that the right-hand side of the latter inequality is non-decreasing in τ, it is sufficient to prove that τ_X is stochastically smaller than τ_Y to derive the desired inequality (C.19). This property is easily derived by observing that the success parameter of the process $X_1, \ldots, X_M$ is lower-bounded by that of $Y_1, \ldots, Y_M$.

C.4 Tested Heuristics

Local search. The algorithm iteratively improves the objective value, where in each step a single unit is transferred from one product to the other, until reaching a local minimum. Starting with an initial inventory vector, we iteratively implement the best swap between products, i.e., one that generates the largest incremental increase in the expected revenue, evaluated through our sampling-based oracle. Specifically, letting $U^{(k)}$ denote the inventory vector obtained at the beginning of step k, a swap is represented by an ordered pair of products (i, j), where the current inventory level $u_i^{(k)}$ of product i is strictly positive. The inventory vector $U_{i \to j}^{(k)}$ resulting from this swap is derived from $U^{(k)}$ through decreasing $u_i^{(k)}$ by one unit and augmenting $u_j^{(k)}$ by one unit. With this definition, we either proceed to step $k+1$ with the inventory vector $U_{i \to j}^{(k)}$ that maximizes $\mathbb{E}[\mathcal{R}(U_{i \to j}^{(k)})]$ over all swaps (i, j), or terminate the algorithm when none of these swaps improves the expected revenue by a factor greater than 1%. To alleviate the risk of 'bad starts', the vector $U^{(1)}$ is defined by initially stocking C units of the product that maximizes $r_i w_i$, similar to Goyal et al. (2016).

Gradient-descent approach. We consider a suitable adaptation of the stochastic gradient-descent algorithm of Mahajan and van Ryzin (2001) to the MNL-based dynamic assortment planning problem. In contrast to the latter paper, here the revenue function is defined only for integer-valued inventory vectors. Hence, similar to the approach of Goyal et al. (2016), we utilize a continuous relaxation of the revenue function, defined through the Lovász extension of a discrete function. Letting $f : \mathbb{Z}^n \to \mathbb{R}$

denote the expected revenue function, its Lovász extension $\hat{f} : \mathbb{R}^n \to \mathbb{R}$ is defined as

$$\hat{f}(U) = f(\lfloor U \rfloor) + \sum_{i=1}^{n} \left(u_{\pi(i)} - u_{\pi(i-1)}\right) \cdot \left[f\left(\lfloor U \rfloor + \sum_{k=1}^{i} e_{\pi(k)} \right) - f\left(\lfloor U \rfloor + \sum_{k=1}^{i-1} e_{\pi(k)} \right) \right] ,$$

where the permutation π sorts products by the increasing fractional part of their inventory, namely, $u_{\pi(1)} - \lfloor u_{\pi(1)} \rfloor \leq \cdots \leq u_{\pi(n)} - \lfloor u_{\pi(n)} \rfloor$. The Lovász extension is piecewise linear, and its gradient can be approximately computed using the sampling-based oracle given in Appendix C.1.1.

Starting with the initial solution $U^{(0)} = 0$, and letting $U^{(k)}$ denote the solution obtained at the end of step k, each iteration consists of computing $U^{(k+1)} = \max\{0, U^{(k)} + \epsilon \nabla f(U^{(k)})\}$, where ϵ is the step size. When the latter vector does not lie in the feasible region $\{U \in \mathbb{R}^n : \|U\|_1 \leq C\}$, it is projected onto the boundary by linear rescaling. Through trial and error, we picked a step size of $\epsilon_k = \max\{0.05 \cdot C, \frac{C - \|U_k\|_1}{2}\}$. The algorithm terminates when $U^{(k+1)}$ hits the boundary (i.e., $\|U^{(k+1)}\|_1 = C$) and the objective value does not improve by a factor greater than 0.5%. Since the gradient-descent algorithm is particularly slow, we force termination after 250 iterations. Finally, it remains to 'round' the resulting inventory vector to an integral one. Suppose that $U^{(k+1)}$ is the inventory vector obtained following the gradient-descent algorithm; then $\lfloor U^{(k+1)} \rfloor$ is augmented greedily, by stocking at each step a unit of the product with maximal marginal expected revenue, until reaching C units.

Dynamic programming. With some similarities to our setting, Topaloglu (2013) studied a joint assortment and inventory problem, where the demand is formed by a Poisson arrival process. However, the problem considered is incomparable to our setting, since his formulation does not take into account stock-out substitution effects. Instead of being governed by stock-outs, the assortment dynamics is at the discretion of the retailer, who can vary the offered assortment over time to better balance stocking constraints. Still, the algorithm devised by Topaloglu (2013) is a reasonable alternative to our approach, especially since the optimal policy in his model was proven to have a compact structure, being a mixture over at most n assortments under a Poisson demand process and a single assortment under a suitable normal approximation.

In the above-mentioned model, the problem formulation is given by:

$$\begin{aligned} \max_{U,y} \quad & \sum_{i\in[n]} \left(r_i \cdot \mathbb{E}\left[\min\left\{ U_i, \text{Poisson}\left(\mathbb{E}\left[M\right] \cdot \sum_{S:i\in S} y(S) \cdot \frac{w_i}{1+w(S)} \right) \right\} \right] - c \cdot U_i \right) \\ \text{s.t.} \quad & \sum_{S\subseteq[n]} y(S) = 1 \end{aligned}$$

Here, U is the offered inventory vector, and for each possible assortment $S \subseteq [n]$ there is a corresponding decision variable $y(S)$ that describes its probability to be offered. In addition, the parameter c stands for the per-unit cost of any product. This parameter can be thought of as the Lagrangian multiplier associated with the cardinality constraint; in our setting, it can be determined through a bisection search. Now, since the objective function above is separable with respect to the products, one can cast this problem in dynamic programming terms. Specifically, we introduce the change of variable $\alpha_i = \sum_{S:i\in S} y(S) \cdot \frac{w_i}{1+w(S)}$, where α_i is the consumption rate of product i, and incorporate simple compatibility constraints between different products: $\alpha_0 + \sum_{i\in[n]} \alpha_i = 1$ and $\alpha_i \leq \frac{w_i}{w_0} \cdot \alpha_0$. At each step of the recursion, corresponding to some product $i \in [n]$, we approximately guess the consumption rate α_i, which immediately implies an optimal stocking level U_i to balance between marginal revenue and cost. We also implement the simplified recursion developed by Topaloglu (2013) under a normal approximation of the demand process. For a detailed description of these algorithms, we refer the reader to Sections 5 and 7 of his paper.

Deterministic relaxation. An additional approach that deals with stock-out substitution is the continuous-time deterministic relaxation developed by Honhon et al. (2010) and later on studied by Honhon and Seshadri (2013). Here, the stochastic nature of the choice process is overlooked. Given the initial inventory vector U and its corresponding assortment S, one assumes that each product $i \in S$ is consumed at a constant rate of $\alpha_i = w_i/(1 + w(S))$, until one of the products in S is depleted. Specifically, the first stock-out occurs at time $\min_{i \in S}(U_i/\alpha_i)$. Similarly, at the beginning of each subsequent epoch, the consumption rates are updated to reflect the changes of assortment, and the current epoch terminates at the next stock-out event. In this setting, the total consumption of products is indeed deterministic with respect to the initial stocking decisions. To optimize the latter, Honhon et al. (2010) devised a dynamic programming approach that exploits the special structure of epochs and runs in time $O(8^n)$. Due to the exponential dependency on the number of products, this approach is not applicable in our experimental setting, with $n = 20$ products. Instead, we cast the resulting deterministic model as a mixed integer program and use a state-of-the-art commercial solver (Gurobi Optimization 2015). To obtain faster convergence, the solver is given access to a warm-start solution, using the same initial inventory vector as the local search heuristic described earlier. In most cases, the solver indeed returns close-to-optimal solutions (to the relaxation) within the allowed time limit of 1000 seconds. This benchmark is informative from a modeling perspective, since it sheds light on the relative merits of using a deterministic demand process rather than the actual stochastic one.

Discrete-greedy. The discrete-greedy algorithm starts with zero inventory levels for all products, and iteratively augments the current inventory vector by a single unit of the product that incurs the largest increase in the expected revenue, until reaching C units. The expected revenue is evaluated using our sampling-based procedure. It is worth mentioning that this approach is the closest in spirit to the way our algorithm operates on heavy-expensive products, where a restricted-non-decreasing and restricted-submodular set function is approximately maximized through a greedy procedure (see Section 5.3.2).

Bibliography

Aggarwal, Gagan, Tomás Feder, Rajeev Motwani, An Zhu. 2004. Algorithms for multiproduct pricing. *Automata, Languages and Programming*. Springer, 72–83.

Ailon, Nir, Moses Charikar, Alantha Newman. 2008. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)* **55**(5) 23.

Alon, Noga, Nabil Kahale. 1998. Approximating the independence number via the theta-function. *Mathematical Programming* **80** 253–264.

Alon, Noga, Joel H. Spencer. 2004. *The probabilistic method*. John Wiley & Sons.

Anupindi, Ravi, Sachin Gupta, Munirpallam A Venkataramanan. 2009. Managing variety on the retail shelf: using household scanner panel data to rationalize assortments. *Retail Supply Chain Management*. Springer, 155–182.

Aouad, Ali, Vivek Farias, Retsef Levi, Danny Segev. 2015. The approximability of assortment optimization under ranking preferences. Working paper. Available at SSRN 2612947 (June 3rd, 2015).

Aouad, Ali, Danny Segev. 2015. Display optimization for vertically differentiated locations under multinomial logit choice preferences .

Belonax, JJ, Y Mittelstaedt. 1978. Evoked set size as a function of number of choice criteria and information variability. *Advances in Consumer Research* 48–51.

Ben-Akiva, Moshe E, Steven R Lerman. 1985. *Discrete Choice Analysis: Theory and Application to Travel Demand*. MIT Press.

Berbeglia, Gerardo, Gwenaël Joret. 2015. Assortment optimisation under a general discrete choice model: A tight analysis of revenue-ordered assortments. *Working paper. Available at SSRN 2620165 (June 19th, 2015)* .

Bertsimas, Dimitris, Velibor V Mišic. 2015. Data-driven assortment optimization. Tech. rep., Working paper, MIT Sloan School.

Bettman, James R, Mary Frances Luce, John W Payne. 1998. Constructive consumer choice processes. *Journal of Consumer Research* **25**(3) 187–217.

Bierlaire, Michel. 2003. Biogeme: a free package for the estimation of discrete choice models. *Swiss Transport Research Conference*.

Blanchet, Jose H., Guillermo Gallego, Vineet Goyal. 2016. A markov chain approximation to choice modeling. *Operations Research* **64**(4) 886–905.

Brandstatter, Eduard, Gerd Gigerenzer, Ralph Hertwig. 2006. The priority heuristic: Making choices without trade-offs. *Psychological Review* **113 (2)** 409–432.

Brisoux, Jacques E, Michel Laroche. 1981. Evoked set formation and composition: An empirical investigation under a routinized response behavior situation. *NA-Advances in Consumer Research* (8).

Bront, Juan José Miranda, Isabel Méndez-Díaz, Gustavo Vulcano. 2009. A column generation algorithm for choice-based network revenue management. *Operations Research* **57**(3) 769–784.

Campbell, Brian Milton. 1969. The existence of evoked set and determinants of its magnitude in brand choice behavior. Ph.D. thesis, Columbia University.

Chandukala, Sandeep R, Jaehwan Kim, Greg M Allenby, Thomas Otter. 2008. *Choice Models in Marketing: Economic Assumptions, Challenges and Trends*. Now Publishers Inc.

Chen, Feng, Yehuda Bassok. 2008. Variety and substitution. Working paper.

Davis, J., G. Gallego, H. Topaloglu. 2013. Assortment planning under the Multinomial Logit model with totally unimodular constraint structures. Work in Progress.

Davis, James M., Guillermo Gallego, Huseyin Topaloglu. 2014. Assortment optimization under variants of the Nested Logit model. *Operations Research* **62**(2) 250–273.

Dawes, Robyn M. 1979. The robust beauty of improper linear models in decision making. *American Psychologist* **34** 571–582.

Debreu, Gerard. 1960. Review of R. D. Luce, Individual choice behavior: A theoretical analysis **50** 186–188.

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

Désir, Antoine, Vineet Goyal, Danny Segev, Chun Ye. 2015. Capacity constrained assortment optimization under the markov chain based choice model. Working paper, available as SSRN report 2626484.

Edmonds, Jack, Rick Giles. 1977. A min-max relation for submodular functions on graphs. *Annals of Discrete Mathematics* **1** 185–204.

Einhorn, Hillel J, Robin M Hogarth. 1975. Unit weighting schemes for decision making. *Organizational Behavior and Human Performance* **13**(2) 171–192.

Farias, Vivek, Srikanth Jagabathula, Devavrat Shah. 2013. A non-parametric approach to modeling choice with limited data. *Management Science* **59**(2) 305–322.

Feige, Uriel. 1998. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM* **45**(4) 634–652.

Feldman, Jacob, Huseyin Topaloglu. 2014. Revenue management under the markov chain choice model. Working paper.

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

Fisher, Marshal. 2011. Don't trust your gut with assortment planning. Harvard Business Review.

Fisher, Marshall L., Ramnath Vaidyanathan. 2009. An algorithm and demand estimation procedure for retail assortment optimization with results from implementation. Working paper (Philadelphia: The Wharton School).

Ford, Lester Randolph. 1957. Solution of a ranking problem from binary comparisons. *American Mathematical Monthly* **64**(8) 28–33.

Gallego, Guillermo, Anran Li, Van-Anh Truong, Xinshang Wang. 2016. Approximation algorithms for product framing and pricing. Tech. rep., Working paper, available online as SSRN report.

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

Gaur, Vishal, Dorothée Honhon. 2006. Assortment planning and inventory decisions under a locational choice model. *Management Science* **52**(10) 1528–1543.

Gigerenzer, Gerd, Daniel G Goldstein. 1996. Reasoning the fast and frugal way: models of bounded rationality. *Psychological review* **103**(4) 650.

Gigerenzer, Gerd, Reinhard Selten. 2002. *Bounded rationality: The adaptive toolbox*. MIT press.

Gilbride, T. J., G. M Allenby. 2004. A choice model with conjunctive, disjunctive, and compensatory screening rules. *Marketing Science* **23**(3) 391–406.

Golrezaei, Negin, Hamid Nazerzadeh, Paat Rusmevichientong. 2014. Real-time optimization of personalized assortments. *Management Science* **60**(6) 1532–1551.

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

Grover, Rajiv, Marco Vriens. 2006. *The Handbook of Marketing Research: Uses, Misuses, and Future Advances*. Sage Publications.

Guadagni, Peter M, John DC Little. 1983. A Logit model of brand choice calibrated on scanner data. *Marketing Science* **2**(3) 203–238.

Gurobi Optimization, Inc. 2015. Gurobi optimizer reference manual. URL `http://www.gurobi.com`.

Halperin, Eran. 2002. Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs. *SIAM Journal on Computing* **31**(5) 1608–1623.

Håstad, Johan. 1996. Clique is hard to approximate within $n^{1-\epsilon}$. *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*. 627–636.

Hauser, John R. 1978. Testing the accuracy, usefulness and significance of probabilistic models: An information theoretic approach. *Operations Research* 406–421.

Hauser, John R., Min Ding, Steven P. Gaskin. 2009. Non compensatory (and compensatory) models of consideration-set decisions. Sawtooth Software Conference Proceedings.

Hauser, John R., Birger Wernerfelt. 1990. An evaluation cost model of consideration sets. *Journal of Consumer Research* 393–408.

Hess, Stephane, Michel Bierlaire, John Polak. 2007. A systematic comparison of continuous and discrete mixture models. *European Transport* (37).

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

Honhon, Dorothée, Vishal Gaur, Sridhar Seshadri. 2010. Assortment planning and inventory decisions under stockout-based substitution. *Operations Research* **58**(5) 1364–1379.

Honhon, Dorothée, Sridhar Seshadri. 2013. Fixed vs. random proportions demand models for the assortment planning problem under stockout-based substitution. *Manufacturing & Service Operations Management* **15**(3) 378–386.

Howard, J. A., J. N. Sheth. 1969. *The Theory of the Buyer Behavior*. John Wiley.

IHL. 2015. Retailers and the ghost economy: $1.75 trillion reasons to be afraid. Tech. rep., IHL Group.

Jagabathula, Srikanth. 2014. Assortment optimization under general choice .

Jagabathula, Srikanth, Paat Rusmevichientong. 2016. A nonparametric joint assortment and price choice model. *Management Science* (Articles in advance). doi: 10.1287/mnsc.2016.2491.

Karger, David R., Rajeev Motwani, Madhu Sudan. 1998. Approximate graph coloring by semidefinite programming. *Journal of The ACM* **45**(2) 246–265.

Katoh, Naoki, Toshihide Ibaraki. 1998. Resource allocation problems. *Handbook of combinatorial optimization* **2** 159–260.

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

Kök, A Gürhan, Marshall L Fisher, Ramnath Vaidyanathan. 2009. Assortment planning: Review of literature and industry practice. *Retail supply chain management*. Springer, 99–153.

Lancaster, Kelvin. 1975. Socially optimal product differentiation. *The American Economic Review* 567–585.

Lancaster, Kelvin J. 1966. A new approach to consumer theory. *The Journal of Political Economy* 132–157.

Laroche, Michel, Chankon Kim, Takayoshi Matsui. 2003. Which decision heuristics are used in consideration set formation? *Journal of Consumer Marketing* **20**(3) 192–209.

Li, Guang, Paat Rusmevichientong, Huseyin Topaloglu. 2015. The d-level nested logit model: Assortment and price optimization problems. *Operations Research* **63**(2) 325–342.

Liu, Qian, Garrett Van Ryzin. 2008. On the choice-based linear programming model for network revenue management. *Manufacturing & Service Operations Management* **10**(2) 288–310.

Luce, Robert Ducan. 1959. *Individual Choice Behavior a Theoretical Analysis*. John Wiley & Sons.

Mahajan, Siddharth, Garrett van Ryzin. 2001. Stocking retail assortments under dynamic consumer substitution. *Operations Research* **49**(3) 334–351.

Maystre, Lucas, Matthias Grossglauser. 2015. Fast and accurate inference of Plackett–Luce models. *Advances in Neural Information Processing Systems*. 172–180.

McBride, Richard D, Fred S Zufryden. 1988. An integer programming approach to the optimal product line selection problem. *Marketing Science* **7**(2) 126–140.

McFadden, Daniel. 1973. Conditional Logit analysis of qualitative choice behavior. *Frontiers in Econometrics* 105–142.

McFadden, Daniel. 1980. Econometric models for probabilistic choice among products. *Journal of Business* **53**(3) S13–S29.

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

Megiddo, Nimrod. 1979. Combinatorial optimization with rational objective functions. *Mathematics of Operations Research* **4**(4) 414–424.

Méndez-Díaz, Isabel, Juan José Miranda-Bront, Gustavo Vulcano, Paula Zabala. 2014. A branch-and-cut algorithm for the latent-class Logit assortment problem. *Discrete Applied Mathematics* **164** 246–263.

Muckstadt, John A, Amar Sapra. 2010. *Principles of Inventory Management: When You Are Down to Four, Order More*. Springer Science & Business Media.

Nagarajan, Mahesh, Sampath Rajagopalan. 2008. Inventory models for substitutable products: optimal policies and heuristics. *Management Science* **54**(8) 1453–1466.

Negahban, Sahand, Sewoong Oh, Devavrat Shah. 2012. Iterative ranking from pair-wise comparisons. *Advances in Neural Information Processing Systems*. 2474–2482.

Nemhauser, George, Laurence Wolsey, Marshall Fisher. 1978. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming* **14**(1) 265–294.

Parkinson, T. L., M. Reilly. 1979. An information processing approach to evoked set formation. *Advances in Consumer Research* **6**(1) 227–231.

Payne, J. W., J. Bettman, R. James, M. F Luce. 1996. When time is money: Decision behavior under opportunity-cost time pressure. *Organizational behavior and human decision processes* **66**(2) 131–152.

Pentico, David W. 1974. The assortment problem with probabilistic demands. *Management Science* **21**(3) 286–290.

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

Posavac, Steven S., Tracy Meyer, Frank R. Kardes, James J. Kellaris. 2005. A selective hypothesis testing perspective on price-quality inference and inference-based choice. *Journal of Consumer Psychology* **15 (2)** 159–169.

Pras, Bernard, John Summers. 1975. A comparison of linear and nonlinear evaluation process models. *Journal of Marketing Research* 276–281.

Ratliff, Richard M, B Venkateshwara Rao, Chittur P Narayan, Kartik Yellepeddi. 2008. A multi-flight recapture heuristic for estimating unconstrained demand from airline bookings. *Journal of Revenue and Pricing Management* **7**(2) 153–171.

Reilly, Michael, Thomas L Parkinson. 1985. Individual and product correlates of evoked set size for consumer package goods. *Advances in Consumer Research* **12**.

Roberts, John H, James M Lattin. 1991. Development and testing of a model of consideration set composition. *Journal of Marketing Research* 429–440.

Rusmevichiengtong, Paat, Benjamin Van Roy, Peter W. Glynn. 2006. Nonparametric approach to multiproduct pricing. *Operations Research* **54**(1) 82–98.

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

Rusmevichientong, Paat, David Shmoys, Chaoxu Tong, Huseyin Topaloglu. 2014. Assortment optimization under the multinomial logit model with random choice parameters. *Production and Operations Management* **23**(11) 2023–2039.

Rusmevichientong, Paat, Huseyin Topaloglu. 2012. Robust assortment optimization in revenue management under the Multinomial Logit choice model. *Operations Research* **60**(4) 865–882.

Rusmevichientong, Paat, Benjamin Van Roy, Peter W. Glynn. 2006. A nonparametric approach to multiproduct pricing. *Operations Research* **54**(1) 82–98.

Ryzin, Garrett van, Siddharth Mahajan. 1999. On the relationship between inventory costs and variety benefits in retail assortments. *Management Science* **45**(11) 1496–1509.

Samet, Hanan. 1990. Applications of spatial data structures. Addison-Wesley.

Sauré, Denis, Assaf Zeevi. 2013. Optimal dynamic assortment planning with demand learning. *Manufacturing & Service Operations Management* **15**(3) 387–404.

Segev, Danny. 2015. Assortment planning with nested preferences: Dynamic programming with distributions as states? Working paper, available as SSRN report #2587440.

Shaked, M., J.G. Shanthikumar. 1994. *Stochastic Orders and Their Applications*. Academic Press, New York.

Silk, Alvin J, Glen L Urban. 1978. Pre-test-market evaluation of new packaged goods: A model and measurement methodology. *Journal of marketing Research* 171–191.

Sinha, Ashish, Anna Sahgal, Sharat K Mathur. 2013. Practice prize paper—category optimizer: A dynamic-assortment, new-product-introduction, mix-optimization, and demand-planning system. *Marketing Science* **32**(2) 221–228.

Smith, Stephen A., Narendra Agrawal. 2000. Management of multi-item retail inventory systems with demand substitution. *Operations Research* **48**(1) 50–64.

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

Talluri, Kalyan T, Garrett J Van Ryzin. 2006. *The Theory and Practice of Revenue Management*, vol. 68. Springer Science & Business Media.

Topaloglu, Huseyin. 2013. Joint stocking and product offer decisions under the multinomial logit model. *Production and Operations Management* **22**(5) 1182–1199.

Tversky, A. 1972a. Choice by elimination. *Journal of Mathematical Psychology* **9** 341–367.

Tversky, A. 1972b. Elimination by aspects : A theory of choice. *Psychological Review* **79** 281–299.

Tversky, A., S. Sattath. 1979. Preference trees. *Psychological Review* **86** 542–573.

Tversky, Amos, Daniel Kahneman. 1975. Judgment under uncertainty: Heuristics and biases. *Utility, probability, and human decision making*, vol. 185. Springer, 141–163.

Urban, Glen L. 1975. Perceptor: A model for product positioning. *Management Science* **21**(8) 858–871.

van Ryzin, Garrett, Gustavo Vulcano. 2014. A market discovery algorithm to estimate a general class of nonparametric choice models. *Management Science* **61**(2) 281–300.

Vulcano, Gustavo, Garrett van Ryzin, Wassim Chaar. 2010. Choice-based revenue management: An empirical study of estimation and optimization. *Manufacturing & Service Operations Management* **12**(3) 371–392.

Wächter, Andreas, Lorenz T Biegler. 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming* **106**(1) 25–57.

Zeithalm, Valarie A. 1988. Consumer perception of price, quality and value: a means-end model and synthesis of evidence. *Journal of Marketing* **52** 2–22.