Fundamental Tradeoffs for Modeling Customer Preferences in Revenue Management

Antoine Désir

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2017

ABSTRACT

Fundamental Tradeoffs for Modeling Customer Preferences in Revenue Management

Antoine Désir

Revenue management (RM) is the science of selling the right product, to the right person, at the right price. A key to the success of RM, which now spans a broad array of industries, is its grounding in mathematical modeling and analytics. This dissertation contributes to the development of new RM tools by: (1) exploring some fundamental tradeoffs underlying any RM problems, and (2) designing efficient algorithms for some RM applications. Another underlying theme of this dissertation is the modeling of customer preferences, a key component of any RM problem.

The first chapters of this dissertation focus on the model selection problem: many demand models are available but picking the right model is a challenging task. In particular, we explore the tension between the richness of a model and its tractability. To quantify this tradeoff, we focus on the assortment optimization problem, a very general and core RM problem. To capture customer preferences in this context, we use *choice models*, a particular type of demand model. In Chapters 1, 2, 3 and 4 we design efficient algorithms for the assortment optimization problem under different choice models. By assessing the strengths and weaknesses of different choice models, we can quantify the cost in tractability one has to pay for better predictive power. This in turn leads to a better understanding of the tradeoffs underlying the model selection problem.

In Chapter 5, we focus on a different question underlying any RM problem: choosing how to sell a given product. We illustrate this tradeoff by focusing on the problem of selling ad impressions via Internet display advertising platforms. In particular, we study how the presence of risk-averse buyers affects the desire for reservation contracts over real time buy via a second-price auction. In order to capture the risk aversion of buyers, we study different utility models.

Contents

List of Figures **iv**

List of Tables **vi**

Acknowledgements **vii**

Introduction **1**

1 Choice models and assortment optimization **4**

1.1 Choice models: introduction and taxonomy 4

1.2 Fundamental tradeoffs in model selection 13

1.3 The assortment optimization problem 14

1.4 Summary of contributions of Chapters 2, 3 and 4 16

2 Near optimal algorithms for capacity constrained assortment under random utility models **24**

2.1 Multinomial logit model 25

2.2 Nested logit model 30

2.3 d-level nested logit 38

2.4 Mixtures of multinomial logit model 41

3 Approximation algorithms for assortment optimization problems under a Markov chain based choice model **47**

3.1 Markov chain model 49

3.2 Hardness of approximation 50
3.3 Local ratio based algorithm design 53
3.4 Unconstrained assortment optimization 64
3.5 Cardinality constrained assortment optimization 66
3.6 Capacity constrained assortment optimization 73
3.7 Computational experiments 77
3.8 Robust assortment optimization 81
3.9 Near optimal algorithm under constant rank 92

4 Mallows-smoothed distribution over rankings approach for modeling choice 96
4.1 Model and problem statement 98
4.2 Choice probabilities: closed-form expression 101
4.3 A PTAS for the assortment optimization 104
4.4 Integer programming formulation 109
4.5 Numerical experiments 114

5 Design of Futures Contract for Risk-averse Online Advertisers 118
5.1 Introduction 118
5.2 Posted price 128
5.3 Auctions 138
5.4 Computational experiments 152

Bibliography 158

Appendices 164

A Near optimal algorithms for capacity constrained assortment under random utility models 165
A.1 FPTAS for mMNL-Capa 165

A.2 Assumption of $u_{i,k} > 0$ in mMNL-Capa 168

A.3 Proof of Theorem 2.8 169

B Approximation algorithms for assortment optimization problems under a Markov chain based choice model 172

B.1 Proof of Theorem 3.2 172

B.2 Proof of Lemma 3.2 174

B.3 Proof of Lemma 3.4 175

B.4 Proof of Lemma 3.6 175

B.5 Application of Algorithm 6 to MNL 176

B.6 FPTAS for MC-Capa under rank one assumption 177

C Mallows-smoothed distribution over rankings approach for modeling choice 181

C.1 Proof of Theorem 4.2 (continued) 181

List of Figures

1.1 Substitution behavior under MNL model. 10

1.2 Substitution behavior under MC model. 11

3.1 A bad example for Algorithm 4. 56

3.2 A bad example for Algorithm 5. 58

3.3 Instance update in local-ratio algorithm. 62

3.4 A tight example for Algorithm 8. 72

3.5 A tight example for Algorithm 10. 77

5.1 Relative improvement in the seller revenue by adding the Market-Maker contract for for the setting with auctions and homogeneous risk aversion. 153

5.2 Relative improvement in the seller revenue and the sum of buyer utilities by adding the Market-Maker contract for the setting with auctions and heterogeneous risk aversions. 155

5.3 Relative changes in the buyer utilities by adding the Market-Maker contract for the setting with auctions and heterogeneous risk aversions. . . . 155

5.4 Welfare performance of the posted price for different values of β. 156

5.5 Relative improvement in the seller revenue and the sum of buyer utilities by adding the Market-Maker contract. 157

B.1 Sketch of our construction for an instance on 4 items, where $L_1 = (1 \succ 2 \succ 3 \succ 4)$, $L_2 = (1 \succ 3 \succ 4)$, $L_3 = (2 \succ 3)$, and $L_4 = (1 \succ 2 \succ 4)$. Note, for example, that the state $(2, 2)$ corresponds to the second item of L_2, but actually corresponds to item 3. 173

List of Tables

1.1 Mean Absolute Percentage Error (MAPE) of various models on the Sushi data set. 9
1.2 Abbreviations for different choice models 17
1.3 Abbreviations for various assortment problems 18
1.4 Summary of contributions 23

3.1 Performance of Algorithm 8 for MC-Card. 80
3.2 Running time of Algorithm 8 and the MIP for setting 2. ** Denotes the cases when we set a time limit of 2 hours. 81

4.1 Running time of the strengthened MIP for various values of $e^{-\theta}$ and n. (**the solver did not terminate in 8 hours) 116

Acknowledgements

Although research can be lonely sometimes, I didn't complete my phd alone and I have many people to thank. First and foremost, I would like to thank my advisor, Vineet for his support throughout the years. Thank you, Vineet, for introducing me to interesting and challenging problems, spending countless hours with me trying to crack them, helping me write my first paper and prepare my first talk. You have shaped the researcher I am today and for that I am really grateful.

I would like to thank Garud, my co-advisor. He gave me my first internship at the end of my first year, a class to teach the following year, which I think has been an important factor in attracting me to academia, and has been involved in my research ever since. I think our department is lucky to have him as our chair.

I want to thank the other members of my committee. Srikanth, it's been a pleasure working with you. Thank you for letting me sit in your class at NYU and coaching me through the job market. Huseyin, your research has always inspired my work. I remember after my first talk at INFORMS that Vineet introduced me to you and said: "That's Huseyin, you know the one you cite on all your slides". I am glad I only got to know you were attending my talk afterwards! I want to thank Omar for being a great mentor especially during my job market year. I'll always remember the mock interview you gave me before INFORMS. You pushed me when I needed it and I am grateful for that.

Soulaymane, thank you for taking me as your TA for all these years. It was a very interesting experience and I'll always remember the Dr.Phil sessions for the presentation sessions which always had a very uncertain end time. I want to thank all the faculty of the department for the many great classes I took here. Thank you to the staff: Adina, Darbi, Jenny, Carmen, Jaya, Krupa, Kristen, Sam, Shi Yee, Yosimir. You keep this place together and make our life so much easier. Thank you for that and a particular thanks to Liz for all the energy she brings to the department. It's good to know that the incoming phds are in good hands.

I want to thank all my co-authors. Thank you Jiawei and Yehua. Although our work on the long chain design does not appear in this dissertation, it represents a good amount of my time as a PhD student and a very fruitful collaboration. Thank you Danny for teaching me how to write a paper. Thank you Chun. It was a great a pleasure to work with you and to turn this class project into a paper together. Finally, thank you Balu, Nitish and Maxime for being great hosts and collaborators during my stay at Google.

Thank you to all the phd students who have made my life here a lot more fun. Thank you to all the friends who are not here anymore: Tulia, Matt, Chun, Aya, Arseny, JB, Brian... and to all the ones that are still here: Francois, Octavio, Mauro, Omar, Vashist, Irene, the great IEOR basketball team... I made some great friendships here and I know that we'll meet again. A special mention to Gonzalo who has always been my partner in crime. We started together five years ago and a few weeks back we walked together. I never imagined, when we were studying for the quals together in our first year, that you and Fernanda would be in Paris to attend my wedding a few years later and the first ones to hug me after the birth of my son another few years forward.

Last but not least, from the bottom of my heart, I want to thank my wife. I cannot think about my phd journey without thinking about our own journey together. When I started my phd, you were my girlfriend. You became my wife during my third year and this year we became parents together. These past years, and especially this last

year, have been very intense and stressful and you have always been there through thick and thin. Thank you for being my rock and for bringing balance to my life. I would never have been able to do this without you.

To my wife and my son

Introduction

Revenue management (RM) is the science, some would say the art, of selling the right product, to the right person, at the right price. The delicate task of RM is to allocate a finite inventory of products to some uncertain demand and is most of the time addressed by carefully modeling the problem at stake and casting it into a well formulated optimization problem. Analyzing such problem and providing efficient solutions is the crux of RM and what has lead to helping practitioners make better decisions. RM now spans across a broad array of industries and the tools of RM have been used to optimally sell airline tickets, hotel rooms, fashion goods and more recently online advertisements. My dissertation contributes to the development of RM technologies by applying mathematical modeling and analytics to different RM problems with an aim to: (1) quantify fundamental tradeoffs, and (2) design efficient algorithms to find (near)-optimal solutions. An underlying theme of this dissertation is the modeling of customer preferences, a key component of any RM problem. Chapters 1, 2, 3 and 4 explore discrete choice models which aim at predicting customer choices when faced with a set of different alternatives. Chapter 5 studies the presence of risk-aversion in customers preferences and uses various utility models to capture such behavior.

Choice model and assortment optimization. For a given problem, many demand models can be used. Deciding on the right model is a complex task. In Chapters 1, 2, 3 and 4, we study the fundamental tradeoffs underlying the model selection problem. In particular, we focus on the tension between expressiveness and tractability of a model. The richness of the model allows capturing fine nuances of customer behavior. On the other hand, looking at the tractability of the model is equally important: does this model lead to a mathematical model that can be solved efficiently? Typically, simple models, from the predictive standpoint, lead to easy problems, from the tractability standpoint. On the other hand, rich models lead to hard problems. To explore these tradeoffs, we focus on a core RM problem known as the assortment optimization problem. In this problem, the decision maker needs to decide on a subset of products to offer arriving customers in order to maximize expected revenue. In this RM problem, the prices are assumed to be given and the decision maker's lever is to decide which products to offer. For example, this situation holds in the context of airline tickets, where a menu of fares is designed to allow the same capacity to be sold at different prices. By nature, this is a hard combinatorial problem as the number of possible offer sets grows exponentially with the number of products. Moreover, the choice of demand model heavily affects the tractability of the assortment optimization problem. Because of the nature of the problem, we use particular demand models known as *choice models*. By designing efficient algorithms for the assortment optimization problem under various choice models, we quantify the cost in tractability one has to pay for better predictive power. Thus, we assess the strengths and weaknesses of different choice models which lead to better understanding of the tradeoffs underlying the model selection problem. Chapter 1 provides an introduction to choice models and assortment optimization. It introduces three main families of model. Chapters 2, 3 and 4 are then each devoted to one paticular model.

Risk averse buyers in online advertising. In Chapter 5, we do not assume that the selling mechanism is fixed but rather explore a different tradeoff in RM, that of choosing how to sell a given product. We illustrate this tradeoff by focusing on the problem of selling ad impressions via Internet display advertising platforms.

Advertisers' buying choices typically include two options: either they commit to a reservation contract in advance or they buy programatically in real time via an exchange. The former case is a manual, time-consuming, and expensive process which comes with a guarantee on the impressions. In the latter case, advertisers typically bid in a second-price auction and they may therefore experience significant allocation uncertainty stemming from the randomness in the number of advertisers participating in the auction as well as the uncertainty in their valuation. Furthermore, the second-price auctions comes with a price uncertainty. In contrast, reservation contracts provide price and allocation guarantees. In Chapter 5, we study how the presence of risk-averse buyers affects the desire for guarantees as well as how to price such reservation contracts. In order to capture the risk aversion of buyers, we use different utility models. This chapter is based on the work done during a research internship at Google NYC.

Chapter 1

Choice models and assortment optimization

1.1 Choice models: introduction and taxonomy

Choice is ubiquitous and pervades everyday life. Am I in the mood for thai food or sushi tonight? Would this black shirt look better on me than this blue one? Should I take the subway or a taxi? Who should I vote for? We make choices multiple times a day. Not surprisingly, trying to model how we choose among possible offered options has been a fundamental topic of research in many different academic fields including marketing, transportation, economics, psychology and operations management.

In many applications, our choice heavily depends on the menu of available options. Did you take this cab because the subway was not running? What happens when your favorite coffee brand is stocked out at the grocery store? Do you buy another brand or do you walk out without anything? Underlying our choice is the *substitution effect*: when our most preferred option is not available we substitute to another option. Modeling this phenomenon is at the heart of the theory of discrete choice modeling which we now discuss. To make things concrete and because of the focus on revenue management applications, we will refer to these options as products and we will think about modeling how customers choose among different offered products. However it should be clear that these models have much broader applications.

Unlike traditional demand models, *choice models* make the demand for each product a function of the entire offer set. This flexibility allows capturing behaviors such as the substitution effect but also significantly increases the complexity of the demand model. Mathematically, a choice model specifies customer preferences in the form of a probability distribution over products in a subset. More precisely, the choice model will be defined by the following choice probabilities:

$$\pi(i, S) = \Pr(\text{customer selects product } i \text{ from offer set } S),$$

where we assume that we have a universe $\mathcal{N}$ consisting of n products such that $i \in \mathcal{N}$ and $S \subseteq \mathcal{N}$. We refer to $\pi(i, S)$ as a *choice probability.* This quantity can equivalently be thought of as the probability that some random customer chooses product i when the offer set is S or as the fraction of customers who will choose product i if the subset S is offered. Such a model allows us to model the substitution effect. For example, having $\pi(i, S) > \pi(i, S \cup \{j\})$ captures a cannibalization of product i by product j: when j is offered, the demand for product i drops. However, this flexibility comes at a cost. Indeed, note that such a model needs to specify the demand of each product for each of the 2^n possible subset $S \subseteq \mathcal{N}$. The theory of discrete choice modeling provides more parsimonious descriptions of these models by adding some assumptions on the form of the choice probabilities. In this dissertation, we study three main families of choice models: random utility models, a Markov chain based choice model and distributions over rankings. Each of these models addresses the modeling of customer preferences in a distinct fashion. Classical economic theory postulates that individuals select an alternative by assigning a utility to each option and selecting the alternative with the maximum utility. This is the basis for the family of random utility models which we study in Chapter 2. More recently, different approaches coming from the operations literature have emerged. The other two models that we consider, a Markov chain based model in Chapter 3 and distribution over rankings in Chapter 4, belong to this stream. We now give a brief literature review for each of these three types of model where we try to highlight how these models relate to each other. We do not introduce the mathematical details of each model and postpone this to their corresponding chapter.

1.1.1 Random utility models

The class of random utility maximization (RUM) models was formally introduced by Nobel prize winner economist Daniel McFadden [53]. They have a long history and have been extensively studied in the literature in several areas including marketing, transportation, economics and operations management (see [54], [8]). In this framework, each customer assigns a random utility U_i to each product i. When the utilities are realized, he/she then chooses the product which maximizes his/her utility among all offered products. More formally, the choice probabilities take the following form under this framework:

$$\pi(i, S) = \Pr(U_i = \max_{j \in S} U_j).$$

Specifying the joint distribution of the random variables U_i generates different RUM models.

Multinomial logit model. The multinomial logit (MNL) model has by far been the most popular model in practice. It was introduced independently by Luce [50] and Plackett [62] and was referred to as the Plackett-Luce model. It came to be known as the MNL model after the work of McFadden [53] who gave it this modern interpretation through the lens of RUM theory. Indeed, the MNL model is an RUM model where the random utilities U_i are assumed to be i.i.d. across products and distributed according to a Gumbel distribution.

Informally, the MNL model assigns a score to each product. Each product is then chosen with probability proportional to its score. This simplicity makes the expression of the choice probabilities very easy to write down but also limits the ability of the model to faithfully capture complex substitution patterns present in various applications. In particular, a commonly recognized limitation of the MNL model is the so-called "Independent of Irrelevant Alternatives" (IIA) property (see [8]), which specifies that the odds of choosing among two products are not affected by the presence of a third product. Recognizing these limitations, researchers have proposed more complex models to capture a richer class of substitution behaviors. We now discuss two such models which uses the MNL model as a building block.

Nested logit model. In a nested logit (NL) model, the products are clustered into different nests. Customers first choose a nest and then choose among products in the chosen nest according to an MNL model. The NL model was introduced by Williams [75] and its justification as a RUM model was later provided in [11]. The NL model alleviates the IIA property by introducing some some correlation between the utilities of products in the same nest. More recently, [48] introduce a generalization of this model called the d-level nested logit (dNL) model. In the same fashion, customers now choose a particular nest by going down a decision tree of depth d. These models are particularly interesting when some predefined nest structure exists on the set of products as it is unclear how to learn the nest structure of these models efficiently.

Mixture of MNL model. Another approach to breaking the IIA property is assuming that there are several classes of customers, each of which choosing according to a different MNL model. Such mixture of MNL (mMNL) model (also sometimes referred to as mixed logit) was introduced in [55] where the authors show that any choice model arising from the random utility framework can be approximated as closely as required by a mixture of a finite (but unknown) number of MNL models. This makes the mMNL model the most general model in the class of RUM models.

There are other RUM models that we do not consider in this work such as the exponomial model [2] and refer to [72] for a detailed overview of these models. We now turn to two different approaches to generating choice model coming from the operations literature.

1.1.2 Markov chain model

Introduced in [10], the main idea motivating the Markov chain (MC) model is to model a customer's choice by explicitly modeling his substitution behavior. Here, customer substitution is captured by a Markov chain, where each product corresponds to a state of the Markov chain, and substitutions are modeled using transitions in the Markov chain. Given an offer set, the states corresponding to the offered products become absorbing. A random customer arrives to each product according to some arrival probabilities. Upon arrival, the customer chooses the product if offered. Otherwise, the customer then substitutes according to the underlying transition probabilities of the Markov chain and continues to do so until he reaches an offer product. At this point, he chooses that product. In other words, in order to determine the chosen product for some random customer, we perform a random walk on the Markov chain and stop when we first hit one of the absorbing state. The corresponding product is chosen. Under this model, we can reformulate the choice probabilities as:

$$\pi(i, S) = \text{Pr}(\text{customer gets absorbed in state } i \text{ when subset } S \text{ of nodes is absorbing}).$$

Interestingly, despite being motivated from a completely different point of view, a salient feature of the MC model is that it generalizes several known model (see [10]) including MNL, generalized attraction model [33], and the exogenous demand model [43]. Moreover, [10] show that the MC model provides a good approximation in choice probabilities to the class of RUM.

Interpretability of parameters. Another very interesting feature of this model is that its parameters have a very nice interpretation as they directly model substitutions. To illustrate this, we use a publicly available data set consisting of preference lists over different sushi types. The Sushi data set consists of 5,000 complete rankings over 10 varieties of sushi (http://www.kamishima.net/sushi/ [42]). Each ranking corresponds to the preferences of one person who was asked to rank the different types of sushis. We use 1,500 rankings for training and 3,500 rankings for validation. In particular, we fit a MC model (using the procedure described in [10]) and a simple MNL model on the training samples. Using those fitted models, we compute the choice probabilities over all possible subsets and compare them to the choice probabilities computed over the 3,500 validation rankings. We report the average error in choice probabilities in Table 1.1. We also report the average error made by the empirical distribution (ED) (on the 1,500 training rankings). The improvement that

Model	MNL	MC	ED
MAPE	15.8 %	8.3 %	6.9 %

Table 1.1: Mean Absolute Percentage Error (MAPE) of various models on the Sushi data set.

we observe using the MC model over the MNL model is significant: the average error is almost reduce by half. Moreover, the error in prediction using the MC model is quite close to the error of the ED. However, the really interesting part consists at looking at the fitted parameters of the MC model. To highlight the flexibility of the MC model, we contrast it with a simple MNL model (a special case of MC model). Figures 1.1 and 1.2 show the parameters of the fitted models in the form of a matrix where each entry of the matrix corresponds to the transition probability of the underlying Markov chain. For instance, the cell at the intersection of the row "tuna" and "shrimp" represents the probability of substituting from tuna to shrimp. The color represents the intensity of the substitution.

First note that the MNL model only allows a very limited behavior. This is a consequence of the IIA property. In particular, the substitutions under an MNL model are independent of the product we are substituting from. Hence, all the columns of the matrix have the same color. Secondly, the gradient of color, from left to right, indicates that the strength of the substitution is dictated by the popularity or market share of a product: the sushi are ordered by popularity on each axis.

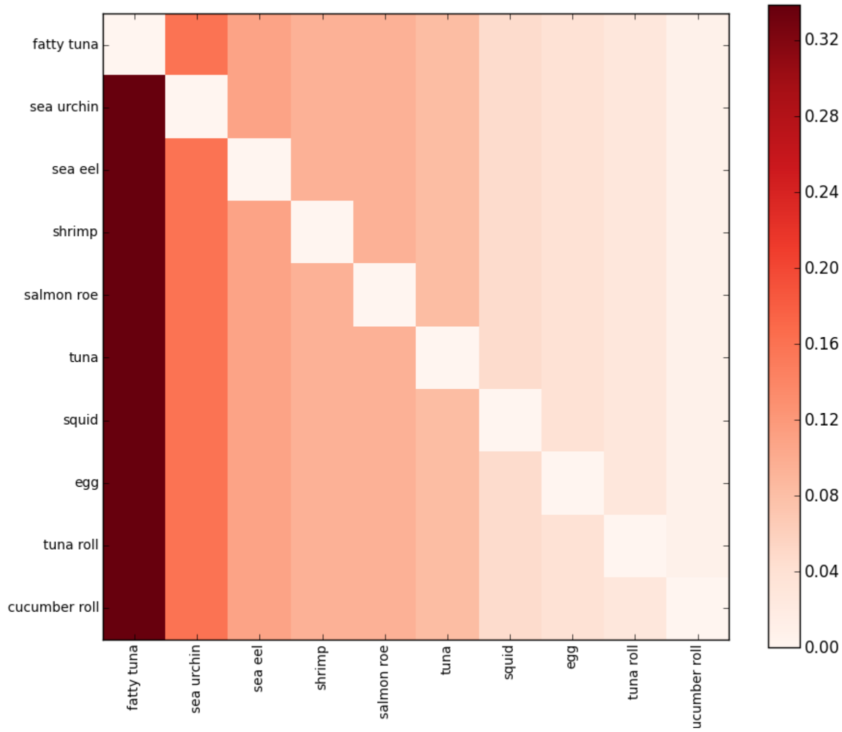


Figure 1.1: Substitution behavior under MNL model.

Now, we turn to the matrix representing the MC model (Figure 1.2). We immediately observe that the captured behavior is much richer. Moreover, several interesting phenomenon are captured. First of all, we observe that all the tuna variations of sushis (fatty tuna, tuna, tuna roll) exhibit strong mutual substitutions. For instance, there is a much higher substitution from fatty tuna to tuna than to any other type of sushis. Similarly, the substitution from tuna roll is highest towards fatty tuna and tuna. This is particularly helpful as we can detect clusters of products customer tend to substitute among just by looking at the parameters of the fitted model. Another interesting phenomenon is the behavior toward the sea urchin sushi, a very atypical sushi. Note that the substitution to the sea urchin sushi are relatively low despite the sea urchin being the second most popular sushi. This is because people tend to exhibit very strong preferences for this sushi: they either rank it first or last, i.e. they

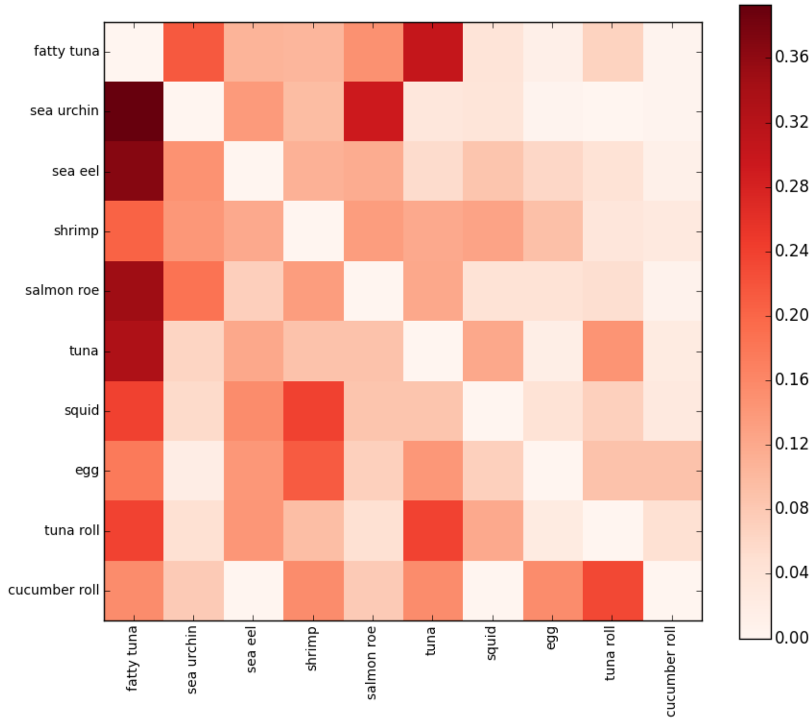


Figure 1.2: Substitution behavior under MC model.

do not substitute to the sea urchin. Note that this phenomenon cannot be captured by a simple MNL model since by IIA, the substitution has to be proportional to the popularity.

1.1.3 Distribution over rankings

In the most general case, a choice model is given by a distribution over preference lists or rankings [26, 73, 36]. A preference list is a ranked ordering of the products of $\mathcal{N}$. Given an offered subset of products, when a random customer arrives, a preference list is sampled from the distribution. The customer then purchases his most preferred item from the offered products using the sampled preference list.

$$\pi(i, S) = \Pr(\text{product } i \text{ is ranked first among product in } S).$$

The rank-based model is very general and accommodates distributions with exponentially large support sizes and, therefore, can capture complex substitution patterns. However, available data are usually not sufficient to identify such a complex model. Therefore, sparsity is used as a model selection criterion to pick a model from the set of models consistent with the data. Specifically, it is assumed that the distribution has a support size K, for some K that is polynomial in the number of products. Sparsity results in data-driven model selection [26], obliviating the need for imposing arbitrary parametric structures.

The need for smoothing. Despite their generality, however, sparse rank-based models cannot account for noise or any deviations from the K ranked-lists in the support. This limits their modeling flexibility, resulting in unrealistic predictions and inability to model individual-level observations. Specifically, because $K \ll n!$, the model specifies that there is a zero chance that a customer uses a ranking that is even slightly different from any of the K rankings in the support and a zero chance of observing certain choices. However, choices may be observed in real (holdout) data that are not consistent with any of the K rankings, making the model predictions unrealistic. In addition, a natural way to interpret sparse choice models is to assume that the population consists of K types of customers, with each type described by one of the ranked lists. When this interpretation is applied to individual-level observations, it implies that all the choice observations of each individual must be consistent with at least one of the K rankings, which again may not be the case in real data.

Mallows-smoothed model. In order to address these issues, we generalize the sparse rank-based models by *smoothing* them using the Mallows kernel. Specifically, we suppose that the choice model is a mixture of K Mallows models.

The Mallows distribution was introduced in the mid-1950's [51] and is the most popular member of the so-called distance-based ranking models, which are characterized by a modal ranking ω and a concentration parameter θ. The probability that a ranking σ is sampled falls exponentially as $e^{-\theta \cdot d(\sigma,\omega)}$, where $d(\cdot,\cdot)$ is the distance between σ and ω. Different distance functions result in different models. The Mallows model uses the Kendall-Tau distance, which measures the number of pairwise disagreements between the two rankings. Intuitively, the Mallows model assumes that consumer preferences are concentrated around a central permutation, with the likelihood of large deviations being low. The mixture of Mallows model with K segments is specified by the modal rankings: $\omega_1, \ldots, \omega_K$, concentration parameters: $\theta_1, \ldots, \theta_K$ and probabilities: $\mu_1, \ldots, \mu_K$ where for any $k = 1, \ldots, K$, μ_k specifies the probability that a random customer belongs to Mallows segment k with modal ranking ω_k and concentration parameter θ_k. This mixture model is a more natural model allowing for deviations from the modal rankings and assigning a non-zero probability to every choice. Further, it is a parsimonious way to extend the support of the distribution to an exponential size, and as $\theta_k \to \infty$ for all k, the distribution concentrates around each of the K modes, yielding the sparse rank-based model. We refer the interested readers to a large body of existing work in the literature on estimating such models from data [49, 4, 22, 46].

1.2 Fundamental tradeoffs in model selection

Which model should ultimately be used for a given problem is a very important yet challenging question. Indeed, the complexity of the choice models presented above is motivated by the need for greater predictive power in order to, for instance, break the IIA property. However, how does this richness affects the tractability of these models? Can we solve any decision problems using these models? This is especially important in revenue management as the goal is often to use these models to formulate some mathematical program which one ultimately would like to solve. Typically, simple models, from the predictive standpoint, lead to easy problems, from the tractability standpoint. On the other hand, rich models lead to hard problems.There is no free lunch: a more complex choice model can capture a richer substitution behavior but leads to increased complexity of the optimization problem. We explore and quantify these tradeoffs in the context of the assortment optimization problem, a core revenue management problem, which we introduce in the next section.

Many other dimensions are important in practice. We do not study them in this dissertation but would like to emphasize that the model selection problem involves carefully balancing all these tradeoffs. For instance, of the utmost importance is the estimation of these choice models from data. In this dissertation, we assume that the models are given and we try to assess the tractability of the associated assortment problem. However, estimating the parameters of the model from data is equally important. Moreover, this task is highly non trivial as in most settings, we are trying to infer customer preferences from very limited information, mainly their purchase data.

1.3 The assortment optimization problem

What subset (or assortment) of product to offer is a fundamental decision problem that commonly arises in several application contexts. A concrete setting is that of a retailer who carries a large universe of products but can offer only a subset of the products in each store, online or offline (see [44], [27]). The objective of the retailer is typically to choose the offer set that maximizes the expected revenue/profit[1] earned from each arriving customer, under stochastic demand. Another example is display-based online advertising where a publisher has to select a set of ads to display to

[1]Note that conversion-rate maximization can be obtained as a special case of revenue/profit maximization by setting the revenue/profit of all the products to be equal.

users. In this context, due to competing ads, the click rates for an individual ad depends on the overall subset of ads to be displayed.

We assume that we have a universe $\mathcal{N} = \{1, \ldots, n\}$ consisting of n products. Moreover, there is always an outside option modeling the fact that a customer could decide not to purchase anything. We denote it by 0. Each product i has an exogenous price p_i. Under this notation, the expected revenue $R(S)$ of the assortment $S \subseteq \mathcal{N}$ can be written as

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

For a given choice model, the associated assortment optimization problem consisting of maximizing the expected revenue can therefore be formulated as

$$\max_{S \subseteq \mathcal{N}} R(S). \tag{Assort}$$

Note that this is a combinatorial problem for which trying all 2^n possible assortment is not an scalable solution. We also consider variants of Assort where we add constraints on the assortment with the aim of capturing more realistic situations. There will be a particular emphasis on the capacity constrained version of the assortment problem. In that context, every product i is associated with a weight w_i, and the decision maker is restricted to selecting an assortment whose total weighs is at most a given bound W. This is also sometimes referred to as a knapsack constraint. We can formulate the capacity constrained assortment optimization problem as

$$\max_{S \subseteq \mathcal{N}} \left\{ R(S) : \sum_{i \in S} w_i \leq W \right\}. \tag{Capa}$$

For the special case of uniform weights (i.e. $w_i = 1$ for all i), the capacity constraint reduces to a constraint on the number of products in the assortment. We refer to this setting as the cardinality constrained assortment optimization problem:

$$\max_{S \subseteq \mathcal{N}} \left\{ R(S) : |S| \leq k \right\}. \tag{Card}$$

Both of these constraints on assortments arise naturally, allowing one to model practical scenarios such as shelf space constraint or budget limitations. We will also consider the case of totally-unimodular constraints. Let $x^S \in \{0,1\}^{|\mathcal{N}|}$ denote the incidence vector for any assortment $S \subseteq \mathcal{N}$ where $x_i^S = 1$ if $i \in S$ and $x_i^S = 0$ otherwise. The assortment optimization problem subject to a totally-unimodular constraint can be formulated as follows:

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

Here, A is a totally-unimodular matrix, and b is an integer vector. Note that Card is a special case of TU. These capture a wide range of practical constraints such as precedence, display locations, and quality consistent pricing constraints [23]. Finally, we will study at a robust version of the assortment optimization problem (Rob). In this variant, we capture the presence of uncertainty in the model parameters which can come, for instance, from their estimation from data. A common approach in that case it to resort to robust optimization, i.e. finding the assortment which maximizes the worst-case revenue under the uncertainty.

1.4 Summary of contributions of Chapters 2, 3 and 4

We summarize the main contributions of Chapters 2, 3 and 4. By collecting these results together, we can better contrast and compare them. Each of the following chapter will focus on a single model and will be self contained.

For the RUM models and the MC model (Chapters 2 and 3), the results presented in this thesis have the same flavor and are two-fold. On the one hand side, we design efficient algorithms with provable guarantees to address different variants of assortment problems. On the other hand, we complement these algorithms with hardness results which helps understanding what is the best possible approximation for a given problem. All our results are tight: the performance of our proposed algorithms matches the best possible lower bound prescribed by the hardness result. Together these results therefore allow us to better understand the limitations and tradeoffs inherent to different models. On the technical side, both these chapters introduce algorithmic frameworks which give unified approaches to various problems.

In Chapter 4, the challenges are slightly different. Indeed, unlike previous chapters, under a mixture of Mallows model computing the choice probabilities is already a non-trivial task because of the exponential support of the distribution. The main message of Chapter 4 is that despite this exponential support the Mallows-smoothed model choice probabilities can be computed efficiently. This in turns leads to efficient algorithms to solve assortment optimization problems.

Notations. To ease the reading and avoid repeating long expressions such as "the cardinality constrained assortment optimization problem under the MNL model", we will use the notation $\mathsf{Model} - \mathsf{Problem}$ to denote a particular problem under a given choice model. For instance, MNL-Card will refer to the cardinality constrained assortment problem under the MNL choice model. Tables 1.2 and 1.3 list all the choice models and problems abbreviations.

Choice model	**Abbreviation**
Multinomial logit	MNL
Nested logit	NL
d-level nested logit	dNL
Mixtures of MNL	mMNL
Markov chain	MC

Table 1.2: Abbreviations for different choice models

Assortment optimization problem	Abbreviation
Unconstrained	Assort
Cardinality constraint	Card
Capacity constraint	Capa
Totally-unimodular constraints	TU
Robust assortment optimization	Rob

Table 1.3: Abbreviations for various assortment problems

1.4.1 Random utility models

The popularity of the MNL comes from its tractability. In particular, MNL-Assort is tractable (see [71] for instance): the optimal assortment can be found in polynomial time. The structure of the optimal assortment is well understood: for MNL-Assort, the optimal assortment consists of the top k most expensive products for some k. There are many proofs of this beautiful result and we provide yet another one in Appendix B.5. [23] give an exact algorithm for MNL-Card, and more generally, for MNL-TU. [67] characterize the optimal assortment for MNL-Rob.

For more general RUM models, [24] give an exact algorithm for NL-Assort. [34] propose an exact algorithm for NL-Card , when the cardinality constraint affects each nest separately, and a constant factor approximation for NL-Capa under the same assumption. [31] present an exact algorithm when the cardinality constraint is across different nests. Under a mixture of MNL model, mMNL-Assort becomes NP-hard, even under a mixture of two MNL [66]. [64] devise a polynomial-time approximation scheme (PTAS) for mMNL-Card.

Contributions. As previously discussed, MNL-Assort and MNL-Card are tractable. However, we show that MNL-Capa is NP-hard. In light of this hardness result, we present a fully polynomial time approximation scheme (FPTAS) for MNL-Capa. In other words, for any $\epsilon > 0$, our algorithm computes a $(1-\epsilon)$-approximation of the optimal assortment in time polynomial in the input size and $1/\epsilon$. This is the best possible approximation for a NP-hard problem. Therefore, our algorithm gives the best possible approximation for MNL-Capa. Our algorithmic approach is very flexible and also gives near-optimal algorithms for NL-Capa, dNL-Capa under some mild assumptions.

When the number of mixtures is constant, we can also give a near-optimal algorithm for mMNL-Capa. [65] give a PTAS for a more general capacitated sum of ratio optimization problem based on a linear programming formulation. [57] give an FPTAS for the same problem. However, they use a black-box construction of an approximate Pareto-optimal frontier introduced by [60]. We would like to note that the running time of our algorithm is polynomial in the input size and $1/\epsilon$, but is exponential in K (number of mixtures in the mixture of MNL model). Therefore, we obtain an FPTAS only when the model is a mixture of a constant number of MNL models. To complement this result, we show that mMNL-Assort is hard to approximate within any reasonable factor when the number of mixtures is not constant. More specifically, there is no polynomial time algorithm (polynomial in number of items and mixtures: n, K and the input size) with an approximation factor better than $O(1/K^{1-\delta})$ for any constant $\delta > 0$ for mMNL-Assort unless $NP \subseteq BPP$. This implies that if we require a near-optimal algorithm for the assortment optimization over the mixture of MNL model, a super-polynomial dependence on the number of mixtures is necessary.

1.4.2 Markov chain model

[10] show that MC-Assort is polynomial time solvable. [76] also consider the Markov chain model in the context of airline revenue management, and present a simulation study. In a recent paper, [30] study the network revenue management problem under the Markov chain model and give a linear programming based algorithm.

Contributions. We show that MC-Card is NP-hard to approximate within a factor better than some given constant, even when all items have uniform prices. It is interesting to note that, while MC-Assort can be solved optimally in polynomial time, MC-Card is APX-hard. In contrast, in both the MNL and NL models, the unconstrained assortment optimization and the cardinality constrained assortment problems have the same complexity. We also consider the case of totally-unimodular (TU) constraints on the assortment. We show that MC-TU is hard to approximate within a factor of $O(n^{1/2-\epsilon})$ for any fixed $\epsilon > 0$, where n is the number of items. This result drastically contrasts with [23] where the authors prove that MNL-TU can be solved in polynomial time.

On the positive side, we develop a new algorithmic technique that gives, through a unified approach, a new alternative strongly polynomial algorithm for MC-Assort, a constant factor approximation for both MC-Card and MC-Capa as well as an exact algorithm for MC-Rob. Moreover, we consider a special case of MC model where the underlying Markov chain has constant rank. Under this additional assumption, we can leverage the tools from Chapter 2 and design a near optimal algorithm for MC-Capa.

1.4.3 Distribution over rankings

The intractability of the problem comes in two folds. First of all, specifying a general distribution over permutations may be expensive, as we may have to explicitly list exponentially many values along with their probabilities. Secondly, even for a general distribution over a small number of preference lists, [3] recently prove that it is NP-hard to compute a subset of products whose expected revenue is within factor better than $O(n^{1-\epsilon})$[2], for any accuracy level $\epsilon > 0$. This hardness of approximation result

[2] The reduction is from the independent set problem to an assortment optimization problem under a distribution over only n rankings.

discourages the hope of coming up with any reasonable approximation heuristic with a provably good approximation guarantee in the worst case. Nonetheless, with certain additional structural assumptions, certain special subclasses of such models can be shown to be tractable [3], [35], [36].

Contributions. We address the two key computational challenges that arise when using a mixture of Mallows model: (a) efficiently computing the choice probabilities and hence, the expected revenue/profit, for a given offer set S and (b) finding a near-optimal assortment. We also present a compact mixed integer program (MIP) and present a variable bound strengthening technique that leads to a practical approach for the constrained assortment optimization problem under a general mixture of Mallows model.

We present two efficient procedures to compute the choice probabilities $\pi(i, S)$ exactly under a general mixture of Mallows model. Because the mixture of Mallows distribution has an exponential support size, computing the choice probabilities for a fixed offer set S requires marginalizing the distribution by summing it over an exponential number of rankings, and therefore, is a non-trivial computational task. In fact, computing the probability of a general partial order under the Mallows distribution is known to be a #P hard problem [49, 13]. The only other known class of partial orders whose probabilities can be computed efficiently is the class of partitioned preferences [46]; while this class includes top-k/bottom-k ranked lists, it does not include other popular partial orders such as pairwise comparisons.

We present a polynomial time approximation scheme (PTAS) for a large class of constrained assortment optimization for the mixture of Mallows model including cardinality constraints, knapsack constraints, and matroid constraints. Our PTAS holds under the assumption that the no-purchase option is ranked last in the modal rankings for all Mallows segments in the mixture; such assumptions are necessary because of hardness of approximation for Assort under a sparse rank-based model mentioned above. Under the above assumption and for any $\epsilon > 0$, our algorithm computes an assortment with expected revenue at least $(1 - \epsilon)$ times the optimal in running time that is polynomial in n and K but depends exponentially on $1/\epsilon$.

1.4.4 Summary

We summarize some of the main results of the following chapters in Table 1.4 to help the reader better navigate through this thesis but also to help compare and contrast the results. No single model dominates the others on all accounts. Rather, we try to understand the price one has to pay, in terms of tractability, for increased predictive power. The hope is that this grid can guide practitioners in the selection of choice model depending on their application. For instance, if time is a constraint and the assortment optimization problem needs to be solved in split seconds (such as an online application for instance), then having a simpler but more tractable model may be interesting. However, if the assortment problem needs to be solve every other month, then a richer model would be the way to go.

Choice model	Assortment optimization problem		
	Unconstrained (Assort)	Cardinality constrained (Card)	Capacity constrained (Capa)
Multinomial logit (MNL)	Polynomially solvable by offering nested assortment ([71] , Appendix B.5)	Polynomially solvable using LP ([23], Theorem 2.1)	• Near-optimal algorithm (FPTAS) (Theorem 2.3) • NP-hardness (Theorem 2.2)
Nested logit (NL)	Polynomially solvable [24]	Polynomially solvable [31]	Near-optimal algorithm (FPTAS) (Theorem 2.4)
d-level nested logit (dNL)	Polynomially solvable [48]	Near-optimal algorithm (FPTAS) (Theorem 2.5)	
Mixtures of MNL (mMNL)	• Assort is NP-hard for constant number of mixture [66] • Assort NP-hard to approximate within $\Omega(1/K^{1-\delta})$ for arbitrary number of mixtures (Theorem 2.7) • Near-optimal algorithm (FPTAS) (Theorem 2.6)		
Markov chain (MC)	Polynomially solvable ([10], Theorem 3.4)	• 2-approximation (Theorem 3.5) • APX-hard (Theorem 3.1)	3-approximation (Theorem 3.6)
Distribution over rankings	• Mixed Integer Program formulation [9] • Assort is NP-hard to approximate within $\Omega(1/K^{1-\delta})$ [3]		
Mixture of Mallows	Mixed Integer Program formulation (Theorem 4.6)		

Table 1.4: Summary of contributions

Chapter 2

Near optimal algorithms for capacity constrained assortment under random utility models

In this chapter, we examine the capacity constrained assortment optimization problem (Capa) under various random utility models. We first show, in Section 2.1, that MNL-Capa is NP-hard. In light of this hardness result, we present a fully polynomial time approximation scheme (FPTAS) for the problem. In other words, for any $\epsilon > 0$, our algorithm computes a $(1 - \epsilon)$-approximation of the optimal assortment in time polynomial in the input size and $1/\epsilon$. This is the best possible approximation for a NP-hard problem. Therefore, our algorithm gives the best possible approximation for MNL-Capa. Our framework is flexible and can be extended to more general random utility models. In particular, we also derive FPTAS for NL-Capa (Section 2.2) and dNL-Capa (Section 2.3).

For the mixture of MNL model, we also obtain an FPTAS for mMNL-Capa (Section 2.4). However, the running time of our algorithm is exponential in the number of mixtures. Therefore, we obtain an FPTAS only when the model is a mixture of a constant number of MNL models. We further show that this super-polynomial dependance is necessary. In particular, even without any constraint, we show that mMNL-Assort is hard to approximate within any reasonable factor when the number of mixtures is not constant. More specifically, there is no polynomial time algorithm with an approximation factor better than $O(1/K^{1-\delta})$, where K is the number of mixtures, for any constant $\delta > 0$ for mMNL-Assort unless NP $\subseteq$ BPP.

2.1 Multinomial logit model

In this section, we examine the assortment optimization problem under the MNL model. The MNL model is given by $(n+1)$ parameters $u_0, \ldots, u_n$ which represent the preference weights of each product as well as the preference weight of the no purchase option. For any $S \subseteq [n]$, the choice probability of product j is given by

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

Each product $i \in [n]$ is also assigned a price p_i and a weight w_i. We denote by W the total available capacity. The capacity constrained assortment optimization can be formulated as follows.

$$\max_{S \subseteq [n]} \left\{ \sum_{j \in S} p_j \cdot \frac{u_j}{u_0 + \sum_{j \in S} u_i} \;\middle|\; \sum_{j \in S} w_j \leq W \right\}. \tag{MNL-Capa}$$

We would like to note that both MNL-Assort and MNL-Card are tractable under the MNL model (see, [71] and [23] respectively). We begin by giving an alternative proof for the LP based algorithm proposed in [23] for MNL-Card.

2.1.1 Cardinality Constraint: LP based Algorithm

As a warmup, we first consider MNL-Card, where there is an upper bound on the number of products in the assortment. We present an LP based optimal algorithm for this case. Our proof is different than [23] and is based on the properties of an optimal basic solution. In particular, we prove the following theorem.

Theorem 2.1. *MNL-Card is equivalent to the following linear program*

$$z_{LP} = \max \left\{ \sum_{j=1}^{n} p_j q_j \;\middle|\; u_0 q_0 + \sum_{j=1}^{n} q_j = 1 \;,\; \sum_{j=1}^{n} \frac{q_j}{u_j} \leq k q_0 \;,\; 0 \leq q_j \leq u_j q_0 \right\}, \tag{2.1}$$

where k is the upper bound on the number of items in the assortment. Furthermore, if $\boldsymbol{q}^*$ is an optimal solution, then $S^* = \{j \,|\, q_j^* = u_j q_0^*\}$ is an optimal assortment.

Proof. We first show that the above LP is a relaxation of MNL-Card. For any feasible solution $S \subseteq [n]$ for MNL-Card, we have the following feasible solution to the LP

$$q_0 = \frac{1}{u_0 + \sum_{i \in S} u_i} \quad \text{and} \quad q_j = \begin{cases} \dfrac{u_j}{u_0 + \sum_{i \in S} u_i} & \text{if } j \in S \\ 0 & \text{otherwise} \end{cases} \quad \forall j \geq 1.$$

Moreover, the two solutions give the same objective value which implies that $z_{LP} \geq z^*$.

We now show that any basic solution $\boldsymbol{q}^*$ to (2.1) satisfies $q_j^* \in \{0, u_j q_0^*\}$ for all $j = 1, \ldots, n$. We have $n + 1$ variables in (2.1) and only one equality constraint. Therefore, in a basic optimal solution, at least n inequalities are tight among

$$\sum_{j=1}^{n} \frac{q_j}{u_j} \leq k p_0 \quad \text{and} \quad 0 \leq q_j \leq u_j q_0 \quad \forall j \geq 1.$$

Consequently, $q_j \in \{0, u_j q_0\}$ for at least $(n-1)$ variables. Suppose exactly $(n-1)$ variables satisfy $q_j^* \in \{0, u_j q_0^*\}$ and one of the variable, say q_1^*, satisfies $0 < q_1^* < u_1 q_0^*$. Therefore, the inequality $\sum\limits_{j=1}^{n} \frac{q_j}{u_j} \leq k q_0$ must be tight and

$$k q_0^* = \sum_{j=1}^{n} \frac{q_j^*}{u_j} = \frac{q_1^*}{u_1} + \sum_{j=2}^{n} \frac{q_j^*}{u_j} = \rho q_0 + k' q_0$$

where k' is an integer and $0 < \rho < 1$. This yields a contradiction. Therefore, any basic solution leads to an integral solution of the original problem which means that $z_{LP} \leq z^*$. □

2.1.2 Hardness under a general capacity constraint

We now show that MNL-Capa, is NP-hard. We prove this by a reduction from the knapsack problem.

Theorem 2.2. *MNL-Capa is NP-hard.*

Proof. We give a reduction from the knapsack. In an instance of the knapsack problem on n items, we are given weights $c_1, \ldots, c_n$ and profits $r_1, \ldots, r_n$ and a knapsack capacity C. The goal is to find the most profitable assortment of items.

Consider the following instance for MNL-Capa:

$$u_0 = 1, \quad W = C \quad \text{and} \quad \forall j \geq 1, \quad u_j = r_j, \quad p_j = 1, \quad w_j = c_j.$$

For this instance, the problem becomes

$$\max_{\boldsymbol{x} \in \{0,1\}^n} \left\{ \frac{\sum_{i=1}^{n} r_i x_i}{1 + \sum_{i=1}^{n} r_i x_i} \;\middle|\; \sum_{i=1}^{n} c_i x_i \leq C \right\}.$$

Note that the function $f(x) = \dfrac{x}{1+x}$ is increasing in x. Therefore, maximizing $f(\boldsymbol{r}^{\boldsymbol{T}}\boldsymbol{x})$ is equivalent to maximizing $\boldsymbol{r}^{\boldsymbol{T}}\boldsymbol{x}$, hence the reduction to the knapsack problem. □

2.1.3 FPTAS for MNL-Capa

We present an FPTAS for MNL-Capa. Note that in view of Theorem 2.2, this is best possible for MNL-Capa. Our algorithm utilizes the rational structure of the objective function and is based on solving a polynomial number of dynamic programs. Since the objective function is rational, we guess the value of the numerator ($\sum_{j \in S^*} p_j u_j$) and denominator ($\sum_{j \in S^*} u_j$), for an optimal solution, S^* within a factor of $(1+\epsilon)$. We then try to find a feasible assortment (satisfying the capacity constraint) with the numerator and denominator values approximately equal to the guesses using a dynamic program. We would like to note that these dynamic programs are similar to multi-dimensional knapsack problems for which there is no FPTAS [32]. However, in our problem, we are allowed to violate the constraints which allows us to obtain an FPTAS.

Let p (resp. P) be the minimum (resp. maximum) revenue and u (resp. U) be the minimum (resp. maximum) MNL parameter. We can assume wlog. that $p, u > 0$; otherwise, we can clearly remove the corresponding product from our collection and continue. For any given $\epsilon > 0$, we use the following set of guesses for the numerator and denominator.

$$\Gamma_\epsilon = \{ru(1+\epsilon)^\ell, \ell = 0, \ldots, L_1\} \;\text{ and }\; \Delta_\epsilon = \{u(1+\epsilon)^\ell, \ell = 0, \ldots, L_2\}, \tag{2.2}$$

where $L_1 = O\left(\log\left(nPU/(ru)\right)/\epsilon\right)$ and $L_2 = O\left(\log\left((n+1)U/u\right)/\epsilon\right)$. The number of guesses is polynomial in the input size and $1/\epsilon$. For a given guess $h \in \Gamma_\epsilon$, $g \in \Delta_\epsilon$, we try to find a feasible assortment S with

$$\sum_{j \in S} p_j u_j \geq h \quad \text{and} \quad \sum_{j \in S} u_j \leq g, \tag{2.3}$$

using a dynamic program. In particular, we consider the following discretized values of $p_j u_j$ and u_j in multiples of $\epsilon h/n$ and $\epsilon g/(n+1)$ respectively,

$$\tilde{p}_j = \left\lfloor \frac{p_j u_j}{\epsilon h/n} \right\rfloor \quad \text{and} \quad \tilde{u}_j = \left\lceil \frac{u_j}{\epsilon g/(n+1)} \right\rceil, \ \forall j. \tag{2.4}$$

Note that we round down the numerator and round up the denominator to maintain the right approximation. For a given set of guesses, note that the problem can be reduced to a multi-dimensional knapsack for which there exists a PTAS, see for example [32]. The main difference is that we do not have hard constraints like in the knapsack. This allows us to round the coefficients while still maintaining feasibility. Also, note that we discretize the product $p_j u_j$ for all j instead of considering separate discretizations for r_j and u_j.

We can now present our dynamic program. Let $I = \lfloor n/\epsilon \rfloor - n$ and $J = \lceil (n+1)/\epsilon \rceil + (n+1)$. For each $(i, j, \ell) \in [I] \times [J] \times [n]$, let $F(i, j, \ell)$ be the minimum weight of any subset $S \subseteq \{1, \ldots, \ell\}$ such that

$$\sum_{s \in S} \tilde{p}_s \geq i \quad \text{and} \quad \sum_{s \in S_+} \tilde{u}_s \leq j.$$

We compute $F(i, j, \ell)$ for $(i, j, \ell) \in [I] \times [J] \times [n]$ using the following recursion.

$$F(i,j,1) = \begin{cases} w_1 & \text{if } 0 \leq i \leq \tilde{p}_1 \text{ and } j \geq \tilde{u}_0 + \tilde{u}_1 \\ 0 & \text{if } i \leq 0 \text{ and } j \geq \tilde{u}_0 \\ \infty & \text{otherwise} \end{cases} \tag{2.5}$$

$$F(i, j, \ell+1) = \min\{F(i, j, \ell), w_{\ell+1} + F(i - \tilde{p}_{\ell+1}, j - \tilde{u}_{\ell+1}, \ell)\}$$

Note that this dynamic program is similar to the one for the knapsack problem. Using this dynamic program, we construct a set of candidate assortments $S_{h,g}$ for all guesses

$(h,g) \in \Gamma_\epsilon \times \Delta_\epsilon$. Algorithm 1 details the procedure to construct the set of candidate assortments.

Algorithm 1 Construct Candidate Assortments

1: For $(h,g) \in \Gamma_\epsilon \times \Delta_\epsilon$,
   (a) Compute discretization of coefficients $\tilde{p}_i$ and $\tilde{u}_i$ using (2.4).
   (b) Compute $F(i,j,\ell)$ for all $(i,j,\ell) \in [I] \times [J] \times [n]$ using (2.5).
   (c) Let $S_{h,g}$ be the subset corresponding to $F(I,J,n)$.
2: Return $\mathcal{A} = \cup_{(h,g)\in\Gamma_\epsilon\times\Delta_\epsilon} S_{h,g}$.

Let us show that Algorithm 1 correctly finds a subset satisfying (2.3). In particular, we have the following lemma.

Lemma 2.1. *Let $\mathcal{A}$ be the set of candidate assortment returned by Algorithm 1. For any guess $(h,g) \in \Gamma_\epsilon \times \Delta_\epsilon$, if there exists S such that $W(S) \leq W$ and (2.3) is satisfied, then $W(S_{h,g}) \leq W$. Moreover, $S_{h,g}$ satisfies (2.3) approximately, i.e.*

$$\sum_{j\in S_{h,g}} p_j u_j \geq h(1-2\epsilon) \quad \textit{and} \quad \sum_{j\in S_{h,g}} u_j \leq g(1+2\epsilon).$$

Proof. Consider S satisfying (2.3) for given guesses h,g. Scaling the two inequalities yield

$$\sum_{j\in S} \frac{p_j u_j}{\epsilon h/n} \geq \frac{h}{\epsilon h/n} \quad \text{and} \quad \sum_{j\in S} \frac{u_j}{\epsilon g/(n+1)} \leq \frac{g}{\epsilon g/(n+1)}.$$

Rounding down and up the previous inequalities gives

$$\sum_{j\in S} \tilde{p}_j \geq \lfloor n/\epsilon \rfloor - n = I \quad \text{and} \quad \sum_{j\in S} \tilde{u}_j \leq \left\lceil \frac{(n+1)}{\epsilon} \right\rceil + (n+1) = J,$$

which implies that $F(I,J,n) \leq W$. Moreover, let $S_{h,g}$ be the corresponding subset. We have

$$\sum_{j\in S_{h,g}} p_j u_j \geq I \frac{\epsilon h}{n} \geq h(1-2\epsilon) \quad \text{and} \quad \sum_{j\in S_{h,g}} u_j \leq J \frac{\epsilon g}{n+1} \leq g(1+2\epsilon).$$

□

Algorithm 2 FPTAS for MNL-Capa

1: Construct a set of candidate assortment $\mathcal{A}$ using Algorithm 1.
2: Return the best feasible solution to MNL-Capa from $\mathcal{A}$.

Now that we have constructed a set of candidate assortment, the second part of the algorithm consists of returning the best possible feasible assortment. Algorithm 2 presents a complete description of the algorithm.

Theorem 2.3. *Algorithm 2 returns an $(1-\epsilon)$-optimal solution to MNL-Capa. Moreover, the running time is $O\left(\log(nPU)\log(nU)n^3/\epsilon^4\right)$.*

Proof. Let S^* be the optimal solution to MNL-Capa and $(\hat{\ell}_1, \hat{\ell}_2)$ such that

$$pu\,(1+\epsilon)^{\hat{\ell}_1} \leq \sum_{i\in S^*} r_i u_i \leq pu\,(1+\epsilon)^{\hat{\ell}_1+1} \quad \text{and} \quad u\,(1+\epsilon)^{\hat{\ell}_2} \leq \sum_{i\in S^*_+} u_i \leq u\,(1+\epsilon)^{\hat{\ell}_2+1}.$$

From Lemma 2.1, we know that for $(h,g) = \left(ru\,(1+\epsilon)^{\hat{\ell}_1}, u\,(1+\epsilon)^{\hat{\ell}_2}\right)$, $\mathcal{A}$ contains an assortment $\tilde{S}$ such that

$$\sum_{i\in\tilde{S}} p_i u_i \geq ru\,(1+\epsilon)^{\hat{\ell}_1}\,(1-2\epsilon) \;\; \text{and} \;\; \sum_{i\in\tilde{S}_+} u_i \leq u\,(1+\epsilon)^{\hat{\ell}_2}\,(1+2\epsilon).$$

Consequently,

$$f(\tilde{S}) = \frac{\sum_{i\in\tilde{S}} p_i u_i}{\sum_{i\in\tilde{S}_+} u_i} \geq \frac{1-2\epsilon}{1+2\epsilon} f(S^*) \geq (1-4\epsilon) f(S^*).$$

Running Time. Note that in Algorithm 1, we try $L_1 \cdot L_2$ guesses for the numerator and denominator values of the optimal solution. For each guess, we formulate a dynamic program with $O\left(n^3/\epsilon^2\right)$ states. Therefore, the running time of Algorithm 2 is $O\left(L_1 L_2 n^3/\epsilon^2\right) = O\left(\log(nPU)\log(nU)n^3/\epsilon^4\right)$ which is polynomial in input size and $1/\epsilon$. Note that $\log P$ and $\log U$ are both polynomial in the input size. □

2.2 Nested logit model

We now consider the capacitated assortment optimization problem for the nested logit (NL) model. In a NL model, the set of products is partitioned into nests (or subsets) and the choice probability for any product j is decomposed in the probability of selecting the nest containing j and the probability of selecting j in that nest. Suppose there are K nests $N_1, \ldots, N_K$ and each nest N_k contains n products with price $p_{i,k}$ and utility parameter $u_{i,k}$. As in the MNL model, we assign a utility of U_0 to the no-purchase alternative. We assume that there is no no-purchase option within each nest, i.e. $u_{0,k} = 0$ for all $k \in [K]$. Each nest N_k has a *dissimilarity parameter*, $\gamma_k \in [0,1]$ that models the influence of nest k over others. Note that these two assumptions are necessary to make NL-Assort tractable [24]. For a set of assortments $(S_1, \ldots, S_K)$, the probability that nest k is selected is given by

$$Q_k(S_1, \ldots, S_K) = \frac{U_k(S_k)^{\gamma_k}}{U_0 + \sum_{j=1}^{K} U_j(S_j)^{\gamma_j}},$$

where $U_k(S_k) = \sum_{i \in S_k} u_{i,k}$ for all $k \in [K]$. Let R_k denote the expected revenue of nest k conditional on nest k being selected. Then

$$R_k(S_k) = \sum_{i \in S_k} p_{i,k} \frac{u_{i,k}}{\sum_{j \in S_k} u_{j,k}} = \frac{\sum_{i \in S_k} p_{i,k} u_{i,k}}{U_k(S_k)}.$$

Additionally, each product is assigned a weight $w_{i,k}$. Let W_k be the available capacity for nest k for $k \in [K]$. We also assume that there is total available capacity W. We introduce the following capacitated assortment optimization for the NL model.

$$\begin{aligned} \max_{(S_1,\ldots,S_K) \subseteq [n]^K} \quad & \sum_{k=1}^{K} Q_k(S_1, \ldots, S_K) R_k(S_k) \\ & W(S_k) \leq W_k, \ \forall k \in [K] \qquad \text{(NL-Capa)} \\ & \sum_{k=1}^{K} W(S_k) \leq W, \end{aligned}$$

where $W(S_k) = \sum_{i \in S_k} w_{i,k}$ for all $k \in [K]$. Note that [34] give a 2-approximation when $W = \infty$ and [31] give a 4-approximation when $W_k = \infty$ for all $k \in [K]$. Here, we allow both a constraint on each nest as well as a constraint across all nests.

Before we describe the algorithm, we first reformulate the problem. The epigraph formulation of NL-Capa is

$$\min z$$
$$z \geq \sum_{k=1}^{K} Q_k(S_1, \ldots, S_K) R_k(S_k),\ \forall (S_1, \ldots, S_K) \subseteq \mathcal{S},$$

where

$$\mathcal{S} = \left\{ (S_1, \ldots, S_K) \subseteq [n]^k, W(S_k) \leq W_k, \forall k \in [K], \sum_{k=1}^{K} W(S_k) \leq W \right\}.$$

We can rewrite the previous problem as

$$\min z$$
$$U_0 z \geq \sum_{k=1}^{K} U_k(S_k)^{\gamma_k} (R_k(S_k) - z),\ \forall (S_1, \ldots, S_K) \subseteq \mathcal{S}.$$

From that formulation, we can see that the optimal revenue z^* is the unique fixed point to the following equation.

$$U_0 z = \max_{(S_1,\ldots,S_K) \subseteq \mathcal{S}} \left\{ \sum_{k=1}^{K} U_k(S_k)^{\gamma_k} (R_k(S_k) - z) \right\}.$$

Note that this reformulation was first used in [34]. We present it here for completeness. The algorithm consists of performing a binary search on z and for each fixed value of z, solving this auxiliary problem

$$\max_{(S_1,\ldots,S_K) \subseteq \mathcal{S}} \left\{ \sum_{k=1}^{K} U_k(S_k)^{\gamma_k} (R_k(S_k) - z) \right\}. \tag{2.6}$$

Since our goal is to design a near-optimal algorithm, we will aim at getting a $(1-\epsilon)$-optimal solution to (2.6). To do so, we introduce the following variant auxiliary problem.

$$\max_{\substack{(S_1,\ldots,S_K) \\ \sum_{k=1}^{K} W(S_k) \leq W \\ S_k \in \mathcal{A}_k,\ \forall k \in [K]}} \left\{ \sum_{k=1}^{K} U_k(S_k)^{\gamma_k} (R_k(S_k) - z) \right\}. \tag{Root}$$

where $\mathcal{A}_k$ is a set of candidate assortments for nest k, for all $k \in [K]$. Moreover, for each nest k, we introduce the following subproblem, parametrized by b

$$\max_{\substack{S_k \subseteq N_k \\ W(S_k) \leq \min(W_k, b)}} \{U(S_k)^{\gamma_k}(R(S_k) - z)\}. \tag{$\mathsf{Child_k}$}$$

Lemma 2.2. *Assume that the collection of candidate assortment $\mathcal{A}_k$ includes a $(1-\epsilon)$-approximate solution* ($\mathsf{Child_k}$) *for any $b \in \mathbb{R}_+$. Then, a $(1-\epsilon)$-approximate solution to* (Root) *also gives a $(1-\epsilon)$-approximate solution to* (2.6).

Proof. For a fixed z, let $(S_1^*, \ldots, S_K^*)$ be the optimal solution to (2.6) and let $b_k^* = W(S_k^*)$ for all $k \in [K]$. Note that (2.6) is therefore equivalent to the following decomposed problem.

$$\sum_{k=1}^{K} \max_{\substack{S_k \subseteq N_k \\ W(S_k) \leq b_k^*}} U_k(S_k)^{\gamma_k}(R_k(S_k) - z). \tag{2.7}$$

Therefore, if for $k \in [K]$, we let $\hat{S}_k \subseteq \mathcal{A}_k$ be the best candidate assortment for b_k^*, then $(\hat{S}_1, \ldots, \hat{S}_K)$ is a $(1-\epsilon)$-approximate solution to (2.6). The optimal solution to (Root) is therefore a $(1-\epsilon)$-approximate solution to (2.6). This concludes the proof. □

We can now give a high-level description of the FPTAS for $\mathsf{NL\text{-}Capa}$. It consists of a binary search on z. Then, for each fixed value of z, we perform the following steps.

- For all $k \in [K]$, construct a set of candidate assortments $\mathcal{A}_k$ for all $k \in [K]$ such that $\mathcal{A}_k$ includes a $(1-\epsilon)$-approximate solution to ($\mathsf{Child_k}$) for any $b \in \mathbb{R}_+$.
- Construct a $(1-\epsilon)$-approximate solution $(\hat{S}_1, \ldots, \hat{S}_K)$ to (Root).
- Adjust z according to the sign of $U_0 z - \sum_{k=1}^{K} U_k(\hat{S}_k)^{\gamma_k}(R_k(\hat{S}_k) - z)$.

We now give more details for each part of the algorithm.

2.2.1 Binary search and preprocessing

In order to perform a binary search on z, our guess on the optimal revenue z^*, we first provide upper and lower bounds on z. For each $k \in [K]$, let S_k^* be the optimal solution to $\mathsf{MNL\text{-}Capa}$, i.e. the constrained assortment that maximizes $R_k(S_k)$ for each single nest. Let $i^* = \arg\max\{R(S_k^*) : k \in [K]\}$. We have the following bounds on z^*,

$$z_{\min} = \frac{U_{i^*}(S_{i^*}^*)^{\gamma_{i^*}}}{U_0 + U_{i^*}(S_{i^*}^*)^{\gamma_{i^*}}} R(S_{i^*}) \leq z^* \leq R(S_{i^*}) = z_{\max} \tag{2.8}$$

Having both a lower and upper bound on the optimal z^*, we can perform a binary search on z. Moreover, this allows us to prune products with too little revenue within each nest. To do so, we first show that we can always remove nest with too little revenue from any assortment.

Lemma 2.3. *Let $(\hat{S}_1, \ldots, \hat{S}_K)$ be a $(1-\epsilon)$-approximate solution to* (Root)*. For all, $k \in [K]$ such that*

$$U_k(S_k)^{\gamma_k}(R_k(S_k) - z) \leq \epsilon U_0 z_{\min}/K, \tag{2.9}$$

replacing $\hat{S}_k$ by $\emptyset$ also give a $(1-\epsilon)$-approximate solution to (Root)*.*

Proof. Let $(S_1^*, \ldots, S_K^*)$ be the optimal solution to parent. For all $k \in [K]$, let $\hat{S}_k$ be a $(1-\epsilon)$-approximate solution to ($\mathsf{Child_k}$). We have

$$\sum_{k=1}^{K} U_k(\hat{S}_k)^{\gamma_k}(R_k(\hat{S}_k) - z) \geq (1-\epsilon) \sum_{k=1}^{K} U_k(S_k^*)^{\gamma_k}(R_k(S_k^*) - z).$$

Let $\hat{K}$ be the set of indices such that (2.9) holds. We have

$$\sum_{k \in \hat{K}} U_k(\hat{S}_k)^{\gamma_k}(R_k(\hat{S}_k) - z) \leq \epsilon U_0 z_{\min} \leq \epsilon \sum_{k=1}^{K} U_k(S_k^*)^{\gamma_k}(R_k(S_k^*) - z).$$

This in turn implies that replacing $\hat{S}_k$ by $\emptyset$ for all $k \in [K]$ yields

$$\sum_{k=1}^{K} U_k(\hat{S}_k)^{\gamma_k}(R_k(\hat{S}_k) - z) \geq (1-2\epsilon) \sum_{k=1}^{K} U_k(S_k^*)^{\gamma_k}(R_k(S_k^*) - z).$$

□

This implies the following corollary that allows us to prune products whose revenue is too small.

Corollary 2.1. *For a given value of z, we can remove products such that*

$$u_{i,k}(p_{i,k} - z) \leq \frac{\epsilon U_0 z_{\min}/K}{nU_k(N_k)^{\gamma_k}} = h_{\min,k}$$

and still approximate (Root) *within factor* $(1-\epsilon)$.

2.2.2 Constructing Candidate Assortments for $(\mathsf{Child_k})$.

In this section, we fix $k \in [K]$. Note that the objective function to $(\mathsf{Child_k})$ can be written as

$$U(S_k)^{\gamma_k}(R(S_k) - z) = \left(\sum_{i \in S_k} u_{i,k}\right)^{\gamma_k - 1} \left(\sum_{i \in S_k} u_{i,k}(r_{i,k} - z)\right).$$

We use Algorithm 1 to construct candidate assortments. Indeed, note that we need to guess the quantities $(\sum_{i \in S} u_i)$ and $(\sum_{i \in S} u_i(p_i - z))$. In order to use Algorithm 1, we need to specify the sets Γ_ϵ and Δ_ϵ that we use for the guesses. Note that by Corollary 2.1, we can assume that for all $i \in [n]$, $u_{i,k}(p_{i,k} - z) > h_{\min,k}$. Therefore, we can use the following set of guesses.

$$\Gamma_\epsilon = \{h_{\min,k}(1+\epsilon)^\ell, \ell = 0, \ldots, L_1\} \text{ and } \Delta_\epsilon = \{u(1+\epsilon)^\ell, \ell = 0, \ldots, L_2\}, \tag{2.10}$$

where $L_1 = O(\log(nPU/h_{\min,k})/\epsilon)$ and $L_2 = O(\log(nU/u)/\epsilon)$ and u, U and P and respectively the minimum utility, maximum utility and maximum revenue of an item in the nest k.

Lemma 2.4. *Let S_k^* be the optimal solution to* $(\mathsf{Child_k})$. *If $U_k(S_k^*)^{\gamma_k}(R_k(S_k^*) - z) > \epsilon U_0 z_{\min}/K$, then the set $\mathcal{A}$ returned by Algorithm 1 using the set of guesses* (2.10) *contains a $(1-\epsilon)$-optimal solution to* $(\mathsf{Child_k})$ *for any $b \in \mathbb{R}_+$. Moreover, both the size of $\mathcal{A}$ and the running time of Algorithm 1 are polynomial in the input size and $1/\epsilon$.*

Proof. Let S_k^* be the optimal solution to $(\mathsf{Child_k})$ for a given value b and $(\hat{\ell}_1, \hat{\ell}_2)$ such that

$$h_{\min,k}\,(1+\epsilon)^{\hat{\ell}_1} \leq \sum_{i \in S_k^*} u_{i,k}(p_{i,k} - z) \leq h_{\min,k}\,(1+\epsilon)^{\hat{\ell}_1+1}\,, \text{ and}$$

$$u\,(1+\epsilon)^{\hat{\ell}_2} \leq \sum_{i \in S^*} u_{i,k} \leq u\,(1+\epsilon)^{\hat{\ell}_2+1}\,.$$

From Lemma 2.1, we know that for $(h, g) = (h_{\min,k}\,(1+\epsilon)^{\hat{\ell}_1}\,, u\,(1+\epsilon)^{\hat{\ell}_2})$, $S_{h,g}$ is such that

$$\sum_{i \in S_{h,g}} u_{i,k}(p_{i,k} - z) \geq pu\,(1+\epsilon)^{\hat{\ell}_1}\,(1-2\epsilon) \;\text{ and }\; \sum_{i \in S_{h,g}} u_{i,k} \leq u\,(1+\epsilon)^{\hat{\ell}_2}\,(1+2\epsilon).$$

Consequently,

$$f(S_{h,g}) = \left(\sum_{i \in S_{h,g}} u_{i,k}\right)^{\gamma_k - 1} \left(\sum_{i \in S_{h,g}} u_{i,k}(p_{i,k} - z)\right) \geq \frac{1-2\epsilon}{(1+2\epsilon)^{1-\gamma_k}} f(S_k^*) \geq (1-4\epsilon) f(S_k^*).$$

Both the size of $\mathcal{A}$ and running time of Algorithm 1 being polynomial in the input size and $1/\epsilon$ follow from the proof of Theorem 2.3. □

2.2.3 FPTAS for (Root)

We show how to approximately maximize (Root) for a given value of z and given sets $\mathcal{A}_k$ for all $k \in [K]$ of candidate assortments for each nest. Note that we have candidate assortments for each nest and we are trying to stitch together an assortment $(S_1, \ldots, S_K)$. Also, note that candidate assortments satisfy individual nest constraints. We will now need to make sure that the assortment $(S_1, \ldots, S_K)$ satisfies the constraint across the different nests. Again, we use ideas similar to Algorithm 2 by guessing the value of the objective function. Consider the following set of guesses.

$$\Gamma = \{U_0 z_{\min}(1+\epsilon)^{\ell}, \ell = 0, \ldots, L\},$$

and $L = O(\log(z_{\max}/z_{\min})/\epsilon)$. For each guess $v \in \Gamma$, we use a dynamic program to find a feasible assortment $(S_1, \ldots, S_K)$ such that

$$\sum_{k=1}^{K} U_k(S_k)^{\gamma_k}(R_k(S_k) - z) \geq v.$$

For every candidate assortment $S_k \in \mathcal{A}_k$, we consider the following discretized values in multiples of $\epsilon z/K$,

$$\tilde{r}_{S_k} = \left\lfloor \frac{U_k(S_k)^{\gamma_k}(R_k(S_k) - z)}{\epsilon z/K} \right\rfloor. \tag{2.11}$$

Let $F(u, v)$ be the minimum weight of any subsets $(S_1, \ldots, S_p) \subseteq (N_1, \ldots, N_p)$ such that

$$\sum_{k=1}^{p} \tilde{r}_{S_k} \geq v.$$

Let $I = \lfloor K/(\epsilon z) \rfloor - K$. We can compute $F(u, v)$ for $(u, v) \in [K] \times [I]$ using the following recursion. Let $\Lambda = \{S_1 \in \mathcal{A}_1 : W(S_1) \leq W_1, r_{S_1} \geq v\}$.

$$F(1, v) = \begin{cases} \min\{W(S_1) : S_1 \in \Lambda\} & \text{if } v > 0 \text{ and } \Lambda \neq \emptyset \\ 0 & \text{if } v \leq 0 \\ \infty & \text{otherwise} \end{cases} \tag{2.12}$$

$$F(u+1, v) = \min\left\{ F(u, v), \min_{\substack{S_{u+1} \in \mathcal{A}_{u+1} \\ W(S_{u+1}) \leq W_{u+1}}} W(S_{u+1}) + F(u, v - r_{S_u}) \right\} \tag{2.13}$$

Algorithm 3 details the procedure.

Algorithm 3 FPTAS for (Root)

1: For $h \in \Gamma_\epsilon$,
  (a) For $k \in [K]$, let $\mathcal{A}_k$ be the set of candidate assortment returned by Algorithm 1.
  (b) For $k \in [K]$ and $S_k \in \mathcal{A}_k$, compute discretization of coefficients $\tilde{r}_{S_i}$ using (2.11).
  (c) Compute $F(u, v)$ for all $(u, v) \in [K] \times [I]$ using (2.12).
  (d) Let S_h be the subset corresponding to the state $F(K, I)$.
2: Return the best feasible solution to (Root) from $\cup_{h \in \Gamma_\epsilon} S_h$

Lemma 2.5. *Algorithm 3 returns a* $(1-\epsilon)$*-approximate solution to* (Root)*. Morever, the running time is polynomial in input size and* $1/\epsilon$*.*

The proof is similar to the proof of Lemma 2.4. Putting together the different results yields the following result.

Theorem 2.4. *There is an FPTAS for* NL-Capa *with running time polynomial in* n*,* K *and the input size when* $\gamma_k \in [0,1]$ *and* $u_{0,k} = 0$ *for all* $k \in [K]$*.*

2.3 d-level nested logit

We also extend our FPTAS to the setting where the choice model is given by a d-level nested logit (dNL) model. [48] show how to solve dNL-Assort in polynomial time. We adapt the technique used in the previous section to approximate dNL-Capa. We have n products indexed by $\{1, 2, \dots, n\}$ and the no purchase option denoted by 0. We additionally have a d-level tree denoted by (T, V) with vertices V and edges E. The tree has n leaf nodes at depth d, corresponding to the n products. We use Children(j) to denote the set of child nodes of node j in the tree and Parent(j) to denote the parent node of node j. Each node $v \in V$ has n_v children and is associated with a set of products, or leaf nodes, that are included in the subtree rooted at node j. Each assortment $S \subseteq [n]$ defines a collection of subsets $(S_v : v \in V)$ at each node of the tree. If v is a leaf node, then

$$S_v = \begin{cases} \{j\} & \text{if } j \in S \\ \emptyset & \text{otherwise} \end{cases}.$$

When v is not a leaf node, we define S_v recursively by setting $S_v = \bigcup_{w \in \mathsf{Children(v)}} S_w$. Each node is associated with a dissimilarity parameter $\gamma_v \in [0,1]$. We define the utility of each leaf node v as

$$S_v = \begin{cases} u_j & \text{if } j \in S \\ \emptyset & \text{otherwise} \end{cases},$$

and the utility of any non leaf node is defined by

$$U_v(S_v) = \left(\sum_{k \in \mathsf{Children(v)}} U_k(S_k) \right)^{\gamma_v}.$$

The revenue are defined similarly by recursion. For all non leaf node, we have

$$R_v(S_v) = \sum_{k \in \mathsf{Children(j)}} \frac{U_k(S_k) R(S_k)}{\sum_{\ell \in \mathsf{Children(j)}} U_\ell(S_\ell)}$$

Furthermore, each assortment S_v has a weight $W(S_v)$ equal to the sum of the weights of all the leave nodes includes in the subtree rooted at v. We assume that there is a capacity constraint W_v associated with each node $v \in V$. The assortment optimization problem under the d-level nested logit can be written as

$$\max_{W(S_v) \leq W_v, \forall v \in V} R_{\mathsf{root}}(S_{\mathsf{root}}). \qquad \text{(dNL-Capa)}$$

We use a similar approach where we construct candidate assortments at each node using a dynamic program . To construct the set of candidate assortments, we use the sets of candidate assortments of the children nodes.

Theorem 2.5. *There is an FPTAS for dNL-Capa with running time polynomial in n, d,and the input size.*

Moreover, note that this framework can be used to solve NL-Capa with additional constraints on the nests as long as they are representable in a tree structure.

Corollary 2.2. *There is a FPTAS for NL-Capa with additional capacity constraints when those constraints have a nested structure.*

We now present the algorithm for $\mathsf{dNL\text{-}Capa}$. As for the NL model, the problem can be formulated as a fixed point equation. More precisely, the optimal revenue z^* is the unique fixed point to the following equation.

$$U_0 z = \max_{\substack{(S_1,\ldots,S_{n_{\mathsf{Root}}})\subseteq \mathsf{Children(Root)} \\ W(S_v)\le W_v, \forall v\in V}} \left\{ \sum_{v\in \mathsf{Children(Root)}} U_v(S_v)^{\gamma_v}(R_v(S_v)-z) \right\}.$$

For a fixed z, we need to solve this problem

$$\max_{\substack{(S_1,\ldots,S_{n_{\mathsf{Root}}})\subseteq \mathsf{Children(Root)} \\ W(S_v)\le W_v, \forall v\in V}} \left\{ \sum_{v\in \mathsf{Children(Root)}} U_v(S_v)^{\gamma_v}(R_v(S_v)-z) \right\}. \tag{2.14}$$

Similarly, we introduce the following auxiliary problems.

$$\max_{\substack{(S_1,\ldots,S_{n_{\mathsf{Root}}})\subseteq \mathsf{Children(Root)} \\ \sum_{v\in \mathsf{Children(Root)}} W(S_v)\le W_{\mathsf{Root}} \\ S_v\in\mathcal{A}_v,\ \forall v\in[n_v]}} \left\{ \sum_{v\in \mathsf{Children(Root)}} U_v(S_v)^{\gamma_v}(R_v(S_v)-z) \right\}. \tag{d-Root}$$

where $\mathcal{A}_k$ is a set of candidate assortments for node v, for all $v \in V$. Moreover, for each node $v \in V$, we introduce the following subproblem, parametrized by b

$$\max_{\substack{(S_1,\ldots,S_{n_v})\subseteq \mathsf{Children(v)} \\ \sum_{w\in \mathsf{Children(v)}} W(S_w)\le \min(W_v,b) \\ S_w\in\mathcal{A}_w, \forall w\in \mathsf{Children(v)}}} \left(\sum_{w\in \mathsf{Children(v)}} U(S_w) \right)^{\gamma_v-1} \left(\sum_{w\in \mathsf{Children(v)}} U(S_w)(R(S_w)-z) \right). \tag{$\mathsf{Node_v}$}$$

Inductively using the the proof of Lemma 2.2, we have the following lemma which allows us to construct a near optimal solution starting from the lower levels of the trees and building up a solution.

Lemma 2.6. *Assume that the collection of candidate assortment $\mathcal{A}_v$ includes a $(1-\epsilon)$-approximate solution* ($\mathsf{Node_v}$) *for all $v \in V\backslash\{\mathsf{Root}\}$ and any $b \in \mathbb{R}_+$. Then, a $(1-\epsilon)$-approximate solution to* (d-Root) *also gives a $(1-\epsilon)$-approximate solution to* (2.14).

For a given z and node $v \in V$, we construct candidate assortments sequentially from the candidate assortment from $\mathsf{Children(v)}$. We only detail this step as the rest of the algorithm is similar to the algorithm for the NL model.

Constructing Candidate Assortment. For a fixed node $v \in V$, the objective function to ($\mathsf{Node_v}$) can be written as

$$\left(\sum_{w \in \mathsf{Children(v)}} U(S_w)\right)^{\gamma_v - 1} \left(\sum_{w \in \mathsf{Children(v)}} U(S_w)(R(S_w) - z)\right).$$

We use a dynamic program to construct a set of candidate assortment for node v based on candidate assortment of its children. The algorithm is similar in spirit to Algorithm 1. The only difference is that instead of items, we have candidate assortments. For each guesses (h, g), we discretize the revenues and utilities of the candidate assortments of the children node as follows. For all $p \in [n_v]$ and all $S \in \mathcal{A}_p$, we define

$$\tilde{r}_S = \left\lfloor \frac{U(S)(R(S) - z)}{\epsilon h / n_v} \right\rfloor \quad \text{and} \quad \tilde{u}_S = \left\lceil \frac{U(S)}{\epsilon g / n_v} \right\rceil.$$

Note that as for the NL model, we can preprocess the quantities and get a universal lower bounds on our guess in order to have polynomially many guesses (h, g). The rest of the construction is exactly similar to Algorithm 3 where instead of returning the best feasible solution, we store all the candidate assortment into a set $\mathcal{A}_v$.

2.4 Mixtures of multinomial logit model

We next study the assortment optimization problem for a mixture of MNL (mMNL) model which is given by a distribution over K different MNL models. For all $k \in [K]$ and $j \in [n]$, let $u_{j,k}$ denote the MNL parameters for segment k and θ_k denote the probability of segment k. For any $S \subseteq [n]$, $j \in S_+ = S \cup \{0\}$, the choice probability of product j is given by

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

Each product $i \in [n]$ has a price p_j and weight w_i. Let W denote the total available capacity. mMNL-Capa can be formulated as follows.

$$\max_{S \subseteq [n]} \left\{ \sum_{k=1}^{K} \theta_k \frac{\sum_{j \in S}^{n} p_j u_{j,k}}{u_{0,k} + \sum_{j \in S} u_{j,k}} \;\middle|\; \sum_{j \in S} w_j \leq W \right\} \qquad \text{(mMNL-Capa)}$$

[66] show that without any constraint mMNL-Assort is NP-hard even when $K = 2$, i.e. for a mixture of two MNL models. We present a FPTAS for the mMNL-Capa problem when the number of mixtures is constant. The idea is similar to the FPTAS for MNL-Capa. Since the objective function is a sum of ratios instead of a single ratio, we guess the value of each numerator $(\sum_{j \in S^*} p_j u_{j,k})$ and each denominator $(\sum_{j \in S^*} u_{j,k})$, for an optimal solution, S^* within a factor of $(1 + \epsilon)$. We then try to find a feasible assortment (satisfying the capacity constraint) with the numerator and denominator values approximately equal to the guesses using a dynamic program. The algorithm is very similar to the FPTAS for MNL-Capa and we defer the details of the algorithm to Appendix A.1.

Theorem 2.6. *There is a fully polynomial time approximation scheme (FPTAS) for mMNL-Capa when the number of mixtures, K, is constant.*

The running time of our algorithm is exponential in the number of mixtures K. We next show that a super polynomial dependence on K is necessary for any near-optimal algorithm. In other words, there exist no near optimal algorithm whose running time depends polynomially on K.

2.4.1 Hardness for arbitrary number of mixtures

We show that even without any constraint, mMNL-Assort is hard to approximate within any reasonable factor when the number of MNL segments, K is not constant. In particular, we show that there is no polynomial time algorithm (polynomial in n, K and the input size) with an approximation factor better than $O(1/K^{1-\delta})$ for any constant $\delta > 0$ for mMNL-Assort unless $NP \subseteq BPP$. This implies that if we require a near-optimal algorithm for mMNL-Assort, a super-polynomial dependence on the number of mixtures is necessary.

[3] show that the assortment optimization problem is hard to approximate within a factor of $O(1/K^{1-\delta})$ for any $\delta > 0$ when the choice model is given by a distribution over K rankings by an approximation preserving reduction from the independent set problem. We adapt the reduction in [3] to show a hardness of approximation mMNL-Assort.

Theorem 2.7. *There is no polynomial time algorithm (polynomial in n, K and the input size) that approximates mMNL-Assort within a factor $O(1/K^{1-\delta})$ for any constant $\delta > 0$ unless $NP \subseteq BPP$.*

Proof. We prove this by a reduction from the independent set problem. In a maximum independent set problem, we are given an undirected graph $G = (V, E)$ where $V = \{v_1, \ldots, v_n\}$. The goal is to find a maximum cardinality subset of vertices that are independent.

We construct an instance of mMNL-Assort as follows. We have one product and one MNL segment corresponding to each vertex in G. Therefore, $n = K = |V|$ in the MMNL model. For any MNL segment k corresponding to $v_k \in V$, we only consider a subset of products corresponding to a subset of neighbors of v_k in G. In particular, we consider the following utility parameters.

$$u_{j,k} = \begin{cases} 1 & \text{if } j = k \text{ or } j = 0 \\ n^2 & \text{if } (v_j, v_k) \in E \text{ and } j < k \\ 0 & \text{otherwise} \end{cases} \tag{2.15}$$

$$p_i = n^{3(i-1)},\ i \in [n]$$

$$\theta_k = \frac{\theta}{n^{3(k-1)}},\ k \in [n]$$

where $\theta \in [1/2, 1]$ is an appropriate normalizing constant. Note that the utility of any product $j \in [n]$ for segment $k \in [n]$, $u_{j,k} > 0$ only if $(v_j, v_k) \in E$ and $j < k$.

We first show that if there is an independent set, $\mathcal{I} \subseteq V$ where $|\mathcal{I}| = t$, we can find an assortment with revenue $\theta t/2$. Consider the set of products, S corresponding to vertices in $\mathcal{I}$, i.e.,

$$S = \{j \mid v_j \in \mathcal{I}\}.$$

Then, it is easy to observe that the revenue of S is exactly $\theta \cdot t/2$.

Next, we show that if there is an assortment S with expected revenue $R(S)$, then there exists an independent set of size at least $\lfloor 2 \cdot R(S)/\theta \rfloor$. For any segment $k \in [K]$, let R_k denote the contribution of segment k to the expected revenue of assortment S, i.e.,

$$R_k = \theta_k \cdot \frac{\sum_{j \in S} p_j u_{j,k}}{u_{0,k} + \sum_{j \in S} u_{j,k}}, \text{ and } R(S) = \sum_{k=1}^{K} R_k.$$

We show $R_k \geq \theta/2$ or $R_k \leq (2\theta)/n^2$. Let

$$N(k) = \{j \mid (v_j, v_k) \in E,\ j < k\}.$$

Case 1 ($N(k) = \emptyset$): If $k \notin S$, then $R_k = 0$. On the other hand, if $k \in S$, then

$$R_k = \theta_k \cdot \frac{p_k u_{k,k}}{u_{0,k} + u_{k,k}} = \frac{\theta}{n^{3(k-1)}} \cdot \frac{n^{3(k-1)}}{2} = \frac{\theta}{2}. \tag{2.16}$$

Case 2 ($N(k) \neq \emptyset$): In this case, $|N(k)| \geq 1$. Therefore,

$$R_k = \frac{\theta}{n^{3(k-1)}} \cdot \frac{n^{3(k-1)} + n^2 \cdot \sum_{j \in N(k)} n^{3(j-1)}}{2 + |N(k)| \cdot n^2} \leq \frac{2 \cdot \theta}{n^2}.$$

Therefore,

$$\left(|\{k \in S \mid N(k) = \emptyset\}| \cdot \frac{\theta}{2} \right) \ \leq \ R(S) \ \leq \ \left(|\{k \in S \mid N(k) = \emptyset\}| \cdot \frac{\theta}{2} \right) + \frac{2 \cdot \theta}{n}. \tag{2.17}$$

We can now construct an independent set, $\mathcal{I}$ as follows:

$$\mathcal{I} = \{v_k \in V \ \mid k \in S, N(k) = \emptyset\}.$$

We claim that $\mathcal{I}$ is an independent set. For the sake of contradiction, suppose there exist $v_i, v_j \in \mathcal{I}$ $(i < j)$ such that $(v_i, v_j) \in E$. Since $v_i, v_j \in \mathcal{I}$, $i, j \in S$ and $N(i) = N(j) = \emptyset$. Moreover, since $i < j$ and $(v_i, v_j) \in E$, $i \in N(j)$ which implies $N(j) \neq \emptyset$; a contradiction. Therefore, $\mathcal{I}$ is an independent set. Also,

$$|\mathcal{I}| = |\{k \in S \mid N(k) = \emptyset\}| = \left\lfloor \frac{2 \cdot R(S)}{\theta} \right\rfloor,$$

where the second equality follows from (2.17). Therefore, if $\mathcal{I}^*$ is the optimal independent set and R^* is the optimal expected revenue of the corresponding mMNL-Assort instance (2.15), then

$$\left\lfloor \frac{2 \cdot R^*}{\theta} \right\rfloor \leq |\mathcal{I}^*| \leq \frac{2 \cdot R^*}{\theta}.$$

Consequently, an α-approximation for MMNL-Assort implies an $O(\alpha)$-approximation for the maximum independent set problem. Since the maximum independent set is hard to approximation within a factor better than $O(1/n^{1-\delta})$ (where $|V| = n = K$) for any constant $\delta > 0$ (see [29]), the above reduction implies the same hardness of approximation for mMNL-Assort. □

The above theorem shows that mMNL-Assort is hard to approximate. The approximation preserving reduction from the independent set problem gives several interesting insights. First, note that each MNL segment in the reduction only contains a subset of products corresponding to a subset of vertices in the neighborhood of the corresponding vertex. This is quite analogous to the consideration set model considered in [39] where a local neighborhood defines the consideration set. Such graphical model based consideration set instances are quite natural and our reduction shows that mMNL-Assort is hard even for these naturally occurring instances. Therefore, our reduction gives a procedure to construct naturally arising hard benchmark instances of mMNL-Assort that may be of independent interest.

We can extend the hardness of approximation even for the continuous relaxation of mMNL-Assort.

Theorem 2.8. *Consider the following continuous relaxation of the* mMNL-Assort *problem.*

$$\max_{\boldsymbol{x}\in[0,1]^n} \left\{ \sum_{k=1}^{K} \theta_k \frac{\sum_{j=1}^{n} p_j u_{j,k} x_j}{u_{0,k} + \sum_{j=1}^{n} u_{j,k} x_j} \right\} \tag{2.18}$$

There is no approximation algorithm (with running time polynomial in K) that has an approximation factor better than $O(1/K^{1-\delta})$ for any constant $\delta > 0$ unless $NP \subseteq BPP$.

We present the proof in Appendix A.3.

In this chapter, we have studied Capa and provided a flexible algorithmic framework to derive FPTAS for various RUM models. For these models, a near-optimal algorithm is best possible since even MNL-Capa is NP-hard. Moreover, for mMNL-Assort, we strengthen the known hardness result (mMNL-Assort is NP-hard under a mixture of 2 MNL) and show that when the number of mixtures is arbitrary, the problem becomes hard to approximate within any reasonable factor. In particular, this precludes a polynomial dependence on the number of mixtures. Recall that from a richness standpoint, the MNL model and the mixture of MNL model sit at the two extremes of the spectrum within the class of RUM models. The stark difference between MNL-Assort and mMNL-Assort in terms of tractability for even the simplest problem Assort is a great example of tradeoff between predictive power and tractability.

Chapter 3

Approximation algorithms for assortment optimization problems under a Markov chain based choice model

In this chapter, we focus on assortment optimization problems under a Markov chain based choice model. MC-Assort admits a polynomial time algorithm through an LP reformulation of the problem [10]. We start by showing in Section 3.2 that adding a cardinality constraint makes the problem much harder. In particular, MC-Card is NP-hard to approximate within a factor better than some given constant, even when all products have uniform prices. It is interesting to note that, while MC-Assort can be solved optimally in polynomial time, MC-Card is APX-hard. In contrast, in both the MNL and NL models, Assort and Card have the same complexity. We also consider the case of totally-unimodular (TU) constraints on the assortment. We show that MC-TU is hard to approximate within a factor of $O(n^{1/2-\epsilon})$ for any fixed $\epsilon > 0$, where n is the number of products. This result drastically contrasts that of [23], who prove that MNL-TU model can be solved in polynomial time.

The harness results motivate us to consider approximation algorithms for MC-Card and MC-Capa. For the special case when all product prices are equal, we show in Section 3.3.1 that we can obtain a $(1-1/e)$-approximation for MC-Card using a greedy algorithm. In fact, for this special case of uniform prices, we can get a $(1-1/e)$-approximation for more general constraints such as a constant number of capacity constraints and matroid constraint.

In Section 3.3.2 we show that a simple greedy algorithm fails when the prices are arbitrary. This motivates us to consider an alternative approach to solving this problem. In particular, we introduce a new algorithmic framework in Section 3.3.3. The algorithm is based on a "local-ratio" paradigm that builds the solution iteratively. In each iteration, the algorithm makes an appropriate greedy choice and then constructs a modified instance such that the final objective value is the sum of the objective value of the current solution and the objective value of the solution in the modified instance. Therefore, the local-ratio paradigm allows us to capture the externality of our action in each iteration on the remaining instance by constructing an appropriate modified instance; thereby, linearizing the revenue function even though the original objective function is non-linear. This technique may be of independent interest.

We next show how to use this framework to solve various assortment optimization problems. In Section 3.4, we give an alternative exact algorithm to MC-Assort. Section 3.5 gives a 2-approximation algorithm for MC-Card and 3.6 gives a 3-approximation algorithm for MC-Capa. On top of these worst-case guarantees, we show in Section 3.7 through numerical experiments that our constant factor algorithm exhibit very good practical performance (both in terms of approximation and running time). Finally, in Section 3.8, we consider a robust variant of the assortment optimization problem (MC-Rob) and show how similar ideas can be applied to design an exact algorithm for this setting. Furthermore, we give insights into the structure of the optimal assortment for MC-Rob.

Finally, in Section 3.9, we consider a special case of Markov chain model when the underlying Markov chain has a constant rank. Under this extra assumption, we can leverage the tools developed in Chapter 2 to design a near optimal algorithm for MC-Capa. The running time of the algorithm is exponential in the rank of the underlying Markov chain. We therefore obtain an FPTAS only when the rank is constant.

3.1 Markov chain model

We denote the universe of n products by the set $\mathcal{N} = \{1, 2, \ldots, n\}$ and the no-purchase option by 0, with the convention that $\mathcal{N}_+ = \mathcal{N} \cup \{0\}$. We consider a Markov chain $\mathcal{M}$ with states $\mathcal{N}_+$ to model the substitution behavior of customers. This model is completely specified by initial arrival probabilities λ_i for all states $i \in \mathcal{N}_+$ and the transition probabilities ρ_{ij} for all $i \in \mathcal{N}_+, j \in \mathcal{N}_+$. If a retailer chooses to offer a subset of products S to consumers, then the corresponding states in S of the Markov chain become absorbing states. A customer arrives in state i with probability λ_i if the state is absorbing. Otherwise, the customer transitions to a different state $j \neq i$ and the process continues until the customer reaches an absorbing state. In other words, the probability of a random customer purchasing product i with S being the offer set of products is the probability that the customer reaches state i before any other absorbing states in the underlying Markov chain. As before, let p_i denote the price of product i.

Following [10], we assume that for each state $j \in \mathcal{N}$, there is a path to state 0 with non-zero probability. For a given offer set $S \subseteq \mathcal{N}$, let $\pi(i, S)$ be the choice probability that item i is chosen when the assortment S is offered. We have

$$\pi(i, S) = \lambda_i + \sum_{j \notin S} \lambda_j \rho_{j,i} + \sum_{j \notin S, k \notin S} \lambda_j \rho_{j,k} \rho_{k,i} + \ldots$$

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

3.2 Hardness of approximation

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

3.2.1 APX-hardness for cardinality constraint with uniform prices

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

Theorem 3.1. *MC-Card is APX-hard, even when all products have equal prices.*

Proof. We establish the claim via a gap preserving reduction from minimum vertex cover on 3-regular (or cubic) graphs. We refer to this problem as VCC. This problem is known to be APX-hard (see [1]). In other words, for some constant $\alpha > 0$, it is NP-hard to distinguish whether the minimum-cardinality vertex cover is of size at most k or at least $(1+\alpha)k$ for cubic graphs.

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

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

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

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

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

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

where S^* is the optimal assortment for $\mathcal{M}(\mathcal{I})$. This implies that MC-Card cannot be approximated within factor larger than $1 - \frac{\alpha}{16}$, unless P = NP. To see this, note that the ratio between $\frac{3}{4} + \frac{k}{4n} - \frac{\alpha}{16}$ and $\frac{3}{4} + \frac{k}{4n}$ is monotone-increasing in k, meaning that the maximum value attained is $1 - \frac{\alpha}{16}$.

Case (a): $|U^*| \leq k$. In this case, we can augment U^* with $k - |U^*|$ additional vertices chosen arbitrarily from $V \setminus U^*$, and obtain a (not-necessarily minimum) vertex cover U with $|U| = k$. Now, consider the assortment $S = \{i : v_i \in U\}$, which is indeed a feasible solution. Since all prices are equal to 1, we can write the expected revenue of this set as

$$R(S) = \mathbb{P}(S \prec 0) = \sum_{i \in S} \lambda_i + \sum_{i \notin S} \lambda_i \mathbb{P}_i(S \prec 0) = \frac{k}{n} + \frac{1}{n} \sum_{i \notin S} \mathbb{P}_i(S \prec 0). \qquad (3.1)$$

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

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

Case (b): $|U^*| \geq (1+\alpha)k$. Let S be some assortment consisting of k products. In this case, equation (3.1) is still a valid decomposition of $R(S)$, and we need to consider two cases for products $i \notin S$. If $N(i) \subseteq S$, then $\mathbb{P}_i(S \prec 0) = 3/4$ as in case (a). However, when $N(i) \nsubseteq S$, there exists $j \in N(i)$ such that $j \notin S$. Therefore, there is a probability of 1/16 that starting from i the Markov chain moves to j and from there to 0. Consequently, for such items, $\mathbb{P}_i(S \prec 0) \leq \frac{3}{4} - \frac{|N(i)\backslash S|}{16}$. Therefore,

$$\begin{aligned} R(S) &= \frac{k}{n} + \frac{1}{n} \sum_{i \notin S, N(i) \subseteq S} \frac{3}{4} + \frac{1}{n} \sum_{i \notin S, N(i) \nsubseteq S} \mathbb{P}_i(S \prec 0) \\ &\leq \frac{3}{4} + \frac{k}{4n} - \frac{1}{16n} \sum_{i \notin S, N(i) \nsubseteq S} |N(i)\backslash S|. \end{aligned} \tag{3.2}$$

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

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

where the second inequality follows from $k > |E|/3$, and the last equality holds since $|E| = 3n/2$, as G is cubic. By inequality (3.2), we have

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

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

3.2.2 Totally-unimodular constraints

We show that MC-TU is NP-hard to approximate within factor $O(n^{1/2-\epsilon})$, for any fixed $\epsilon > 0$ for the Markov chain model. This result drastically contrasts that of [23], who proved that the assortment optimization problem with totally-unimodular constraints can be solved in polynomial time when consumers choose according to the MNL model.

Theorem 3.2. *MC-TU cannot be approximated in polynomial-time within a factor $O(n^{1/2-\epsilon})$, for any fixed $\epsilon > 0$, unless $P = NP$.*

To establish our inapproximability results for MC-TU, we demonstrate that totally-unimodular constraints in the Markov chain model capture the distribution over rankings model as a special case. [3] show that even Assort under a general distribution over rankings model is hard to approximate within factor $O(n^{1-\epsilon})$ for any fixed $\epsilon > 0$ (n is the number of substitutable products). We present the proof in Appendix B.1.

3.3 Local ratio based algorithm design

3.3.1 Special case: uniform price products

When all prices are equal, we show that the revenue function is submodular and monotone. Using the classical result of [59], we have that a greedy algorithm guarantees a $(1-1/e)$-approximation for MC-Card for this special case of uniform prices. We start with a few definitions. It is worth mentioning that, from a practical point of view, the uniform-price setting turns the objective function into that of maximizing sales probability. This scenario is very common when products are horizontally-differentiated, i.e., differ by characteristics that do not affect quality or price, such as iPads coming in a variety of colors, or yogurt with different amounts of fat-content.

Definition 3.1. *A revenue function $R : 2^{\mathcal{N}} \rightarrow \mathbb{R}_+$ is monotone when for all $S \subseteq \mathcal{N}$ and $i \in \mathcal{N}$, we have $R(S \cup \{i\}) \geq R(S)$.*

Definition 3.2. *A revenue function $R : 2^{\mathcal{N}} \rightarrow \mathbb{R}_+$ is submodular when for all $S \subseteq T \subseteq \mathcal{N}$ and $i \in \mathcal{N} \backslash T$, we have $R(S \cup \{i\}) - R(S) \geq R(T \cup \{i\}) - R(T)$.*

Theorem 3.3. *When all products have uniform prices, the revenue function $R(\cdot)$ is submodular and monotone.*

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

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

Recall that $\mathbb{P}(i \prec 0 \prec S)$ is the probability that the Markov chain visits state i and then visits state 0 without visiting any state in S. When all prices are equal, the marginal increase in revenue by adding product i is only due to the additional demand that product i is able to capture. Consequently, $R(\cdot)$ is monotone as the quantity $p \cdot \mathbb{P}(i \prec 0 \prec S)$ is non-negative. Moreover, the submodularity of R follows from the fact that for all $S \subseteq T$, we have

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

□

Therefore, from the classical result of [59] for maximizing a monotone submodular function subject to a cardinality constraint, we know that the greedy algorithm gives a $(1 - 1/e)$-approximation bound for MC-Card with uniform prices. Algorithm 4 describes this procedure in detail. Note that for uniform prices, when $|S| < k < n$,

Algorithm 4 Greedy Algorithm

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

the condition in Step 2 that there exists $i \in \mathcal{N} \backslash S$ such that $R(S \cup \{i\}) - R(S) \geq 0$ is redundant as the revenue function is monotone, which is not necessarily true for the case of arbitrary prices. We include this condition to describe the greedy algorithm for the general case and to discuss implications for arbitrary prices.

More general constraints for uniform prices. For the special case of uniform prices, since the revenue function is monotone and submodular, we can exploit the existing machinery for approximately maximizing submodular monotone functions subject to a wide range of constraints (see, for instance, [47], [14], [45], [17]). This way, constant-factor approximations can be obtained for the assortment optimization under the Markov chain model for more general constraints. For instance, [45] give a $(1-1/e)$-approximation algorithm for maximizing a monotone submodular function under a fixed number of knapsack (capacity) constraints, and [17] give a $(1-1/e)$-approximation for maximizing a monotone submodular function under a matroid constraint.

3.3.2 Bad examples for arbitrary prices

The approximation guarantees we establish for uniform prices do not extend to the more general setting with arbitrary prices, even for MC-Card. In what follows, we point out the drawbacks of the natural greedy heuristics, including Algorithm 4, in approximating MC-Card for arbitrary prices. Intuitively, the performance of Algorithm 4 for general prices can be bad since it can make a low-price product absorbing that subsequently blocks all probabilistic transitions going into high price products. We formalize this intuition in the following lemma.

Lemma 3.1. *For arbitrary instances of MC-Card with a cardinality constraint of k, Algorithm 4 can compute solutions whose expected revenue is only $O(1/k)$ times the optimum.*

Proof. Consider the following instance of MC-Card with $n = k + 1$ items, where k is the upper bound specified by the cardinality constraint. We have a state s and states

$i = 0, \dots, k$. The arrival rates are all equal to 0, except for λ_s which is equal to 1. Moreover

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

where $\epsilon \leq 1/(2k)$. Figure 3.1 provides a graphical representation of this instance. Algorithm 4 first picks item s as $R(\{s\}) = (1/k) + \epsilon$ while $R(\{i\}) = (1/k)$, for

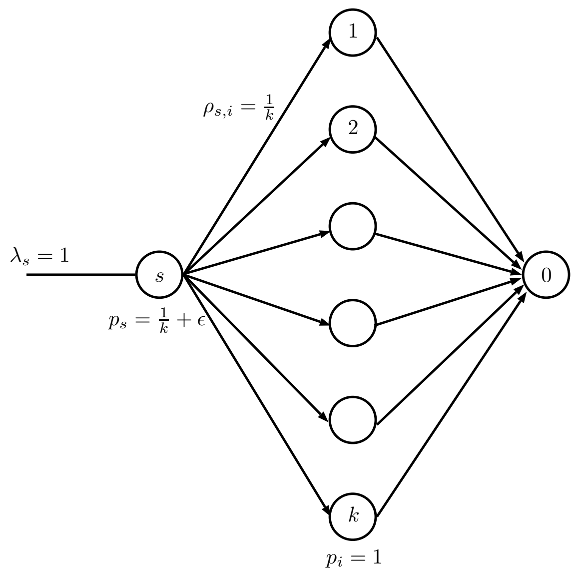


Figure 3.1: A bad example for Algorithm 4.

$i = 1, \dots k$. Once s is selected, adding any other state cannot increase the revenue. Therefore, the greedy algorithm gives a revenue of $(1/k) + \epsilon$. However, the optimal solution is to offer items 1 to k, which gives a revenue of 1 in total. When ϵ tends to 0, the approximation ratio goes to $1/k$. $\square$

In fact, we can show that the above example is the worst possible and Algorithm 4 gives a $1/k$-approximation for MC-Card.

Lemma 3.2. *Algorithm 4 guarantees a $1/k$-approximation for MC-Card.*

We present the proof of the above lemma in Appendix B.2.

Modified greedy algorithm. The bad instance for Algorithm 4 shows that the algorithm may focus too much on local improvements in each iteration, without taking into account the information of the entire network induced by the probability transition matrix or the number of remaining iterations. Therefore, we consider a modified greedy algorithm that accounts for the Markov chain structure by using the optimal solution to the unconstrained assortment problem, where there is no restriction on the number of products picked. This solution can be computed via an algorithm proposed by [10] (we also give an alternative strongly-polynomial algorithm for the unconstrained problem in Section 3.4). Intuitively, the products picked by the unconstrained optimal assortment should not block each other's demand too much. Let U^* be the optimal unconstrained assortment whose associated revenue can be written as

$$R(U^*) = \sum_{i \in U^*} \mathbb{P}(i \prec U^*_+ \backslash \{i\}) \cdot p_i. \tag{3.3}$$

A natural candidate algorithm takes the k products with the largest $\mathbb{P}(i \prec U^*_+ \backslash \{i\}) \cdot p_i$ value within an unconstrained optimal solution, and sets these states to be absorbing. Algorithm 5 describes this procedure.

Algorithm 5 Greedy Algorithm on Optimal Unconstrained Assortment

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

We show in the following lemma that even Algorithm 5 performs poorly in the worst case. In fact, we present an example where every subset of k items of the optimal solution U^* has revenue a factor k away from the optimal.

Lemma 3.3. *There are instances where the revenue obtained by Algorithm 5 is far from optimal by a factor of $k/|U^*|$ where k is the upper bound in the cardinality constraint.*

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

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

where $\epsilon > 0$. Figure 3.2 provides a graphical representation of this instance. For this

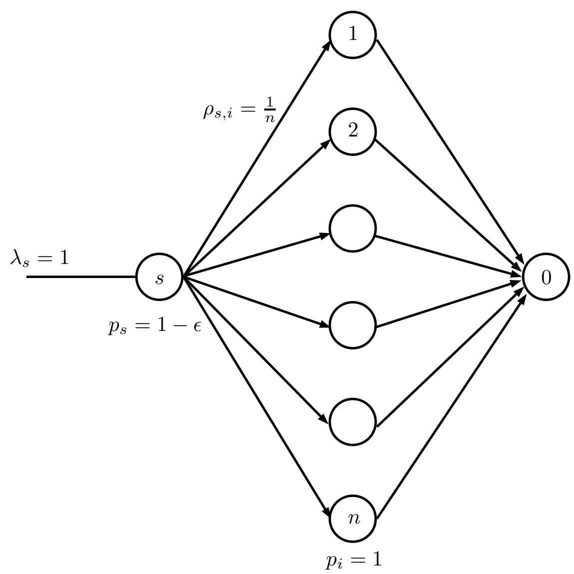


Figure 3.2: A bad example for Algorithm 5.

example, the unconstrained optimal assortment is $U^* = \{1, \ldots, n\}$, and the greedy algorithm on U^* selects k products among U^*, meaning that a total revenue of k/n is obtained. However, the optimal solution of the constrained problem is to only offer item s, which gives a revenue of $1-\epsilon$. As ϵ tends to 0, the approximation ratio goes to $k/|U^*|$. □

The poor performance of Algorithm 5 on the above example illustrates that an optimal assortment for the constrained problem may be very different from that of its unconstrained counterpart. Hence, searching within an unconstrained optimal solution for a good approximate solution to the constrained problem can be unfruitful in general. It is worth noting that the lower bound of $k/|U^*|$ for Algorithm 5 is tight, as stated in the following lemma, whose proof is given in Appendix B.3.

Lemma 3.4. *Algorithm 5 guarantees a $k/|U^*|$-approximation algorithm to* MC-Card.

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

3.3.3 High-level ideas for algorithm design

As the example in Figure 3.1 illustrates, Algorithm 4 could end up with a highly suboptimal solution due to picking products that cannibalize, i.e. block, the demand for higher price products. Picking the highest price product will eliminate such a concern. However, a high price product might only capture very little demand, and therefore, generate very small revenue as illustrated in the example in Figure 3.2. When there is a capacity constraint on the assortment, picking such products may not be an optimal use of the capacity. This motivates us to choose the highest price product in an appropriate consideration set. Intuitively, the consideration set will consist of products that generate sufficiently high incremental revenue.

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

1. *Greedy Selection.* Define an appropriate consideration set C_t of products, and pick the "highest price" product from C_t.

2. *Instance Update.* Construct a new instance, $\mathcal{M}_{t+1}$, of the constrained assortment optimization problem with appropriately modified product prices and

transition probabilities such that

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

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

The instance update step linearizes the revenue function even though the original revenue function is non-linear, which is crucial for our iterative solution approach. We can also view the update rule as a framework to capture the externality of our actions in each iteration of the algorithm. To completely specify the algorithm, we need to provide a precise definition for the consideration set in the greedy step and for the instance update step. For both MC-Card and MC-Capa, the instance update step is similar, as explained in Section 3.3.4. The consideration set, however, depends on the particular optimization problem being considered and will be defined later on. The intuition is to include products whose incremental revenue is above an appropriately chosen threshold. Our algorithm can be viewed in a local-ratio framework (see, for instance, [6], [5] and [7]). Therefore, we will interchangeably refer to the instance updates as local-ratio updates. However, we would like to note that the local-ratio framework does not provide a general recipe for designing an update rule or analyzing the performance bound. In most algorithms in this framework, the update rule follows from a primal-dual algorithm. However, for MC-Capa, we do not even know of any good LP formulation and the instance update rule requires new ideas.

3.3.4 Instance update in local ratio algorithm

Notation. Given an instance $\mathcal{M}$ of the Markov chain model, we define an updated instance $\mathcal{M}(S)$ given that S is made absorbing by modifying the product prices as well as the probability transition matrix. Note that we index the updates by a set

S. Therefore, the instance $\mathcal{M}_t$ introduced in the preceding discussion is going to be thought of as $\mathcal{M}(S_{t-1})$, where S_{t-1} denotes the set of products picked up to (and including) step $t-1$. For an instance $\mathcal{M}(S)$, we will denote by p_i^S the updated price of product i, and by ρ_{ij}^S the updated transition probabilities for every $i \in \mathcal{N}, j \in \mathcal{N}_+$. Note that we do not change the arrival rate to any state, i.e., $\lambda_i^S = \lambda_i$ for all $i \in \mathcal{N}$. We also denote by $R^S : 2^{\mathcal{N}} \to \mathbb{R}$ the revenue function associated with the instance $\mathcal{M}(S)$ and by $\mathbb{P}^S(\cdot)$ the probability of any event with respect to the instance $\mathcal{M}(S)$.

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

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

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

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

which we denote as the adjusted price p_i^S. Note that the adjusted prices can be negative, corresponding to the situation where adding a product decreases the overall revenue. The price update is explicitly described in Figure 3.3.

Transition probabilities update. Since the subset of states S is set to be absorbing, we will simply redirect the outgoing probabilities from all states in S to 0. This is described in Figure 3.3.

We would like to note that the probabilities $\mathbb{P}_i(j \prec S_+ \backslash \{j\})$, needed for our price updates, can be interpreted as the choice probability $\pi(j, S)$ for a modified instance

Price update:

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

Transition probabilities update:

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

Figure 3.3: Instance update in local-ratio algorithm.

with $\lambda_i = 1$ and $\lambda_\ell = 0$ for $\ell \neq i$. Therefore, these quantities can be efficiently computed via traditional Markov chain tools (see, for instance, [**?**]).

3.3.5 Structural properties of the updates

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

Lemma 3.5. *$R(S_1 \cup S_2) = R(S_1) + R^{S_1}(S_2)$ for every $S_1, S_2 \subseteq \mathcal{N}$.*

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

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

With the definition of ρ^{S_1}, note that all products of S_1 are redirected to 0. This, together with the fact that $S_1 \cap S_2 = \emptyset$ implies that for all $i \in S_2$, we have $\mathbb{P}^{S_1}(i \prec$

$S_{2+}\backslash\{i\}) = \mathbb{P}(i \prec (S_2 \cup S_1)_+\backslash\{i\})$. Consequently,

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

where the second equality holds since

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

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

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

Lemma 3.6. *Let $S_1 \subseteq \mathcal{N}$ be some assortment, and let $\mathcal{M}_1 = \mathcal{M}(S_1)$. For any S_2 with $S_1 \cap S_2 = \emptyset$, the instance $\mathcal{M}_1(S_2)$ is identical to the instance $\mathcal{M}(S_1 \cup S_2)$ in terms of product prices and transition probabilities.*

It suffices to verify that $(p_i^{S_1})^{S_2} = p_i^{S_1\cup S_2}$ for all S_1,S_2 and $i \notin S_1\cup S_2$, as the above identity clearly holds for the transition matrix updates. The proof is similar to that of Lemma 3.5, and is presented in Appendix B.4. Putting the previous two lemmas together gives the following claim.

Lemma 3.7. $R^{S_1}(S_2 \cup S_3) = R^{S_1}(S_2) + R^{S_1 \cup S_2}(S_3)$ *for any pairwise-disjoint sets* $S_1, S_2, S_3 \subseteq \mathcal{N}$.

3.4 Unconstrained assortment optimization

As a warmup, we first present an alternative exact algorithm for MC-Assort by using the local-ratio framework. Our algorithm is based on the observation that it is always optimal to offer the highest price product for the unconstrained problem, as it does not cannibalize the demand of other products. The latter property is implied by a slightly more general claim, formalized as follows. For any $x \in \mathbb{R}$, let $[x]^+ = \max(x, 0)$.

Lemma 3.8. *Let* $S \subseteq \mathcal{N}$. *For any product* $i \notin S$ *with price* $p_i \geq [\max_{j \in S} p_j]^+$, *we have* $R(S \cup \{i\}) \geq R(S)$.

Proof. From Lemma 3.5, we have that

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

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

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

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

The Algorithm. Based on the above lemma, we present an alternative exact algorithm for MC-Assort. In particular, we define the consideration set in each iteration to be the set of all products. Therefore, we select the highest adjusted price product in every iteration (breaking ties arbitrarily) and update the prices and transition probabilities according to the local ratio updates described in Figure 3.3. This selection and updating process is repeated until all adjusted prices are non-positive, as explained in Algorithm 6.

Algorithm 6 Local Ratio for Unconstrained Assortment

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

Theorem 3.4. *Algorithm 6 computes an optimal solution for* MC-Assort*.*

Proof. The correctness of Algorithm 6 is based on the observation that it is always optimal to offer the highest adjusted price product, as long as this price is non-negative. Suppose product 1 is the highest price product. From Lemma 3.8, we get $R(S \cup \{1\}) \geq R(S)$ for any assortment S. Therefore, we can assume that product 1 belongs to the optimal assortment. From Lemma 3.5, we can write

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

It remains to show that, when we get to an iteration where our current absorption set is X, and the adjusted price of every state in the modified instance $\mathcal{M}(X)$ is non-positive, then X is an optimal solution to $\mathcal{M}$. To see this, by repeated applications of Lemmas 3.5 and 3.6, we have

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

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

Implications. Our algorithm for MC-Assort provides interesting insights for some known results about both the optimal stopping problem and MNL-Assort. [10] relate MC-Assort to the optimal stopping time on a Markov chain (see [20]). In this problem, we need to decide at each state i whether to stop and get the reward p_i, or transition according to the transition probabilities of the Markov chain. Moreover, there is an absorbing state 0 with price $p_0 = 0$. Algorithm 6 for MC-Assort gives an alternative strongly polynomial time algorithm for the optimal stopping problem.

[10] prove that the MNL choice model is a special case of the Markov chain based choice model. By analyzing Algorithm 6 to solve MNL-Assort, we can recover the structure of the optimal assortment being nested by prices, i.e., the optimal assortment consists of the ℓ top-priced items for some ℓ. We give an explicit expression for our local ratio updates when the underlying choice model is MNL in Appendix B.5.

3.5 Cardinality constrained assortment optimization

In this section, we present a $(1/2-\epsilon)$-approximation for MC-Card, for any fixed $\epsilon > 0$. Following the local-ratio framework described in Section 3.3.3, our algorithm for the cardinality constrained case also selects a product with high adjusted price in each step from an appropriate consideration set. The consideration set is defined to avoid picking products that have a high adjusted price but capture very little demand. In particular, the consideration set includes only products whose incremental revenue is at least a certain threshold.

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

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

where S^* is the optimal solution, k is the cardinality bound, and $\alpha \in (0,1)$ is a parameter whose value will be optimized later. Note that C_t is defined at the beginning of step t, whereas S_t is defined at the end of step t, and includes the product selected in this step. Once the item σ_t is selected, we recompute the adjusted prices via the local ratio update described in Figure 3.3, and update the consideration set to get C_{t+1}. The algorithm terminates when either k products have already been picked (i.e., upon the completion of step k), or when the consideration set C_t becomes empty.

Guessing the value of $R(S^*)$. Since the optimal revenue $R(S^*)$ is not known a-priori, we need to describe how the value of $R(S^*)$ is approximately guessed to complete the algorithm's description. A natural upper bound for $R(S^*)$ is $R(U^*)$, when U^* is the optimal unconstrained solution. From Lemma 3.4, we know that $R(S^*) \geq \frac{k}{|U^*|} R(U^*)$. Now, given an accuracy parameter $0 < \epsilon < 1$, let

$$\begin{aligned} B_j &= \frac{k}{|U^*|} R(U^*)(1+\epsilon)^j, \quad j = 1, \ldots, J \\ J &= \min \{ j \in \mathbb{N} \; : \; B_j \geq R(U^*) \} . \end{aligned} \tag{3.4}$$

Note that $J = O(\frac{1}{\epsilon} \log k)$. For each guess B_j for the true value of $R(S^*)$, we run the algorithm, and eventually return the best solution found over all runs. Algorithm 7 describes the resulting procedure for a particular choice of B_j and threshold α for the consideration set. Algorithm 8 describes the full procedure for any given $\epsilon > 0$.

Algorithm 7 Algorithm with guess B_j and threshold α

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

3.5.1 Technical lemmas

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

Algorithm 8 Local-ratio Algorithm for MC-Card with threshold α

1: Given any $\epsilon > 0$, let J and B_j, $j \in [J]$ be as defined in (3.4).
2: For all $j \in [J]$, let S_j be the solution returned by Algorithm 7 with guess B_j and threshold α
3: Return $\arg\max_{j\in[J]} R(S_j)$.

product prices.

Lemma 3.9. *For all $S_1, S_2 \subseteq \mathcal{N}$ consisting only of non-negative priced products, $R(S_1 \cup S_2) \leq R(S_1) + R(S_2)$.*

Proof. We have that

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

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

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

- $Z_t = S^* \cap C_t$ denotes the products of S^* which are in the consideration set C_t.
- $Y_t = Z_{t-1} \setminus Z_t$ denotes the products of S^* which disappear from the consideration set during step $t - 1$.

- $Y_t^+ = \{i \in Y_t : p_i^{S_{t-1}} \geq 0\}$ denotes the products of Y_t which have a non-negative adjusted price at step t.

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

Lemma 3.10. *For all* $t \geq 1$, $R(S_t) - R(S_{t-1}) \geq R^{S_{t-1}}(Z_t) - (R^{S_t}(Z_{t+1}) + R^{S_t}(Y_{t+1}^+))$.

Proof. Recall that, by definition, Z_t contains the products of S^* that are in the consideration set at the beginning of step t. Since our algorithm picks the highest adjusted price product, σ_t, in the consideration set C_t, we have $p_{\sigma_t}^{S_{t-1}} \geq p_i^{S_{t-1}} \geq 0$ for all products $i \in Z_t$. Therefore, by Lemma 3.8,

$$R^{S_{t-1}}(Z_t) \leq R^{S_{t-1}}(Z_t \cup \{\sigma_t\}). \tag{3.5}$$

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

Case (a): $\sigma_t \notin S^*$**.** From Lemma 3.7, $R^{S_{t-1}}(Z_t \cup \{\sigma_t\}) = R^{S_{t-1}}(\{\sigma_t\}) + R^{S_t}(Z_t)$. Consequently, from inequality (3.5), we have

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

where the second inequality holds since removing all negative adjusted price products can only increase net revenue, and the last inequality follows from Lemma 3.9. Adding $R(S_{t-1})$ on both sides of the inequality yields the desired inequality by Lemma 3.5.

Case (b): $\sigma_t \in S^*$. From Lemma 3.7, $R^{S_{t-1}}(Z_t) = R^{S_{t-1}}(\{\sigma_t\}) + R^{S_t}(Z_t \backslash \{\sigma_t\})$. Then, similar to the previous case, we have

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

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

From the above result, we obtain the following claim.

Lemma 3.11. *For all $t \geq 0$, we have $R(S_t) \geq R(S^*) - (R^{S_t}(Z_{t+1}) + \sum_{j=1}^{t+1} R^{S_{j-1}}(Y_j^+))$.*

Proof. By summing the inequality stated in Lemma 3.10 over $j = 1, \ldots, t$, we obtain a telescopic sum which yields

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

Since every product in S^* must have non-negative price and $S^* = Z_1 \cup Y_1$ by definition, we have $R(S^*) \leq R(Z_1) + R(Y_1)$ by sublinearity of the revenue function (see Lemma 3.9). Combining these two inequalities concludes the proof. □

3.5.2 Analysis of the local-ratio algorithm

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

Theorem 3.5. *For any fixed $\epsilon > 0$, Algorithm 8 gives a $(1/2 - \epsilon/2)$-approximation for MC-Card. Moreover, the running time is polynomial in the input size and $1/\epsilon$.*

Proof. For a fixed $\epsilon > 0$, let j^* be such that $\frac{R(S^*)}{1+\epsilon} \leq B_{j^*} \leq R(S^*)$. Let $B = B_{j^*}$ and consider the solution returned by Algorithm 7 with guess B and threshold α. We consider two cases based on the condition by which the algorithm terminates.

1. If the algorithm stops after completing step k, then by linearity of the revenue when using the local ratio updates (Lemmas 3.5 and 3.6), the resulting solution S_k has a revenue of

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

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

2. Now, suppose the algorithm stops at the end of step $k' < k$, after discovering that $C_{k'+1} = \emptyset$. From Lemma 3.11, we get

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

Now, since $C_{k'+1} = \emptyset$, this implies that $Z_{k'+1} = \emptyset$. Moreover, from Lemma 3.9, we also have $R^{S_{j-1}}(Y_j^+) < |Y_j^+| \cdot \alpha \cdot B/k$ for all $j = 1, \dots, k'+1$. Therefore,

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

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

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

This shows that the approximation ratio attained by our algorithm is

$$\min\{(1-\epsilon)\alpha,\ 1-\alpha\}.$$

Picking $\alpha = 1/2$ we obtain a $(1/2 - \epsilon/2)$-approximation for MC-Card.

Running time. Algorithm 8 considers $J = O(\frac{1}{\epsilon}\log n)$ guesses for $R(S^*)$. For any given guess B_j, the running time of Algorithm 7 is polynomial in the input size. Therefore, the overall running time of Algorithm 8 is polynomial in the input size and $1/\epsilon$. □

Tight example. We show that Algorithm 8 is tight in the following sense: consider Algorithm 7 with input guess as the true value of $R(S^*)$ and threshold $\alpha = 1/2$, then there are instances for which the approximation ratio is $1/2$. In particular, we consider an instance with 3 products. The Markov chain has 4 states $\mathcal{N}_+ = \{s, 1, 2, 0\}$. The prices are: $p_s = 1, p_1 = p_2 = 2$. The arrival rate for state s is $\lambda_s = 1$ and all other states have an arrival rate of zero. The transition probabilities are given in Figure 3.4. Consider the cardinality constrained assortment problem with cardinality bound, $k = 1$. The optimal assortment is $S^* = \{s\}$ with $R(S^*) = 1$. With guess $R(S^*)$ and $\alpha = 1/2$, the consideration set in the first step is $\{s, 1, 2\}$, and therefore Algorithm 7 picks either 1 or 2, obtaining a revenue of $R(S^*)/2$.

We would like to note that our algorithm runs Algorithm 7 for different guesses $B_j, j = 1, \ldots, J$ and returns the best solution across all runs. Therefore, the performance bound of our algorithm is at least $(1/2 - O(\epsilon))$ and possibly better. In fact, in our computational study, we observe that the empirical performance of our algorithm is significantly better than the theoretical bound of $(1/2 - O(\epsilon))$. We describe the computational study in Section 3.7. It is an interesting open question to provide a tighter analysis of the approximation bound for Algorithm 8 that returns the best solution among several guesses of $R(S^*)$.

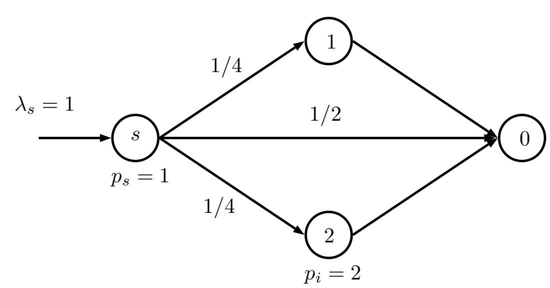


Figure 3.4: A tight example for Algorithm 8.

3.6 Capacity constrained assortment optimization

In this section, we show how to approximate MC-Capa within factor $1/3 - \epsilon$, for any fixed $\epsilon > 0$. Recall that, unlike the simpler cardinality case, now each product i has an arbitrary weight w_i, and we have an upper bound W on the available capacity. We assume without loss of generality that each product individually satisfies the capacity constraint, i.e., $w_i \leq W$ for all $i \in \mathcal{N}$.

The Algorithm. We describe a local-ratio based algorithm, similar in spirit to the one for the cardinality constrained problem, by suitably adapting the way consideration sets are defined. For this purpose, instead of considering products whose incremental absorption revenue exceeds a certain threshold, we only consider products whose incremental absorption revenue per unit of weight exceeds a certain threshold.

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

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

where S^* is the optimal solution, W is the capacity bound, and $\alpha \in (0,1)$ is a parameter whose value will be optimized later. Once the product σ_t is selected, we recompute the adjusted prices via the local ratio update described in Figure 3.3. This selection and update process is repeated in every step until either the consideration set becomes empty or adding the current product violates the capacity constraint. Let t' be such a step. In the former case, we stop and return $S_{t'-1}$. In the latter case, we take either $S_{t'-1}$ or $\{\sigma_{t'}\}$, depending on which of these sets has a larger total revenue.

Guessing $R(S^*)$. As in the case of cardinality constraints, since the value of $R(S^*)$ is unknown, we need to approximately guess the value $R(S^*)$. We will use a procedure similar to the one given in Section 3.5, with the exception of utilizing $\frac{1}{|U^*|}R(U^*)$ as a lower bound (see proof of Lemma 3.2 in Appendix B.2), where U^* is the optimal unconstrained solution. In particular, we consider the following guesses for $R(S^*)$.

$$\begin{aligned} B_j &= \frac{1}{|U^*|}R(U^*)(1+\epsilon)^j,\ j = 1, \dots, J \\ J &= \min\left\{j \in \mathbb{N}\ :\ B_j \geq R(U^*)\right\}. \end{aligned} \tag{3.6}$$

Note that $J = O(\frac{1}{\epsilon}\log n)$. Algorithm 9 provides a description of our approximation algorithm for Capa, given a particular guess B_j for $R(S^*)$ and threshold α, while Algorithm 10 describes the complete procedure.

Algorithm 9 Algorithm with guess B_j and threshold α

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

Algorithm 10 Local-ratio Algorithm for MC-Capa with threshold α

1: Given any $\epsilon > 0$, let J and B_j, $j \in [J]$ be as defined in (3.6).
2: For all $j \in [J]$, let S_j be the solution returned by Algorithm 9 with guess B_j and threshold α
3: Return $\arg\max_{j \in [J]} R(S_j)$.

3.6.1 Analysis

To analyze the above algorithm, it is convenient to have a technical lemma similar to Lemma 3.11. By defining the same sets Y_t and Z_t with respect to the optimal assortment S^* to MC-Capa and the adapted consideration sets C_t, the exact same lemma holds. We therefore do not restate this claim and its proof, as these are identical to those of Lemma 3.11. The following theorem shows that the local-ratio algorithm gives a $(1/3-\epsilon)$-approximation for MC-Capa for any fixed $\epsilon > 0$.

Theorem 3.6. *For any fixed $\epsilon > 0$, Algorithm 10 gives a $(1/3-\epsilon/3)$-approximation for MC-Capa. Moreover, the running time is polynomial in the input size and $1/\epsilon$.*

Proof. For a fixed $\epsilon > 0$, let j^* be such that $\frac{R(S^*)}{1+\epsilon} \leq B_{j^*} \leq R(S^*)$. Let $B = B_{j^*}$ and consider the solution returned by Algorithm 9 with guess B and threshold α. We consider two cases based on the condition by which the algorithm terminates. Let t' be the step at which the algorithm terminates.

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

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

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

2. On the other hand, suppose the algorithm terminates since $C_{t'+1} = \emptyset$. Using Lemma 3.11 adapted to the capacitated case, we have

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

Since $C_{t'+1} = \emptyset$, this implies that $Z_{t'+1} = \emptyset$. Moreover, from Lemma 3.9, for all $j = 1, \ldots, t'+1$, we have

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

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

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

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

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

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

Running Time. Algorithm 10 considers $J = O(\frac{1}{\epsilon}\log n)$ guesses of $R(S^*)$. Each run of Algorithm 9 for a given guess is polynomial time. Therefore, the overall running time of Algorithm 10 is polynomial in the input size and $1/\epsilon$. □

Tight example. Our analysis is tight in the following sense. When Algorithm 10 is run with the true value of $R(S^*)$, there are instances for which the approximation ratio is 1/3. For example, consider the instance given in Figure 3.5. For a capacity bound of $W = 1$, the optimal assortment is $S^* = \{b, c\}$. Initially, all the products are in the consideration set and the algorithm picks product a, the highest price product. In the next step, no product can be added to the assortment. The algorithm therefore returns $S = \{a\}$ since $R(\{a\}) > R(\{d\})$ and yields a revenue of $R(S^*)/3+O(\epsilon)$. When ϵ goes to 0, the approximation ratio goes to 1/3.

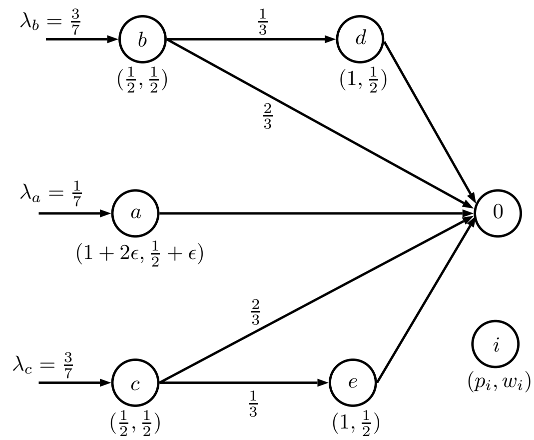


Figure 3.5: A tight example for Algorithm 10.

3.7 Computational experiments

In this section, we present our results from a computational study to test the performance of Algorithm 8 for MC-Card. In particular, we focus on testing: i) the performance of our algorithm with respect to an optimal algorithm, and ii) the running time of this algorithm. We first present a mixed-integer programming (MIP) formulation of MC-Card.

3.7.1 A mixed-integer programming formulation

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

$$
\begin{aligned}
\max \ & \sum_{i=1}^{n} \alpha_i p_i \\
\text{s.t. } & \alpha_i + \beta_i - \sum_{j=1}^{n} \rho_{ji}\beta_j = \lambda_i, && \forall i = 1, \ldots, n \\
& y_i \geq \alpha_i, && \forall i = 1, \ldots, n \\
& \sum_{i=1}^{n} y_i \leq k \\
& \alpha_i \geq 0, \beta_i \geq 0, y_i \in \{0, 1\}, && \forall i = 1, \ldots, n.
\end{aligned}
\tag{3.7}
$$

Lemma 3.12. *The mixed-integer program* (3.7) *is an exact reformulation of* MC-Card.

Proof. Consider the following LP:

$$
\begin{aligned}
\max \ & \sum_{i=1}^{n} \alpha_i p_i \\
\text{s.t. } & \alpha_i + \beta_i - \sum_{j=1}^{n} \rho_{ji}\beta_j = \lambda_i, && \forall i = 1, \ldots, n \\
& \alpha_i \geq 0, \beta_i \geq 0, && \forall i = 1, \ldots, n.
\end{aligned}
\tag{3.8}
$$

Let (α, β) be an extreme point solution to the above LP, and let $S = \{i : \alpha_i > 0\}$. [30] show that α_i is the choice probability $\pi(i, S)$ when the assortment S is offered under the Markov chain choice model. Hence, the objective value $\sum_{i=1}^{n} \alpha_i r_i$ equals to $R(S)$. By adding the indicator variables y_i, we are restricting ourselves to the subset of feasible solutions where at most k of the α_i-s are allowed to be strictly positive. Note that the extreme points of this polytope, corresponding to the projection of the feasible space of the MIP down to the (α, β) coordinates, are exactly the set of assortments S with cardinality at most k. Hence, (3.7) is a mixed-integer formulation of MC-Card. □

3.7.2 Settings tested

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

1. We generate n^2 independent random variables X_{ij}, each picked uniformly over the interval $[0, 1]$. We then set $\rho_{ij} = X_{ij}/\sum_{j=0}^{n} X_{ij}$ for all i, j such that $i \neq j$. Since we do not allow self-loops (i.e. $\rho_{ii} = 0$), the number of random variables needed is n^2. For the arrival rates, we then generate n independent random variables Y_i, each picked uniformly over the interval $[0, 1]$, and set $\lambda_i = Y_i/\sum_{j=1}^{n} Y_j$ for all $i \neq 0$.

2. In this setting, we sparsify the transition matrix of setting 1. More precisely, we additionally generate n^2 independent random variable Z_{ij}, each following a Bernoulli distribution with parameter 0.2. For all i, j such that $i \neq j$, we set $\rho_{ij} = Z_{ij}X_{ij}/\sum_{j=0}^{n} Z_{ij}X_{ij}$, where X_{ij} are generated as in setting 1. This is equivalent to eliminating each transition (i, j) with probability 0.8 and then renormalizing. The arrival rates are generated similarly to setting 1.

3. The transition matrix in this last setting is one of a random walk. More precisely, we generate n^2 independent random variable X_{ij}, each following a Bernoulli distribution with parameter 0.5. We then set $\rho_{ij} = X_{ij}/\sum_{j=0}^{n} X_{ij}$ for all i, j such that $i \neq j$. We also generate n random variables Y_i, each following

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

Table 3.1: Performance of Algorithm 8 for MC-Card.

a Bernoulli distribution with parameter 0.5, and set $\lambda_i = Y_i / \sum_{j=1}^{n} Y_j$ for all $i \neq 0$.

3.7.3 Results

We examine how our algorithm performs in term of both approximation and running time. Table 3.1 shows the approximation ratio of Algorithm 8 (with $\epsilon = 0.1$) for the different settings and the different values of n. As can be observed, the actual performance of our algorithm is significantly better than its worst case theoretical guarantee. Indeed, in all settings tested, the average approximation ratio is always above 0.97. Moreover, the worst approximation ratio over all instances is above 0.77.

The running time of our algorithm also scales nicely. Table 3.2 shows the performance of Algorithm 8 in terms of running time for setting 2. The running times are very similar for the other settings. On the other hand, while the MIP running time can be competitive in some cases, it blows up when the number of products n gets large (see Table 3.2). Note that for $n = 100$, 12 out of the 100 instances had a running time of at least 30 minutes. For $n = 200$, we set a time limit of 2 hours for the MIP. Out of the 20 random instances generated, 16 reached the time limit with-

n	Average Running Time		Maximum Running Time		# instances
	Algorithm 8	MIP	Algorithm 8	MIP	
30	0.18	0.17	0.67	0.25	1,000
60	0.74	0.67	1.25	29.34	1,000
100	3.18	278.20	9.16	10,226.98	100
200	31.98	**	47.38	**	20

Table 3.2: Running time of Algorithm 8 and the MIP for setting 2. ** Denotes the cases when we set a time limit of 2 hours.

out terminating. Therefore, these numerical experiments suggest that Algorithm 8 is computationally efficient and that its numerical performance is significantly better than the theoretical worst-case guarantee. Numerical experiments conducted for Algorithm 10 yield similar observations for MC-Capa.

3.8 Robust assortment optimization

In this section, we consider a robust assortment problem under the Markov chain model. To formulate the robust assortment problem, for any given offer set S, we let $\pi(i, S, \boldsymbol{P})$ be the choice probability that product i is chosen when the assortment S is offered and the transition probabilities are given by the matrix $\boldsymbol{P} = (\rho_{i,j})_{i \in \mathcal{N}, j \in \mathcal{N}_+}$. Let p_i denote the price of product i. For any assortment S, the expected revenue can be written as

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

The uncertainty in the parameters of the Markov chain model is represented by an uncertainty set $\mathcal{P} \in \mathbb{R}^{n \times n}$. In our model, we want to find an assortment S that maximizes the worst-case expected revenue over all model parameters $\boldsymbol{P} \in \mathcal{P}$, corresponding to the optimization problem

$$\max_{S \subseteq \mathcal{N}} \min_{\boldsymbol{P} \in \mathcal{P}} R(S, \boldsymbol{P}). \tag{MC-Rob}$$

We further make the following assumption on the form of the uncertainty set.

Assumption 3.1. *The uncertainty set is a row-wise uncertainty. In particular,*

$$\mathcal{P} = \times_{i=1}^{n} \mathcal{P}_i,$$

where $\mathcal{P}_i \subseteq \{(\rho_{i1}, \ldots, \rho_{in}) \in \mathbb{R}_+^n \mid \rho_{i1} + \ldots + \rho_{in} \leq 1\}$ *is a convex set of uncertain probability transition vectors* $(\rho_{ij})_{j=1}^n$ *out of state* i.

We start by giving some structural properties of the optimal solution. These are in the same spirit than those known for MNL-Rob [67]. We further show that we can adapt our local-ratio framework presented in Section 3.3.3 to give a exact algorithm for MC-Rob. Finally, we provide some comparative statistics and operational insights.

Notation Let S^* be an optimal assortment for MC-Rob. Moreover, for all $\boldsymbol{P} \in \mathcal{P}$, let $S^*(\boldsymbol{P})$ be an optimal assortment when the transition matrix is given by $\boldsymbol{P}$. Similarly, let $\boldsymbol{P}^*(S)$ be the worst case matrix $\boldsymbol{P}$ for assortment S.

3.8.1 Characterization of the optimal assortment

For each product i and set S, we define $R^i(S, \boldsymbol{P})$ as the expected revenue when transitioning out of state i. In particular, for all $i \in [n]$, we have

$$R^i(S, \boldsymbol{P}) = \sum_{j \in S} \rho_{i,j} p_j + \sum_{j \notin S} \rho_{i,j} R^j(S, \boldsymbol{P}).$$

Note that even if $i \in S$, $R^i(S, \boldsymbol{P})$ assumes that we transition out of i. In the following lemma, we characterize when adding a product increases the expected revenue.

Lemma 3.13 (When is adding a product beneficial?)**.** *For any assortment* S *and* $i \notin S$*, the following three statements are equivalent.*

(a) $p_i \geq \min_{\boldsymbol{P} \in \mathcal{P}} R^i(S, \boldsymbol{P})$,

(b) $p_i \geq \min_{\boldsymbol{P} \in \mathcal{P}} R^i(S \cup \{i\}, \boldsymbol{P})$,

(c) $\min_{\boldsymbol{P} \in \mathcal{P}} R(S, \boldsymbol{P}) \leq \min_{\boldsymbol{P} \in \mathcal{P}} R(S \cup \{i\}, \boldsymbol{P})$.

Proof. We first prove that (a) is equivalent to (b) and then that (a) is equivalent to (c).

$(a) \implies (b)$**.** We prove that $p_i < \min_{\boldsymbol{P}\in\mathcal{P}} R^i(S\cup\{i\}, \boldsymbol{P}) \implies p_i < \min_{\boldsymbol{P}\in\mathcal{P}} R^i(S, \boldsymbol{P})$. Let $\boldsymbol{P}^* = \boldsymbol{P}^*(S)$. We have by Markov property,

$$R^i(S \cup \{i\}, \boldsymbol{P}^*) = P^*(i \prec S_+|i)p_i + (1 - P^*(i \prec S_+|i))R^i(S, \boldsymbol{P}^*).$$

Furthermore, we have by assumption and definition of $\boldsymbol{P}^*$,

$$R^i(S \cup \{i\}, \boldsymbol{P}^*) \geq \min_{\boldsymbol{P}\in\mathcal{P}} R^i(S \cup \{i\}, \boldsymbol{P}) > p_i.$$

Combining the first equality with the second inequality yields

$$.(1 - P^*(i \prec S_+|i))R^i(S, \boldsymbol{P}^*) \geq (1 - P^*(i \prec S_+|i))p_i.$$

Moreover, we can assume without loss of generality that $P^*(i \prec S_+|i) < 1$ which implies

$$\min_{\boldsymbol{P}\in\mathcal{P}} R^i(S, \boldsymbol{P}) = R^i(S, \boldsymbol{P}^*) > p_i.$$

$(b) \implies (a)$**.** Similarly, we prove that $p_i < \min_{\boldsymbol{P}\in\mathcal{P}} R^i(S, \boldsymbol{P}) \implies p_i < \min_{\boldsymbol{P}\in\mathcal{P}} R^i(S\cup \{i\}, \boldsymbol{P})$. Let $\boldsymbol{P}^* = \boldsymbol{P}^*(S \cup \{i\})$. We have,

$$\begin{aligned} R^i(S \cup \{i\}, \boldsymbol{P}^*) &= P^*(i \prec S_+|i)p_i + (1 - P^*(i \prec S_+|i))R^i(S, \boldsymbol{P}^*) \\ &\geq P^*(i \prec S_+|i)p_i + (1 - P^*(i \prec S_+|i)) \min_{\boldsymbol{P}\in\mathcal{P}} R^i(S, \boldsymbol{P}) \\ &> p_i. \end{aligned}$$

$(a) \implies (c)$**.** We prove the last two implications using the following fact. For a fixed set S, consider the dynamic problem where for any random walk on the Markov chain, an adversary is allowed to choose a new transition matrix $\boldsymbol{P} \in \mathcal{P}$ before every transition in order to minimize the expected revenue. Note that by the

Markov property, any knowledge of previous past transitions cannot contribute to the decision. Moreover, because of Assumption 3.1, i.e. the row-wise structure of the uncertainty set, the adversary only needs to choose a single row at a time. This implies that there exists a stationary policy, corresponding to a single matrix $\boldsymbol{P} \in \mathcal{P}$ to that problem. In particular, if we let $\hat{R}(S, \boldsymbol{P})$ be the revenue of a problem where we follow $\boldsymbol{P}$ but if we reach i we switch to $\boldsymbol{P}^*(S)$, we have

$$\min_{\boldsymbol{P}\in\mathcal{P}} \hat{R}(S, \boldsymbol{P}) = \min_{\boldsymbol{P}\in\mathcal{P}} R(S, \boldsymbol{P}).$$

However, by assumption, we have for all $\boldsymbol{P} \in \mathcal{P}$

$$\hat{R}(S, \boldsymbol{P}) \leq R(S \cup \{i\}, \boldsymbol{P}).$$

Minimizing both sides with respect to $\boldsymbol{P} \in \mathcal{P}$ yields the desired result.

$(c) \implies (a)$**.** We prove that

$$p_i < \min_{\boldsymbol{P}\in\mathcal{P}} R^i(S, \boldsymbol{P}) \implies \min_{\boldsymbol{P}\in\mathcal{P}} R(S, \boldsymbol{P}) > \min_{\boldsymbol{P}\in\mathcal{P}} R(S \cup \{i\}, \boldsymbol{P}).$$

Let $\hat{R}(S, \boldsymbol{P})$ be the revenue of a problem where we follow $\boldsymbol{P}$ but if we reach i we switch to $\boldsymbol{P}^*(S)$. By the above discussion, we have

$$\min_{\boldsymbol{P}\in\mathcal{P}} \hat{R}(S, \boldsymbol{P}) = \min_{\boldsymbol{P}\in\mathcal{P}} R(S, \boldsymbol{P}).$$

However, by assumption, we have for all $\boldsymbol{P} \in \mathcal{P}$

$$\hat{R}(S, \boldsymbol{P}) > R(S \cup \{i\}, \boldsymbol{P}).$$

Minimizing both sides with respect to $\boldsymbol{P} \in \mathcal{P}$ yields the desired result.

$\square$

Another interesting implication of the proof of Lemma 3.13 is that $\boldsymbol{P}^*(S)$ also minimizes $R^i(S, \boldsymbol{P}^*(S))$ for all i. In other words, the worst case matrix $\boldsymbol{P}$ for a given set S is independent of the arrival rate λ. We will see that the optimal robust assortment is also independent of λ. Using the same proof ideas, we get the following corollary.

Corollary 3.1. *For all set S and $i \notin S$, we have*

$$p_i \geq R^i(S, \boldsymbol{P}) \implies R^j(S \cup \{i\}, \boldsymbol{P}) \geq R^j(S, \boldsymbol{P}), \forall j \neq i.$$

We now provide a structural property of the optimal robust assortment S^*. The result shows that the optimal robust assortment consists of products $i \in [n]$ whose revenues exceed a particular value, which corresponds to the expected revenue the optimal assortment S^* gets when transitioning out of i.

Theorem 3.7 (Characterization of the optimal robust assortment)**.** *There exists an optimal assortment S^* such that*

$$S^* = \left\{ i : p_i \geq R^i(S^*, \boldsymbol{P}^*(S^*)) \right\}.$$

Proof. We first show that $\{i : p_i \geq R^i(S^*, \boldsymbol{P}^*(S^*))\} \subseteq S^*$. Suppose on the contrary that there exists a product i such that $p_i \geq R^i(S^*, \boldsymbol{P}^*(S^*))$ and $i \notin S^*$. By Lemma 3.13, $\min_{\boldsymbol{P} \in \mathcal{P}} R(S^* \cup \{i\}, \boldsymbol{P}) \geq \min_{\boldsymbol{P} \in \mathcal{P}} R(S^*, \boldsymbol{P})$. Therefore, $S^* \cup \{i\}$ is also an optimal assortment.

To complete the proof, we show that $S^* \subseteq \{i : p_i \geq R^i(S^*, \boldsymbol{P}^*(S^*))\}$. Assume on the contrary that there exists a product $i \in S^*$ such that $p_i < R^i(S^*, \boldsymbol{P}^*(S^*))$. By Lemma 3.13, $\min_{\boldsymbol{P} \in \mathcal{P}} R(S^*, \boldsymbol{P}) > \min_{\boldsymbol{P} \in \mathcal{P}} R(S^* \cup \{i\}, \boldsymbol{P})$. This contradicts the optimality of S^* and concludes the proof.

□

Note that when $\mathcal{P}$ is a singleton, this provides an alternative characterization of the optimal assortment when there is no uncertainty, i.e. for MC-Assort. Moreover, in the case of the MNL model, which is a special case of the Markov chain based choice model, $R^i(S, \boldsymbol{P})$ is independent of i and we recover the characterization of the optimal solution given in [67] for the case where there is no uncertainty. However, note that this result does not imply the characterization of the robust solution for MNL obtained in [67]. Indeed, a row-wise uncertainty set for the Markov chain model is not the same as having an uncertainty set of the MNL parameters.

3.8.2 Computing the optimal assortment

For the MNL model, Theorem 3.7 implies that the optimal robust assortment is nested by revenue since the threshold is independent of the product. Therefore, it provides a very efficient way of computing the optimal assortment. Indeed, one only needs to enumerate over the n possible nested assortment and return the revenue maximizing one. In our setting, it is not a priori clear how to compute the optimal assortment using the characterization of Theorem 3.7. However, using the ideas developed in this chapter, in particular, the local ratio framework, we give an efficient algorithm to find S^*. In particular, we give a sequential algorithm which adds a product at every step and finishes with an optimal assortment. Interestingly, the greedy step will be similar to that of Algorithm 6 and we will see how to appropriately modify the update step to accommodate for the parameters uncertainty.

The following lemma allows us to decide which product to add at every step.

Lemma 3.14 (Which product to add next?)**.** *Let S be a given assortment and $\boldsymbol{P} \in \mathcal{P}$ a given transition matrix. Let*

$$i^* = argmax_{i \notin S} \left\{ p_i - R^i(S, \boldsymbol{P}) \right\}.$$

If $p_{i^*} - R^{i^*}(S, \boldsymbol{P}) \geq 0$, then for all $S' \subseteq [n] \backslash S \cup \{i^*\}$, we have

$$\min_{\boldsymbol{P} \in \mathcal{P}} R(S \cup S' \cup \{i^*\}, \boldsymbol{P}) \geq \min_{\boldsymbol{P} \in \mathcal{P}} R(S \cup S', \boldsymbol{P}).$$

Proof. For a given $\boldsymbol{Q} \in \mathcal{P}$ and $S' \subseteq [n] \backslash S \cup \{i^*\}$, we have

$$\begin{aligned} R^{i^*}(S \cup S', \boldsymbol{P}) &= \sum_{j \in S \cup S'} Q(j \prec (S \cup S')_+ | i^*) p_j \\ &= \sum_{j \in S} Q(j \prec (S \cup S')_+ | i^*) p_j + \sum_{j \in S'} Q(j \prec (S \cup S')_+ | i^*) p_j. \end{aligned}$$

By definition of i^*, we have for all $j \in S'$,

$$p_j - \sum_{k \in S} P(k \prec S_+ | j) p_k \leq p_{i^*} - \sum_{k \in S} P(k \prec S + | i^*) p_k,$$

which implies

$$p_j \leq p_{i^*} + \sum_{k \in S} P(k \prec S_+ | j) p_k - \sum_{k \in S} P(k \prec S_+ | i^*) p_k.$$

Therefore,

$$\begin{aligned} R^{i^*}(S \cup S', \boldsymbol{Q}) \leq & \sum_{j \in S} Q(j \prec (S \cup S')_+ | i^*) p_j \\ & + \sum_{j \in S'} Q(j \prec (S \cup S')_+ | i^*) \left(p_{i^*} + \sum_{k \in S} P(k \prec S_+ | j) p_k - \sum_{k \in S} P(k \prec S_+ | i^*) p_k \right) \\ = & \sum_{j \in S'} Q(j \prec (S \cup S')_+ | i^*) \left(p_{i^*} - \sum_{k \in S} P(k \prec S_+ | i^*) p_k \right) \\ & + \sum_{k \in S} p_k \left(Q(k \prec (S \cup S')_+ | i^*) + \sum_{j \in S'} P(k \prec S_+ | j) Q(j \prec (S \cup S')_+ | i^*) \right) \\ \leq & \; p_{i^*} + \sum_{k \in S} (A_k(\boldsymbol{Q}) - P(k \prec S_+ | i^*) p_k, \end{aligned}$$

where for all $\boldsymbol{Q} \in \mathcal{P}$,

$$A_k(\boldsymbol{Q}) = Q(k \prec (S \cup S')_+ | i^*) + \sum_{j \in S'} P(k \prec S_+ | j) Q(j \prec (S \cup S')_+ | i^*).$$

Note that for all $k \in S$, by the Markov property, we have

$$A_k(\boldsymbol{P}) = P(k \prec S_+ | i^*).$$

Therefore, minimizing on both sides of the inequality with respect to $\boldsymbol{Q} \in \mathcal{P}$ yields

$$\min_{\boldsymbol{Q} \in \mathcal{P}} R^{i^*}(S \cup S', \boldsymbol{Q}) \leq p_{i^*}.$$

Using Lemma 3.13 concludes the proof.

□

Motivating by Lemma 3.14, we build a consideration set at every step and add the highest adjusted price product to our current assortment. We stop when the consideration set becomes empty. Algorithm 11 describes the procedure in more detail.

Algorithm 11 Algorithm for MC-Rob

1: Let S be the set of states picked so far, starting with $S = \emptyset$.
2: For all S, let $C(S) = \{i : p_i \geq \min_{\boldsymbol{P} \in \mathcal{P}} R^i(S, \boldsymbol{P})\}$
3: While there exists $i \in C(S)$,
   (a) Let i^* be the product for which $p_i - \min_{\boldsymbol{P} \in \mathcal{P}} R^i(S, \boldsymbol{P})$ is maximized, breaking ties arbitrarily.
   (b) Add i^* to S.
4: Return S.

Theorem 3.8. *Algorithm 11 returns an optimal assortment to* MC-Rob*.*

The correctness of the algorithm follows from inductively using Lemma 3.14. Note that the running time of Algorithm 11 is polynomial in the time needed to compute $\min_{\boldsymbol{P} \in \mathcal{P}} R^i(S, \boldsymbol{P})$ for a given S. Note that this algorithm can be interpreted through our local ratio framework where the greedy rule corresponds to picking the highest adjusted price product and the update step consists of updating all prices according to the following update rule:

$$\hat{p}_i = p_i - \min_{\boldsymbol{P} \in \mathcal{P}} R^i(S, \boldsymbol{P}).$$

Note how this is a robust version of the update presented in Figure 3.3.

Polyhedral uncertainty set. We show how to efficiently compute $\min_{\boldsymbol{P} \in \mathcal{P}} R^i(S, \boldsymbol{P})$ when each $\mathcal{P}_i$ is a polyhedron. For a given assortment S, we can find $\boldsymbol{P}^*$ using the following linear program, where P^i is the i^{th} row of $\boldsymbol{P}$,

$$\begin{aligned} \max \quad & \lambda^T g \\ & g_i = p_i, \forall i \in S \\ & g_i \leq \min_{P^i \in \mathcal{P}_i} (P^i)^T g, \forall i \notin S \\ & g \geq 0. \end{aligned}$$

Taking the dual in the minimization yields for all i

$$\begin{aligned} \min \quad & g^T P^i & = \quad \max \quad & (b^i)^T x \\ & Q^j P^i = b^i & & (Q^i)^T x \leq g \\ & P^i \geq 0 & & \end{aligned}$$

Therefore, the problem is equivalent to solving the following linear program.

$$\begin{aligned} \max \quad & \lambda^T g \\ & g_i = p_i, \forall i \in S \\ & g_i \leq (b^i)^T x, , \forall i \notin S \\ & (Q^i)^T x \leq g, \forall i \notin S \\ & g \geq 0. \end{aligned}$$

Note that when solving the above linear program, we will have for all $i \notin S$, $g_i^* = R^i(S, \boldsymbol{P}^*(S))$.

3.8.3 Comparative statistics and operational insights

We begin by showing that surprisingly there exists a min-max relation for our problem.

Theorem 3.9.

$$\min_{\boldsymbol{P} \in \mathcal{P}} \max_{S} R(S, \boldsymbol{P}) = \max_{S} \min_{\boldsymbol{P} \in \mathcal{P}} R(S, \boldsymbol{P})$$

Proof. Suppose by contradiction that $\max_S \min_{\boldsymbol{P} \in \mathcal{P}} R(S, \boldsymbol{P}) < \min_{\boldsymbol{P} \in \mathcal{P}} \max_S R(S, \boldsymbol{P})$. Our assumption implies that

$$R(S^*, \boldsymbol{P}^*(S^*)) < R(S^*(\boldsymbol{P}^*), \boldsymbol{P}^*(S^*)).$$

By Lemma 3.14, we know that $S^*(\boldsymbol{P}^*) \subseteq S^*$. Therefore, there exists $\{i_1, \ldots, i_K\}$ such that

$$S^* = S^*(\boldsymbol{P}^*) \cup \{i_1, \ldots, i_K\}.$$

By Theorem 3.7, for all $k \in [K]$,

$$p_{i_k} \geq R^{i_k}(S^*(\boldsymbol{P}^*) \cup \{i_1, \ldots, i_K\}, \boldsymbol{P}^*).$$

Therefore, for all $k = 1, \ldots K-1$,

$$p_{i_k} \geq R^{i_k}(S^*(P^*) \cup \{i_1, \ldots, i_{K-1}\}, \boldsymbol{P}^*)$$

by Corollary 3.1. Iterating this procedure, we get that

$$p_{i_1} \geq R^{i_1}(S^*(\boldsymbol{P}^*), \boldsymbol{P}^*)$$

which contradicts the optimality of $S^*(\boldsymbol{P}^*)$.

□

Note that for general uncertainty sets, this relations is not true. We can restate the min-max relation by saying that the optimal robust assortment is also optimal for its worst case matrix, i.e.

$$S^* = S^*(\boldsymbol{P}^*(S^*)).$$

We next prove that the robust assortment corresponds to the largest optimal assortment among $\{S^*(\boldsymbol{P}) : \boldsymbol{P} \in \mathcal{P}\}$.

Lemma 3.15 (Largest optimal assortment is robust)**.**

$$S^* = \bigcup_{\boldsymbol{P} \in \mathcal{P}} S^*(\boldsymbol{P}).$$

Proof. By Theorem 3.9, we have

$$S^* = S^*(\boldsymbol{P}^*(S^*)) \subseteq \bigcup_{\boldsymbol{P} \in \mathcal{P}} S^*(\boldsymbol{P}).$$

By Theorem 3.7, note that for all $\boldsymbol{P} \in \mathcal{P}$,

$$\begin{aligned} S^*(\boldsymbol{P}) &= \{i : p_i \geq R^i(S^*(\boldsymbol{P}), \boldsymbol{P})\} \\ &= \{i : p_i \geq \max_S R^i(S, \boldsymbol{P})\} \\ &\subseteq \{i : p_i \geq \min_{\boldsymbol{P} \in \mathcal{P}} \max_S R^i(S, \boldsymbol{P})\} \\ &\subseteq \{i : p_i \geq \max_S \min_{\boldsymbol{P} \in \mathcal{P}} R^i(S, \boldsymbol{P})\} = S^*. \end{aligned}$$

□

We next analyze how the robust assortment changes with the uncertainty set. It turns out that to protect against larger uncertainty in the model parameters, we should offer a larger assortment. This result is stated in the next corollary, whose proof follows immediately from Theorem 3.15.

Corollary 3.2 (Larger uncertainty implies larger assortment)**.** *For any* $\mathcal{P}_1 \subseteq \mathcal{P}_2$*,*

$$\min_{\boldsymbol{P} \in \mathcal{P}_1} R(S, \boldsymbol{P}) \leq \min_{\boldsymbol{P} \in \mathcal{P}_2} R(S, \boldsymbol{P}) \quad \textit{and} \quad S_1^* \subseteq S_2^*,$$

where S_i^* *is the optimal assortment when the uncertainty set is* $\mathcal{P}_i$*.*

The next corollary shows that as the revenue of each product increases by the same additive increment, the robust optimal assortment becomes larger. For any $\delta \geq 0$, let S_δ^* be the optimal assortment when all the revenues are increased by δ.

Corollary 3.3 (Additive incremental revenues lead to larger robust assortment)**.** *For any* $\delta \geq 0$*,*

$$S^* \subseteq S_\delta^*.$$

Proof. For any assortment S, $\boldsymbol{P} \in \mathcal{P}$ and $i \in S$, we have

$$R^i(S_\delta, \boldsymbol{P}) \leq R(S, \boldsymbol{P}) + \delta.$$

Minimizing on both side with respect to $\boldsymbol{P}$ and then maximizing with respect to S yields

$$\max_S \min_{\boldsymbol{P}\in\mathcal{P}} R^i(S_\delta, \boldsymbol{P}) \leq \max_S \min_{\boldsymbol{P}\in\mathcal{P}} R^i(S_\delta, \boldsymbol{P}) + \delta.$$

Therefore,

$$S^* = \{i : p_i \geq R^i(S^*, \boldsymbol{P}^*(S^*)\} \subseteq \{i : p_i + \delta \geq R^i(S^*_\delta, \boldsymbol{P}^*(S^*_\delta)\} = S^*_\delta.$$

□

3.9 Near optimal algorithm under constant rank

As a consequence of Theorem 2.7, no near-optimal algorithm is possible for MC-Capa in general. In order to get a near-optimal algorithm, we need to make additional assumptions on the structure of the Markov chain. We explore one such assumption in this section. In particular, we assume that the matrix of transition probabilities has a fixed rank K and propose a FPTAS for MC-Capa when the rank K is constant using ideas from Chapter 2. [30] study the network revenue management problem under the Markov chain model and give a linear programming based algorithm. They show that for any assortment S and $i \in S$, the choice probabilities $\pi(i, S)$ can be computed using the following system of linear equations.

$$\begin{aligned} \pi(i, S) &= \lambda_i + \sum_{j \notin S} \beta_j \rho_{ji}, \forall i \in S, \\ \beta_i &= \lambda_i + \sum_{j \notin S} \beta_j \rho_{ji}, \forall i \notin S. \end{aligned} \tag{3.9}$$

We leverage this formulation to give an FPTAS for MC-Capa. In order to leverage the algorithmic ideas of Chapter 2, we express the revenue of any assortment as a function of a small number of linear terms.

3.9.1 Rank one Markov chain

We begin with the case where $K = 1$, i.e. a rank one underlying transition probability matrix. [10] show that special cases of rank one Markov chain models are equivalent to the MNL model or the Generalized Attraction Model (GAM). When $K = 1$, we can without loss of generality assume that there exist $(u_i)_{i\in[n]_+}$ and $(v_i)_{i\in[n]_+}$ such that for all $(i, j) \in [n]_+ \times [n]_+$, $\rho_{ij} = u_i v_j$. The system of equations (3.9) then becomes.

$$\pi(i, S) = \lambda_i + v_i \sum_{j\notin S} \beta_j u_j, \forall i \in S$$
$$\beta_i = \lambda_i + v_i \sum_{j\notin S} \beta_j u_j, \forall i \notin S.$$

Using the set of equations for $i \notin S$, we have

$$\sum_{j\notin S} \beta_j u_j = \frac{\sum_{j\notin S} u_j\lambda_j}{1 - \sum_{j\notin S} u_j v_j} = \left(\sum_{j\notin S} u_j\lambda_j\right)\left(\sum_{m=0}^{\infty}\left(\sum_{j\notin S} u_j v_j\right)^m\right).$$

Consequently, MC-Capa can be reformulated as

$$\max_{S\subseteq[n]}\left\{\sum_{i\in S} p_i\left(\lambda_i + v_i\left(\sum_{j\notin S} u_j\lambda_j\right)\left(\sum_{m=0}^{\infty}\left(\sum_{j\notin S} u_j v_j\right)^m\right)\right) \;\middle|\; \sum_{j\in S} w_j \le W\right\}.$$

This reformulation allows us to use the machinery from Chapter 2. In particular, for given guesses (ℓ, h, g), we try to find, using a dynamic program, a minimum weight assortment assortment S such that

$$\sum_{j\in S} p_i\left(\lambda_i + v_i\frac{h}{1-g}\right) \ge \ell,\ \sum_{j\notin S} u_j\lambda_j \ge h \ \text{ and } \ \sum_{j\notin S} u_j v_j \ge g.$$

Ideas from Algorithm 2 can be adapted straightforwardly to this setting to get an FPTAS.

Theorem 3.10. *When the underlying Markov chain has rank one, MC-Capa admits an FPTAS.*

We give the details of the algorithm and the proof of correctness in Appendix B.6.

3.9.2 Constant rank markov chain

We extend the result to a constant rank Markov chain model in a similar way we extended the FPTAS from MNL to a mixture of MNL. Let K be the rank of the underlying Markov chain. We can write $\rho_{ij} = \sum_{k=1}^{K} u_i^k v_j^k$ for some $(u_i^k)_{i\in[n]+}$ and $(v_i)_{i\in[n]+}$ for all $k \in [K]$. The system of linear equations (3.9) becomes

$$\pi(i,S) = \lambda_i + \sum_{k=1}^{K} v_i^k \sum_{j\notin S} \beta_j u_j^k, \forall i \in S$$
$$\beta_i = \lambda_i + \sum_{k=1}^{K} v_i^k \sum_{j\notin S} \beta_j u_j^k, \forall i \notin S.$$

For all $k \in [K]$, let $L^k = \sum_{j\notin S} \beta_j u_j^k$. We can rewrite for all $i \notin S$,

$$\beta_i = \lambda_i + \sum_{k=1}^{K} v_i^k L^k.$$

For all $k \in [K]$, we therefore get

$$L^k = \sum_{j\notin S} u_j^k \lambda_j + \sum_{m=1}^{K} \left(\sum_{j\notin S} u_j^k v_j^m \right) L^m.$$

Let $Q(S)$ be a $K \times K$ matrix such that for all $(k,m) \in [K]\times[K]$, $Q(S)_{km} = \sum\limits_{j\notin S} u_j^k v_j^m$ and $b(S)$ be a K length vector such that for all k, $b(S)_k = \sum_{j\notin S} u_j^k \lambda_j$. With this notation,

$$L^k = [(I - Q(S))^{-1} b(S)]_k = \left[\left(\sum_{m=0}^{\infty} Q(S)^m \right) b(S) \right]_k.$$

Consequently, we can rewrite MC-Capa as

$$\max_{S\subseteq[n]} \left\{ \sum_{k=1}^{K} \sum_{i\in S} p_i \left(\lambda_i + v_i^k L_k \right) \middle| \sum_{j\in S} w_j \leq W,\ L^k = \left[\left(\sum_{m=0}^{\infty} Q(S)^m \right) b(S) \right]_k, \forall k \right\}$$

Instead of guessing numerators and denominators, we guess the entries of $Q(S)$ and $b(S)$. In particular, for given guesses $\tilde{\boldsymbol{Q}}$ and $\tilde{\boldsymbol{b}}$ as well as a guess $\boldsymbol{\ell}$, we find, using a dynamic program, the minimum weight assortment such that

$$\sum_{i\in S} r_i(\lambda_i + v_i^k L_k) \geq \ell_k,\ \sum_{j\notin S} u_j^k v_j^m \geq \tilde{Q}_{km}, \forall (k,m) \in [K,K], \quad \text{and} \ \sum_{j\notin S} u_j^k \lambda_j \geq \tilde{b}_k, \forall k \in [K],$$

where

$$L_k = \left[\left(\sum_{m=0}^{\infty} \tilde{Q}^m \right) \tilde{b} \right]_k .$$

Note that this guarantees the desired approximation because all the entries of $Q(S)$ and $b(S)$ are non-negative. Consequently, L_k is increasing as a function of any of these entries. This is very similar to the setting for mMNL-Capa and we can adapt Algorithm 15 to get an FPTAS for this problem. Note that the running time is exponential in the rank K.

Theorem 3.11. *There is an FPTAS for MC-Capa when the rank of the underlying Markov chain is constant.*

In this chapter, we have studied a wide variety of assortment problems under the Markov chain model. [10] show that the Markov chain model generalizes the MNL model and approximates a mixture of MNL. We help assess its tractability: it is less tractable than the MNL model, as our hardness results show, but more tractable than a mixture of MNL. In particular, we are able to develop a new algorithmic framework which lead to efficient and practical algorithms for different variant assortment problems. This suggests that the Markov chain model strikes a good balance between expressiveness and tractability.

Chapter 4

Mallows-smoothed distribution over rankings approach for modeling choice

In this chapter, we address the two key computational challenges that arises when using a mixture of Mallows model: (a) efficiently computing the choice probabilities and hence, the expected revenue, for a given offer set S and (b) finding a near-optimal assortment.

In Section 4.2, we present an efficient procedure to compute the choice probabilities $\pi(i, S)$ exactly under a general mixture of Mallows model. We exploit the structural symmetries in the Mallows distribution to derive an efficiently computable *closed-form expression* for the choice probabilities for a given offer set under the mixture of Mallows distribution. In particular, we first consider a single Mallows distribution and show that the choice probabilities under the Mallows distribution can be expressed as a discrete convolution. Using fast Fourier transform, the choice probabilities can be computed in $O(n^2 \cdot \log n)$ time where n is the number of products. Therefore, we obtain a procedure with running time $O(K \cdot n^2 \cdot \log n)$ to compute the choice probabilities for a fixed offer set under a mixture of K Mallows distribution.

In Section 4.3, we present a polynomial time approximation scheme (PTAS) for a large class of constrained assortment optimization for the mixture of Mallows model including cardinality constraints, knapsack constraints, and matroid constraints. Our PTAS holds under the assumption that the no-purchase option is ranked last in the modal rankings for all Mallows segments in the mixture. Under the above assumption and for any $\epsilon > 0$, our algorithm computes an assortment with expected revenue at least $(1 - \epsilon)$ times the optimal in running time that is polynomial in n and K but depends exponentially on $1/\epsilon$. The PTAS is based on establishing a surprising sparsity property about near-optimal assortments, namely, that there exist a near-optimal assortment of size $O(1/\epsilon)$. To the best of our knowledge, this is the first provably near-optimal algorithm for the assortment optimization under Mallows or the mixture of Mallows model in such generality.

In Section 4.4.1, we present a compact mixed integer linear program (MIP) with $O(K \cdot n^3)$ variables, $O(n)$ binary variables and $O(K \cdot n^3)$ constraints for the constrained assortment optimization under a general mixture of Mallows model with K segments. The compact formulation is based on an alternative efficient procedure to compute the choice probabilities for a fixed offer set exactly. In particular, we exploit the repeated insertion method (RIM) introduced by [25] for sampling rankings according to the Mallows distribution and show that the choice probabilities for the Mallows model can be expressed as the unique solution to a system of linear equations that can be solved in $O(n^3)$ time. This gives us an alternate procedure to efficiently compute the choice probabilities for a fixed offer set exactly. While this is less efficient than using fast Fourier transform ($O(n^3)$ versus $O(n^2 \cdot \log n)$), it allows us to formulate a compact MIP for the constrained assortment optimization problem under a general mixture of Mallows model. Our MIP formulation holds for general mixture of Mallows model and does not require any assumption on the rank of no-purchase in the modal rankings.

We conduct numerical experiments to test the computational performance of the MIP. In particular, we implement a variable bound strengthening and observe that the MIP is efficient for reasonably sized assortment optimization problems. Therefore, the MIP provides a practical approach for assortment optimization under a general mixture of Mallows model.

4.1 Model and problem statement

Notation. We consider a universe $\mathcal{N}$ of n products. In order to distinguish products from their corresponding ranks, we let $\mathcal{N} = \{a_1, \ldots, a_n\}$ denote the universe of products, under an arbitrary indexing. Preferences over this universe are captured by an anti-reflexive, anti-symmetric, and transitive relation $\succ$, which induces a total ordering (or ranking) over all products; specifically, $a \succ b$ means that a is preferred to b. We represent preferences through rankings or permutations. A complete ranking (or simply a ranking) is a bijection $\sigma \colon \mathcal{N} \to [n]$ that maps each product $a \in \mathcal{N}$ to its rank $\sigma(a) \in [n]$, where $[j]$ denotes the set $\{1, 2, \ldots, j\}$ for any integer j. Lower ranks indicate higher preference so that $\sigma(a) < \sigma(b)$ if and only if $a \succ_\sigma b$, where $\succ_\sigma$ denotes the preference relation induced by the ranking σ. For simplicity of notation, we also let σ_i denote the product ranked at position i. Thus, $\sigma_1\sigma_2 \cdots \sigma_n$ is the list of the products written by increasing order of their ranks. Finally, for any two integers $i \leq j$, let $[i, j]$ denote the set $\{i, i+1, \ldots, j\}$.

Mallows model. The Mallows model is a member of the distance-based ranking family models (see [58]). This model is described by a modal ranking ω, which denotes the central or modal permutation, and a concentration parameter $\theta \in \mathbb{R}_+$, such that the probability of each permutation σ is given by

$$\lambda(\sigma) = \frac{e^{-\theta \cdot d(\sigma,\omega)}}{\psi(\theta)},$$

where $\psi(\theta) = \sum_\sigma \exp(-\theta \cdot d(\sigma, \omega))$ is the normalization constant, and $d(\cdot, \cdot)$ is the Kendall-Tau metric of distance between permutations defined as

$$d(\sigma, \omega) = \sum_{i<j} \mathbb{1}[(\sigma(a_i) - \sigma(a_j)) \cdot (\omega(a_i) - \omega(a_j)) < 0].$$

In other words, $d(\sigma, \omega)$ counts the number of pairwise disagreements between the permutations σ and ω. It can be verified that $d(\cdot, \cdot)$ is a distance function that is right-invariant under the composition of the symmetric group, i.e., $d(\pi_1, \pi_2) = d(\pi_1\pi, \pi_2\pi)$

for every π, π_1, π_2, where the composition $\sigma\pi$ is defined as $\sigma\pi(a) = \sigma(\pi(a))$. This symmetry can be exploited to show that the normalization constant $\psi(\theta)$ has a closed-form expression [52] given by

$$\psi(\theta) = \prod_{i=1}^{n+1} \frac{1 - e^{-i \cdot \theta}}{1 - e^{-\theta}}.$$

Note that $\psi(\theta)$ depends only on the concentration parameter θ and does not depend on the modal ranking. Intuitively, the Mallows model defines a set of consumers whose preferences are "similar", in the sense of being centered around a common permutation, where the probability for deviations thereof are decreasing exponentially. The similarity of consumer preferences is captured by the Kendall-Tau distance metric.

Mixture of Mallows model. The mixture of K Mallows models is given by K segments where for each segment $k = 1, \ldots, K$, we are given its probability μ_k and the Mallows distribution with modal ranking ω_k and concentration parameter θ_k. Therefore, the probability of any permutation σ in the mixture model is given by

$$\lambda(\sigma) = \sum_{k=1}^{K} \mu_k \cdot \frac{e^{-\theta_k \cdot d(\sigma, \omega_k)}}{\psi(\theta_k)}.$$

4.1.1 Problem statement

Choice probabilities computation. We first focus on efficiently computing the probability that a product a will be chosen from an offer set $S \subseteq \mathcal{N}$ under a given mixture of Mallows model. When offered the subset S, the customer is assumed to sample a preference list according to the mixture of Mallows model and then choose the most preferred product from S according to the sampled list. Therefore, the probability of choosing product a from the offer set S is given by

$$\pi(a_i, S) = \sum_{\sigma} \lambda(\sigma) \cdot \mathbb{1}[\sigma, a_i, S], \tag{4.1}$$

where $\mathbb{1}[\sigma, a_i, S]$ indicates whether $\sigma(a_i) < \sigma(a_j)$ for all $a_j \in S$, $j \neq i$. Note that the above sum runs over $n!$ preference lists, meaning that it is a priori unclear if $\pi(a_i, S)$ can be computed efficiently.

Assortment optimization. Once we are able to compute the choice probabilities, we consider the assortment optimization problem. In the assortment optimization problem, each product a has an endogenously fixed price p_a. Moreover, there is an additional product a_q that represents the outside option (no-purchase), with price $p_q = 0$ that is always included in the assortment. Let $\mathscr{S} \subseteq 2^{\mathcal{N}}$ be denote a set of feasible assortments. We assume that $\mathscr{S}$ satisfied the following assumption:

Assumption 4.1. *Let $\mathscr{S}$ be the set of feasible assortments. We assume that $\mathscr{S}$ satisfies the following properties.*

- **(Membership)** *For any $S \subseteq \mathcal{N}$, it is easy to test whether $S \in \mathscr{S}$ or not.*
- **(Closure)** *For any $S \in \mathscr{S}$ and $T \subseteq S$ implies that $T \in \mathscr{S}$.*

This is a fairly general assumption satisfied for a large class of constraints including cardinality constraints, multi-dimensional knapsack constraints and matroid constraints. The goal in the assortment optimization problem is to determine a feasible subset of products that maximizes the expected revenue (4.2):

$$\max_{S \in \mathscr{S}} \mathcal{R}(S) = \max_{S \in \mathscr{S}} \sum_{a \in S} \pi(a, S \cup \{r_q\}) \cdot p_a. \tag{4.2}$$

We would like to note that even the unconstrained version where $\mathscr{S}$ contains all possible subsets of $\mathcal{N}$ is hard to approximate within a factor better than $O(1/n^{1-\epsilon})$ under a general distribution over permutation model [3].

4.2 Choice probabilities: closed-form expression

In this section, we show that the choice probabilities can be computed efficiently under the Mallows model. Note that this directly give a efficient procedure to compute the choice probabilities under a mixture of Mallows model. Without loss of generality, we assume from this point on that the products are indexed such that the central permutation ω ranks product a_i at position i, for all $i \in [n]$. The next theorem shows that, when the offer set is contiguous, the choice probabilities enjoy a rather simple form. Using these expressions as building blocks, we further derive a closed-form expression for general offer sets.

Theorem 4.1 (Contiguous offer set)**.** *Suppose $S = a_{[i,j]} = \{a_i, \ldots, a_j\}$ for some $1 \leq i \leq j \leq n$. Then, the probability of choosing product $a_k \in S$ under the Mallows model with modal ranking ω and concentration parameter θ is given by*

$$\pi(a_k, S) = \frac{e^{-\theta\cdot(k-i)}}{1 + e^{-\theta} + \cdots + e^{-\theta\cdot(j-i)}}.$$

The choice probability under a general offer set has a more involved structure for which additional notation are needed. For a pair of integers $1 \leq m \leq q \leq n$, define

$$\psi(q, \theta) = \prod_{s=1}^{q} \sum_{\ell=0}^{s-1} e^{-\theta\cdot\ell} \quad \text{and} \quad \psi(q, m, \theta) = \psi(m, \theta) \cdot \psi(q - m, \theta).$$

In addition, for a collection of M discrete functions $h_m \colon \mathbb{Z} \to \mathbb{R}$, $m = 1, \ldots, M$ such that $h_m(r) = 0$ for any $r < 0$, their discrete convolution is defined as

$$\left(h_1 \star \cdots \star h_m\right)(r) = \sum_{\substack{r_1,\ldots,r_M: \\ \sum_m r_m = r}} h_1(r_1) \cdots h_M(r_M).$$

Theorem 4.2 (General offer set)**.** *Suppose $S = a_{[i_1,j_1]} \cup \cdots \cup a_{[i_M,j_M]}$ where $i_m \leq j_m$ for $1 \leq m \leq M$ and $j_m < i_{m+1}$ for $1 \leq m \leq M-1$. Let $G_m = a_{[j_m,i_{m+1}]}$ for $1 \leq m \leq M-1$, $G = G_1 \cup \cdots \cup G_M$, and $C = a_{[i_1,j_M]}$. Then, the probability of choosing $a_k \in a_{[i_\ell,j_\ell]}$ can be written as*

$$\pi(a_k, S) = e^{-\theta\cdot(k-i_1)} \cdot \frac{\prod_{m=1}^{M-1} \psi(|G_m|, \theta)}{\psi(|C|, \theta)} \cdot (f_0 \star \tilde{f}_1 \star \cdots \star \tilde{f}_\ell \star f_{\ell+1} \star \cdots \star f_M)(|G|),$$

where:

- $f_m(r) = e^{-\theta \cdot r \cdot (j_m - i_1 + 1 + r/2)} \cdot \frac{1}{\psi(|G_m|, r, \theta)}$, *if* $0 \leq r \leq |G_m|$, *for* $1 \leq m \leq M$.
- $\tilde{f}_m(r) = e^{\theta \cdot r} \cdot f_m(r)$, *for* $1 \leq m \leq M$.
- $f_0(r) = \psi(|C|, |G| - r, \theta) \cdot \frac{e^{\theta \cdot (|G|-r)^2/2}}{1 + e^{-\theta} + \cdots + e^{-\theta \cdot (|S|-1+r)}}$, *for* $0 \leq r \leq |G|$.
- $f_m(r) = 0$, *for* $0 \leq m \leq M$ *and any* r *outside the ranges described above.*

Proof. At a high level, deriving the expression for a general offer set involves breaking down the probabilistic event of choosing $a_k \in S$ into simpler events for which we can use the expression given in Theorem 4.1, and then combining these expressions using the symmetries of the Mallows distribution.

For a given vector $R = (r_0, \ldots, r_M) \in \mathbb{R}^{M+1}$ such that $r_0 + \ldots r_M = |G|$, let $h(R)$ be the set of permutations which satisfy the following two conditions: i) among all the products of S, a_k is the most preferred, and ii) for all $m \in [M]$, there are exactly r_m products from G_m which are preferred to a_k. We denote this subset of products by $\tilde{G}_m$ for all $m \in [M]$. This implies that there are r_0 products from G which are less preferred than a_k. With this notation,

$$\pi(a_k, S) = \sum_{R: r_0 + \ldots r_M = |G|} \sum_{\sigma \in h(R)} \lambda(\sigma).$$

Recall that for all σ, we have

$$\lambda(\sigma) = \frac{e^{-\theta \cdot \sum_{i,j} \xi(\sigma, i, j)}}{\psi(\theta)},$$

where $\xi(\sigma, i, j) = \mathbb{1}[(\sigma(a_i) - \sigma(a_j)) \cdot (\omega(a_i) - \omega(a_j)) < 0]$. For all σ, we can break down the sum in the exponential as follows:

$$\sum_{i,j} \xi(\sigma, i, j) = C_1(\sigma) + C_2(\sigma) + C_3(\sigma),$$

where

- $C_1(\sigma)$ contains pairs of products (i, j) such that $a_i \in \tilde{G}_m$ for some $m \in [M]$ and $a_j \in S$,
- $C_2(\sigma)$ contains pairs of products (i, j) such that $a_i \in \tilde{G}_m$ for some $m \in [M]$ and $a_j \in G_{m'} \backslash \tilde{G}_{m'}$ for some $m \neq m'$,
- $C_3(\sigma)$ contains the remaining pairs of products.

For a fixed R, we show that $C_1(\sigma)$ and $C_2(\sigma)$ are constant for all $\sigma \in h(R)$.

Part 1. $C_1(\sigma)$ counts the number of disagreements (i.e., number of pairs of products that are oppositely ranked in σ and ω) between some product in S and some product in $\tilde{G}_m$ for any $m \in [M]$. For all $m \in [M]$, a product in $a_i \in \tilde{G}_m$ induces a disagreement with all product $a_j \in S$ such that $j < i$. Therefore, the sum of all these disagreements is equal to,

$$C_1(\sigma) = \sum_{m=1}^{M} \sum_{\substack{a_j \in S \\ a_i \in \tilde{G}_m}} \xi(\sigma, i, j) = \sum_{m=1}^{M} r_m \cdot (j_m - i_1 + 1).$$

Part 2. $C_2(\sigma)$ counts the number of disagreements between some product in any $\tilde{G}_m$ and some product in any $G_{m'} \backslash \tilde{G}_{m'}$ for $m' \neq m$. The sum of all these disagreements is equal to,

$$\begin{aligned} C_2(\sigma) &= \sum_{m \neq m'} \sum_{\substack{a_i \in \tilde{G}_m \\ a_j \in G_{m'} \backslash \tilde{G}_{m'}}} \xi(\sigma, i, j) = \sum_{m=2}^{M} r_m \cdot \sum_{j=1}^{m-1} (|G_j| - r_j) \\ &= \sum_{m=2}^{M} r_m \cdot \sum_{j=1}^{m-1} |G_j| - \sum_{m=2}^{M} r_m \cdot \sum_{j=1}^{m-1} r_j \\ &= \sum_{m=2}^{M} r_m \sum_{j=1}^{m-1} |G_j| - \frac{1}{2}(|G| - m_0)^2 + \frac{1}{2} \sum_{m=1}^{M} r_m^2. \end{aligned}$$

Consequently, for all $\sigma \in h(R)$, we can write $d(\sigma, \omega) = C_1(R) + C_2(R) + C_3(\sigma)$ and therefore,

$$\pi(a_k, S) = \sum_{R: r_0 + \cdots + r_M = |G|} \frac{e^{-\theta \cdot (C_1(R) + C_2(R))}}{\psi(\theta)} \cdot \sum_{\sigma \in h(R)} e^{-\theta . C_3(\sigma)}.$$

Computing the inner sum requires a similar but more involved partitioning of the permutations as well as using Theorem 4.1. The details are presented in Appendix C.1. In particular, we can show that for a fixed R, $\sum_{\sigma\in h(R)} e^{-\theta.C_3(\sigma)}$ is equal to

$$\psi(|G|-m_0,\theta)\cdot\psi(|S|+m_0,\theta)\cdot\frac{e^{-\theta\cdot(k-1-\sum_{m=1}^{\ell-1} r_m)}}{1+\cdots+e^{-\theta\cdot(|S|+m_0-1)}}\cdot\prod_{m=1}^{M}\frac{\psi(|G_m|,\theta)}{\psi(r_m,\theta)\cdot\psi(|G|_m-r_m,\theta)}.$$

Putting all the pieces together yields the desired result. □

Due to representing $\pi(a,S)$ as a discrete convolution, we can efficiently compute this probability using fast Fourier transform in $O(n^2\cdot\log n)$ time (see for instance [21]), which is a dramatic improvement over the exponential sum (4.1) that defines the choice probabilities. Note that for a mixture of K Mallows, this implies that we can compute the choice probability $\pi(a,S)$ in $O(K\cdot n^2\cdot\log n)$ time.

4.3 A PTAS for the assortment optimization

In this section, we present a polynomial time approximation scheme (PTAS) for the assortment optimization problem under the mixture of Mallows model described by (4.2). In other words, for any accuracy level $\epsilon>0$, we compute an assortment with expected revenue at least $(1-\epsilon)$ times the optimal. For every fixed ϵ, the running time is polynomial in n and K. For ease of exposition, we first focus on a single Mallows model and thus drop the index corresponding to the Mallows segment. At the end of the section, we explain how the results extend to a mixture of Mallows model. Before describing the algorithm, we introduce a number of structural properties relative to the Mallows distribution.

4.3.1 Probabilistic claims

We first show that for any pair of products (a_i,a_j) such that $i<j$ (i.e. a_i is preferred to a_j in ω), and for a permutation σ drawn from a Mallows distribution, we have

$\boldsymbol{P}(a_i \succ_\sigma a_j) \geq 1/2$. Note that when $\theta = 0$, since the distribution is uniform, we have $\boldsymbol{P}(a_i \succ_\sigma a_j) = 1/2$. Moreover, when $\theta \to \infty$, $\boldsymbol{P}(a_i \succ_\sigma a_j) = 1$. Our result extends these extreme cases to all values of θ.

Claim 4.1. *For any pair of products (a_i, a_j) such that $i < j$, if σ is drawn from a Mallows distribution, we have,*

$$\boldsymbol{P}(a_i \succ_\sigma a_j) \geq \frac{1}{2}.$$

Proof. Let $A = \{\sigma : a_i \succ_\sigma a_j\}$ and $B = \{\sigma : a_i \prec_\sigma a_j\}$. We consider the bijection $f : A \to B$ which switches a_i and a_j. More precisely, for all $\sigma \in A$,

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

We show that for all $\sigma \in A$, $d(\sigma, \omega) \leq d(f(\sigma), \omega)$ which in turn implies the desired result. Note that for any $\sigma \in A$, we have

$$d(f(\sigma), \omega) - d(\sigma, \omega) = 1 + \sum_{k : a_i \succ_\sigma a_k \succ_\sigma a_j} [\xi(f(\sigma), i, k)) + \xi(f(\sigma), j, k) - \xi(\sigma, i, k)) - \xi(\sigma, j, k)],$$

where $\xi(\sigma, i, j) = \mathbb{1}[(\sigma(a_i) - \sigma(a_j)) \cdot (\omega(a_i) - \omega(a_j)) < 0]$. Since $a_i \succ_\omega a_j$, we have three cases to consider.

Case 1: $a_i \succ_\omega a_k \succ_\omega a_j$. In that case, $\xi(f(\sigma), i, k)) + \xi(f(\sigma), j, k) = 2$ and $\xi(\sigma, i, k)) + \xi(\sigma, j, k) = 0$.

Case 2: $a_k \succ_\omega a_i$. In that case, $\xi(f(\sigma), i, k)) + \xi(f(\sigma), j, k) = 1$ and $\xi(\sigma, i, k)) + \xi(\sigma, j, k) = 1$.

Case 3: $a_j \succ_\omega a_k$. In that case, $\xi(f(\sigma), i, k)) + \xi(f(\sigma), j, k) = 1$ and $\xi(\sigma, i, k)) + \xi(\sigma, j, k) = 1$.

In each case, $\xi(f(\sigma), i, k)) + \xi(f(\sigma), j, k) - \xi(\sigma, i, k)) - \xi(\sigma, j, k) \geq 0$, which concludes the proof.

□

We also extend this result to a tuple of products $(i_1, \ldots, i_m)$. More precisely, if $i_1 < \cdots < i_m$ (i.e. a_{i_1} is the most preferred product of $(a_{i_1}, \ldots, a_{i_m})$ in ω), then if σ is drawn from a Mallows distribution, we have $\boldsymbol{P}(a_{i_1} \succ_\sigma a_{i_j}, \forall j \geq 2) \geq 1/m$. Again, note that when $\theta = 0$, we have $\boldsymbol{P}(a_{i_1} \succ_\sigma a_{i_j}, \forall j \geq 2) = 1/m$. Moreover, when $\theta \to \infty$, $\boldsymbol{P}(a_{i_1} \succ_\sigma a_{i_j}, \forall j \geq 2) = 1$.

Claim 4.2. *For any tuple of products $(a_{i_1}, \ldots, a_{i_m})$ such that $i_1 < \cdots < i_m$, if σ is drawn from a Mallows distribution, we have*

$$\boldsymbol{P}(a_{i_1} \succ_\sigma a_{i_m}, \forall j \geq 2) \geq \frac{1}{m}.$$

Proof. Let $A_k = \{\sigma : a_{i_k} \succ_\sigma a_{i_\ell}, \forall \ell \neq k\}$ be the set of permutations in which a_{i_k} appears first among $a_{i_1}, \ldots, a_{i_m}$. For a fixed pair (k, m) such that $k < m$, consider the bijection $f : A_m \to A_k$ which switches m and k. More precisely, for all $\sigma \in A_m$,

$$f(\sigma)(\ell) = \begin{cases} \sigma(a_k) & \text{if } \ell = m \\ \sigma(a_m) & \text{if } \ell = k \\ \sigma(a_\ell) & \text{otherwise} \end{cases}.$$

The proof of Claim 4.1 shows that for all $\sigma \in A_m$, $d(\sigma, \omega) \leq d(f(\sigma), \omega)$. This in turn implies that for all $k < m$,

$$\boldsymbol{P}(a_{i_k} \succ_\sigma a_{i_\ell}, \forall \ell \neq k) \geq \boldsymbol{P}(a_{i_m} \succ_\sigma a_{i_\ell}, \forall \ell \neq m),$$

and concludes the proof. □

4.3.2 A PTAS for the assortment optimization problem

We now present a polynomial time approximation scheme (PTAS) for the assortment optimization problem under the Mallows distribution under an additional assumption. Our algorithm is based on establishing a surprising sparsity property, proving the existence of small-sized near-optimal assortments, crucially utilizing certain symmetries in the distribution over permutations.

Description of the algorithm. Let S^* be the optimal assortment. Let $M = 1/\epsilon$, where without loss of generality, assume that M takes an integer value. We enumerate all possible subsets of $\mathscr{S}$ of size less or equal than M and return the best candidate assortment. Algorithm 12 describes the procedure.

Algorithm 12 Computing choice probabilities

1: Let $\hat{S} = \arg\max_{S \in \mathscr{S}} \{\mathcal{R}(S) : |S| \leq M\}$.
2: Return $\hat{S}$.

Assumption 4.2. *The outside option is ranked last in the central permutation* ω, *i.e.* $q = n$.

Theorem 4.3. *Under Assumption 4.2, Algorithm 12 is a PTAS for the assortment optimization problem* (4.2) *under the Mallows distribution.*

Proof. We first argue the correctness of the algorithm, i.e., that the assortment returned is a $(1-\epsilon)$-optimal solution. Again, let S^* be the optimal assortment. Note that if $|S^*| < M$, then S^* is one of the candidate assortments we examine, and therefore the algorithm returns the optimal solution. We therefore assume that $|S^*| \geq M$. In this case, let S_ϵ consists of the M highest revenue product of S^*. Note that S_ϵ is among the candidate assortment constructed by the algorithm. Moreover, it is a feasible assortment by Assumption 4.1. We show that S_ϵ is $(1-\epsilon)$-optimal using a sample-path analysis. In particular, let σ be a fixed preference list. Let $\mathcal{R}(\sigma, S)$ be the revenue obtained by σ when assortment S is offered. In particular,

$$\mathcal{R}(\sigma, S) = \sum_{a \in S} \mathbb{1}[\sigma, a, S] \cdot r_a.$$

We consider two cases.

Case 1. We first assume that $a_q \succ_\sigma a_i$ for all $i \in S_\epsilon$. In this case, $\mathcal{R}(\sigma, S_\epsilon) = 0$. On the other hand, offering a single product a_i is always a feasible solution. Therefore, by Claim 4.1,

$$\mathcal{R}(S^*) \geq \mathcal{R}(\{i^*\}) = \boldsymbol{P}(a_{i^*} \succ a_q) \cdot r_{i^*} \geq \frac{r_{i^*}}{2},$$

where a_{i^*} is the highest price product. Moreover, note that all product in $S^*\backslash S_\epsilon$ have revenue smaller or equal to r_{i^*}. Therefore, any product in $S^*\backslash S_\epsilon$ has revenue less or equal than $2 \cdot \mathcal{R}(S^*)$. This implies that $\mathcal{R}(\sigma, S^*) \leq 2 \cdot \mathcal{R}(S^*)$.

Case 2. In this case, we assume that in the permutation σ, there exists a product $a_i \in S_\epsilon$ such that $a_i \succ a_q$ and $\mathcal{R}(\sigma, S_\epsilon) = r_{a_i}$. We show that $\mathcal{R}(\sigma, S_\epsilon) \geq \mathcal{R}(\sigma, S^*)$. Indeed, suppose that there exists a product a_j is in $S^*\backslash S_\epsilon$ such that $a_j \succ_\sigma a_i$. Since S_ϵ contains the M highest revenue product of S^*, it must be that $r_j \leq r_j$. Therefore, $\mathcal{R}(\sigma, S_\epsilon) \geq \mathcal{R}(L, S^*)$.

We now combine the two cases. For case 1 to happen, note that a_q has to be preferred to all products from S_ϵ. From Claim 4.2, this event occurs with probability at most $1/|S_\epsilon| = 1/M = \epsilon$. Consequently,

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

From a running time perspective, the number of candidate assortment is equal to $n^{1/\epsilon}$. By Theorem 4.2, we can compute $\mathcal{R}(S)$ for any assortment S in $O(n^3 \cdot \log(n))$. Therefore the overall running time of the algorithm is $O(n^{1/\epsilon} \cdot n^3 \cdot \log(n))$. □

Extension to a mixture of Mallows model. The PTAS extends to a mixture of Mallows model as long as Assumption 4.2 holds for each segment. Indeed, since the probabilistic claims hold for each segment, we can adapt the proof of Theorem 4.3 to the case of a mixture of Mallows model. Moreover, we would like to emphasize that the running time scales linearly in K, the number of segments of the Mallows model. More precisely, Algorithm 12 returns a $(1-\epsilon)$-optimal assortment in $O(K \cdot n^{1/\epsilon} \cdot n^3 \cdot \log(n))$.

The PTAS presented in this section provides an approximation algorithm with provable guarantees for a special case of a mixture of Mallows model. However, it requires Assumption 4.2 to hold. We next present an alternative way of solving the assortment optimization problem using a MIP formulation which does not require any assumption. To this end, we first present an alternative method for computing the choice probabilities by means of dynamic programming.

4.4 Integer programming formulation

While Section 4.2 allows computing the choice probabilities efficiently, the approach does not lend itself to solving the assortment optimization problem. For that reason, we present an alternative algorithm for computing the choice probabilities which will then lead to a MIP formulation for the unconstrained assortment optimization problem.

4.4.1 Choice probabilities: a dynamic programming approach

In what follows, we present an alternative algorithm for computing the choice probabilities under a Mallows model. Again, note that this directly implies an efficient algorithm for computing the choice probabilities under a mixture of Mallows model. Our approach is based on an efficient procedure to sample a random permutation according to a Mallows model with modal ranking ω and concentration parameter θ. The random permutation is constructed sequentially, as explained in Algorithm 13.

Lemma 4.1 (Theorem 3 in [49]). *The repeated insertion procedure generates a random sample from a Mallows distribution with modal ranking ω and concentration*

Algorithm 13 Repeated insertion procedure

1: Let $\sigma = \{a_1\}$.
2: For $i = 2, \dots, n$, insert a_i into σ at position $s = 1, \dots, i$ with probability

$$\alpha_{i,s} = \frac{e^{-\theta \cdot (i-s)}}{1 + e^{-\theta} + \cdots + e^{-\theta \cdot (i-1)}}.$$

3: Return σ.

parameter θ.

Based on the correctness of this procedure, we describe a dynamic program to compute the choice probabilities of a general offer set S. The key idea is to decompose these probabilities to include the position at which a product is chosen. In particular, for $i \leq m$ and $s \in [m]$, let $\pi(i, s, m)$ be the probability that product a_i is chosen (i.e., appears first among products in S) at position s after the m-th step of Algorithm 13. In other words, $\pi(i, s, m)$ corresponds to a choice probability when restricting $\mathcal{N}$ to the first m products, $a_1, \dots, a_m$. With this notation, we have for all $i \in [n]$,

$$\pi(a_i, S) = \sum_{s=1}^{n} \pi(i, s, n).$$

We compute $\pi(i, s, m)$ iteratively for $m = 1, \dots, n$. In particular, in order to compute $\pi(i, s, m+1)$, we use the correctness of the sampling procedure. Specifically, starting from a permutation σ that includes the products $a_1, \dots, a_m$, the product a_{m+1} is inserted at position j with probability $\alpha_{m+1,j}$, and we have two cases to consider.

Case 1: $a_{m+1} \notin S$. In this case, $\pi(m+1, s, m+1) = 0$ for all $s = 1, \dots, m+1$. Consider a product a_i for $i \leq m$. In order for a_i to be chosen at position s after a_{m+1} is inserted, one of the following events has to occur:

i) a_i was already chosen at position s before a_{m+1} is inserted, and a_{m+1} is inserted at a position $\ell > s$,

ii) a_i was chosen at position $s - 1$, and a_{m+1} is inserted at a position $\ell \leq s - 1$.

Consequently, we have for all $i \leq m$,

$$\pi(i, s, m+1) = \sum_{\ell=s+1}^{m+1} \alpha_{m+1,\ell} \cdot \pi(i, s, m) + \sum_{\ell=1}^{s-1} \alpha_{m+1,\ell} \cdot \pi(i, s-1, k)$$
$$= (1 - \gamma_{m+1,s}) \cdot \pi(i, s, m) + \gamma_{m+1,s-1} \cdot \pi(i, s-1, m),$$

where $\gamma_{m,s} = \sum_{\ell=1}^{s} \alpha_{m,\ell}$ for all m, s.

Case 2: $a_{m+1} \in S$. Consider a product a_i with $i \leq m$. This product is chosen at position s only if it was already chosen at position s and a_{m+1} is inserted at a position $\ell > s$. Therefore, for all $i \leq m$, $\pi(i, s, m+1) = (1 - \gamma_{m+1,s}) \cdot \pi(i, s, m)$. For product a_{m+1}, it is chosen at position s only if all products a_i for $i \leq m$ are at positions $\ell \geq s$ and a_{m+1} is inserted at position s, implying that

$$\pi(m+1, s, m+1) = \alpha_{m+1,s} \cdot \sum_{i \leq m} \sum_{\ell=s}^{n} \pi(i, \ell, m).$$

Algorithm 14 summarizes this procedure.

Algorithm 14 Computing choice probabilities

1: Let S be a general offer set. Without loss of generality, we assume that $a_1 \in S$.
2: Let $\pi(1, 1, 1) = 1$.
3: For $m = 1, \ldots, n-1$,
   (a) For all $i \leq m$ and $s = 1, \ldots m+1$, let

$$\pi(i, s, m+1) = (1 - \gamma_{m+1,s}) \cdot \pi(i, s, m) + \mathbb{1}[a_{m+1} \notin S] \cdot \gamma_{m+1,s-1} \cdot \pi(i, s-1, m).$$

   (b) For $s = 1, \ldots, m+1$, let

$$\pi(m+1, s, m+1) = \mathbb{1}[a_{m+1} \in S] \cdot \alpha_{m+1,s} \cdot \sum_{i \leq m} \sum_{\ell=s}^{n} \pi(i, \ell, m).$$

4: For all $i \in [n]$, return $\pi(a_i, S) = \sum_{s=1}^{n} \pi(i, s, n)$.

Theorem 4.4. *For any offer set S, Algorithm 14 returns the choice probabilities under a Mallows distribution with modal ranking ω and concentration parameter θ.*

This dynamic programming approach provides an $O(n^3)$ time algorithm for computing $\pi(a, S)$ for all products $a \in S$ simultaneously. Moreover, as explained in the next section, these ideas lead to an algorithm to solve the assortment optimization problem.

4.4.2 Assortment optimization: integer programming formulation

Building on Algorithm 14 and introducing a binary variable for each product, we can reformulate the assortment optimization problem (4.2) under a mixture of Mallows model as a mixed inter program (MIP). Although the MIP formulation does not enjoy the theoretical guarantees of the PTAS (i.e. upper bound on the running time), it does not require Assumption 4.2 to hold. Again, we start with a single Mallows model. In particular, we give a MIP with only $O(n^3)$ variables and constraints, with only n 0-1 variables. We assume for simplicity that the first product of S (say a_1) is known. Since this product is generally not known a-priori, in order to obtain an optimal solution to problem (4.2), we need to guess the first offered product and solve the above integer program for each of the $O(n)$ guesses. We would like to note that the MIP formulation is presented for the unconstrained assortment optimization problem but is quite powerful and can handle a large class of constraints on the assortment (such as cardinality and capacity constraints).

Theorem 4.5. *Conditional on $a_1 \in S$, the following mixed integer program (MIP) computes an optimal solution to the unconstrained assortment optimization problem* *under a Mallows model:*

$$
\begin{aligned}
\max \quad & \sum_{i,s} p_i \cdot \pi(i,s,n) & \\
\text{s.t.} \quad & \pi(1,1,1) = 1, \pi(1,s,1) = 0, & \forall s = 2, \ldots, n \\
& \pi(i,s,m+1) = (1 - w_{m+1,s}) \cdot \pi(i,s,m) + y_{i,s,m+1}, & \forall i, s, \forall m \geq 2 \\
& \pi(m+1,s,m+1) = z_{s,m+1}, & \forall s, \forall m \geq 2 \\
& y_{i,s,m} \leq \gamma_{m+1,s-1} \cdot \pi(i,s-1,m-1), & \forall i, s, \forall m \geq 2 \\
& 0 \leq y_{i,s,m} \leq \gamma_{m+1,s-1} \cdot (1 - x_m), & \forall i, s, \forall m \geq 2 \\
& z_{s,m} \leq \alpha_{m+1,s} \cdot \sum_{\ell=s}^{n} \sum_{i=1}^{m-1} \pi(i,\ell,m-1), & \forall s, \forall m \geq 2 \\
& 0 \leq z_{s,m} \leq \alpha_{m+1,s} \cdot x_m, & \forall s, \forall m \geq 2 \\
& x_1 = 1, x_q = 1, x_m \in \{0,1\} &
\end{aligned}
$$

Proof. Let $x = (x_1, \ldots, x_n)$ be a feasible binary vector to the MIP and let $S = \{a_i : x_i = 1\}$. Note that there is a one to one correspondence between feasible vector x to the MIP and feasible assortment S such that $a_1 \in S$ and $a_q \in S$. Consequently, we can rewrite the MIP as

$$
\begin{aligned}
\max_{\substack{S \subseteq \mathscr{U} \\ a_q \in S}} \max \quad & \sum_{i,s} p_i \cdot \pi(i,s,n) & \\
\text{s.t.} \quad & \pi(i,s,m+1) = (1 - w_{m+1,s}) \cdot \pi(i,s,m) + y_{i,s,m+1}, & \forall i, s, \forall m \geq 2 \\
& \pi(m+1,s,m+1) = z_{s,m+1}, & \forall s, \forall m \geq 2 \\
& 0 \leq y_{i,s,m} \leq \mathbb{1}[a_{m+1} \notin S] \cdot \gamma_{m+1,s-1} \cdot \pi(i,s-1,m-1), & \forall i, s, \forall m \geq 2 \\
& 0 \leq z_{s,m} \leq \mathbb{1}[a_{m+1} \in S] \cdot \alpha_{m+1,s} \cdot \sum_{\ell=s}^{n} \sum_{i=1}^{m-1} \pi(i,\ell,m-1), & \forall s, \forall m \geq 2 \\
& \pi(1,1,1) = 1 &
\end{aligned}
$$

Note that it is always optimal to set $y_{i,s,m}$ and $z_{s,m}$ at their upper bound because all the coefficients in the objective function are non-negative. The correctness of Algorithm 14 then shows that the MIP is an equivalent formulation of the unconstrained assortment optimization problem under a Mallows model. □

We now present the MIP formulation for the unconstrained assortment optimization problem for a mixture of Mallows model. Again, we want to emphasize that the binary variables allow capturing a wide variety of constraints.

Theorem 4.6. *Conditional on $a_1 \in S$, the following mixed integer program (MIP) computes an optimal solution to the unconstrained assortment optimization problem under a mixture of Mallows model:*

$$
\begin{aligned}
\max \quad & \sum_{i,s,k} p_i \cdot \mu_k \cdot \pi^k(i,s,n) & \\
\text{s.t.} \quad & \pi^k(1,1,1) = 1, \pi^k(1,s,1) = 0, & \forall s = 2,\ldots,n, \forall k \\
& \pi^k(i,s,m+1) = (1 - w_{m+1,s}) \cdot \pi^k(i,s,m) + y^k_{i,s,m+1}, & \forall i,s,k, \forall m \geq 2 \\
& \pi^k(m+1,s,m+1) = z^k_{s,m+1}, & \forall s,k, \forall m \geq 2 \\
& y^k_{i,s,m} \leq \gamma_{m+1,s-1} \cdot \pi^k(i,s-1,m-1), & \forall i,s,k, \forall m \geq 2 \\
& 0 \leq y^k_{i,s,m} \leq \gamma_{m+1,s-1} \cdot (1 - x_{\omega_k(a_m)}), & \forall i,s,k, \forall m \geq 2 \\
& z^k_{s,m} \leq \alpha_{m+1,s} \cdot \sum_{\ell=s}^{n} \sum_{i=1}^{m-1} \pi^k(i,\ell,m-1), & \forall s,k, \forall m \geq 2 \\
& 0 \leq z^k_{s,m} \leq \alpha_{m+1,s} \cdot x_{\omega_k(a_m)}, & \forall s,k, \forall m \geq 2 \\
& x_1 = 1, x_q = 1, x_m \in \{0,1\} &
\end{aligned}
$$

4.5 Numerical experiments

In this section, we examine the numerical performance of the MIP. We consider the following simulation setup for a single Mallows model. Product prices are sampled independently and uniformly at random from the interval [0, 1]. The modal ranking is fixed to the identity ranking with the outside option ranked at the top. The outside option being ranked at the top is characteristic of applications in which the retailer captures a small fraction of the market and the outside option represents the (much larger) rest of the market. Indeed, most of the customers visiting a website or a store leave without making a purchase. Because the outside option is always offered, we need to solve only a single instance of the MIP (described in Theorem 4.5). Note that in the more general setting, the number of MIPs that must be solved is equal the minimum of the rank of the outside option and the rank of the highest revenue item. Because the MIPs are independent of each other, they can be solved in parallel. We solved the MIPs using the Gurobi Optimizer version 6.0.0 on a computer with processor 2.4GHz Intel Core i5, RAM of 8GB, and operating system Mac OSX El Capitan. In order to improve the running time of the MIP, we first strengthen the big-M constraints. We describe this strengthening below.

Strengthening of the MIP formulation. We use some structural properties of the optimal solution to tighten some of the upper bounds involving the binary variables in the MIP formulation. In particular, for all i, s, and m, we replace the constraint

$$y_{i,s,m} \leq \gamma_{m+1,s-1} \cdot (1 - x_m),$$

by the following constraint

$$y_{i,s,m} \leq \gamma_{m+1,s-1} \cdot u_{i,s,m} \cdot (1 - x_m),$$

where $u_{i,s,m}$ is the probabliity that product a_i is selected at position $(s-1)$ after the m^{th} step of Algorithm 13 when the offer set is $S = \{a_{i^*}, a_q\}$, i.e. when only the highest priced product is offered. Since we know that the highest price product is always offered in the optimal assortment, this is a valid upper bound to $\pi(i, s-1, m-1)$ and therefore a valid strenghtening of the constraint. Similarly, for all s and m, we replace the constraint,

$$z_{s,m} \leq \alpha_{m+1,s} \cdot x_m,$$

by the following constraint

$$z_{s,m} \leq \alpha_{m+1,s} \cdot v_{s,m} \cdot x_m,$$

where $v_{s,m}$ is equal to the probability that product that product i is selected at position $\ell = s, \ldots, n$ when the offer set is $S = \{a_q\}$ if $a_i \succ_w a_{i^*}$, and $S = \{a_q, a_{i^*}\}$ otherwise. Again using the fact that the highest price product is always offered in the optimal assortment, we can show that this is a valid upper bound.

Results and discussion. Table 4.1 shows the running time of strengthened MIP formulation for different values of $e^{-\theta}$ and n. For each pair of parameters, we generated 50 different instances.

n	$e^{-\theta}$	Without strengthening		With strengthening	
		Average (s)	Max (s)	Average (s)	Max (s)
10	0.8	4.60	5.64	4.65	7.17
10	0.9	4.72	5.80	4.58	5.73
15	0.8	19.04	27.08	17.4	18.73
15	0.9	21.30	28.79	19.67	23.61
20	0.8	65.43	87.48	48.08	58.09
20	0.9	222.19	626.08	105.30	189.93
25	0.8	**	**	143.21	183.78
25	0.9	**	**	769.78	1,817.98

Table 4.1: Running time of the strengthened MIP for various values of $e^{-\theta}$ and n. (**the solver did not terminate in 8 hours)

We would like to note that the strengthening improves the running time considerably. Under the initial formulation, the MIP did not terminate after several hours for $n = 25$ whereas it was able to terminate in a few minutes with the additional strengthening. Our MIP obtains the optimal solution in a reasonable amount of time for the considered parameter values. Outside of this range, i.e. when $e^{-\theta}$ is too small or when n is too large, there are potential numerical instabilities. The strengthening we propose is one way to improve the running time of the MIP but other numerical optimization techniques may be applied to improve the running time even further. Finally, we emphasize that the MIP formulation is necessary because of its flexibility to handle versatile business constraints (such as cardinality or capacity constraints) that naturally arise in practice.

In this chapter, we have studied the mixture of Mallows model. Despite being a distribution over rankings whose support is exponential, we show that this distribution is still very tractable. Using the symmetries of the Mallows distribution, we develop efficient procedures to compute choice probabilities and give tractable approaches to the assortment optimization problem. Therefore, smoothing a sparse distribution increases its predictive power and interpretability without affecting its tractability.

Chapter 5

Design of Futures Contract for Risk-averse Online Advertisers

5.1 Introduction

Why does *advance selling* (i.e., buyers purchasing items before they actually become available for use) occur? There are several explanations for this widespread phenomenon, some of which have been well studied in the academic literature. One reason is that sellers can benefit by more accurately forecasting demand, reducing the risk of either insufficient inventory or overproduction and wastage, thus managing their production costs and supply chains more effectively. To incentivize buyers to provide such forecasts, sellers often offer *discounts* for *pre-ordering*, commonly seen in the publishing and manufacturing industries. Another reason is that it allows sellers to segment the market by using a price discrimination strategy, which is typical in the travel and tourism industries: For example, leisure travelers with more flexibility get a lower price for booking flights or hotel rooms far in advance, while relatively price-insensitive business travelers pay a higher price for urgent last-minute bookings. In both these cases, as it has been shown in the Marketing literature, advance selling induces a discounted price for the consumers that are willing to commit and buy in advance (see, e.g., [69] and [28]).

A slightly different setting is that of futures contracts in finance, which emerged in the fifties. These contracts allow sellers and buyers to agree upon a price of a commodity (e.g., oil) that will be delivered at a specified future date. Here, the contract exists to protect both buyers and sellers from uncertainty in future prices (often due to unpredictable factors such as market fluctuations, weather, or supply-demand mismatches). Since both parties benefit from advance selling, the price is not necessarily discounted; it usually depends on risk factors that leads to a no-arbitrage pricing policy (see, e.g., [41] and [38]).

In this chapter, we study a different case for advance selling, that does not appear to have been considered before in the literature in this form: In a supply-constrained world, particularly with variable demand, buyers may face significant uncertainty in both pricing and allocation. To reduce this uncertainty, risk-averse buyers may be willing to pay a *premium* for an advance purchase that guarantees they will both (a) receive the item being sold and (b) pay a fixed price. That is, a risk-averse buyer can hedge against the possibility of a 'stock out' (i.e., not receiving the item) due to high demand, and against the possibility of a high price that prevents the buyer from spending her budget in a controlled manner.[1] Our motivating application throughout this chapter will be that of Internet display advertising, which inspired this work, but these ideas apply to other settings, such as pricing cloud computing services.

One of the main goals of this chapter is to propose a model for advance selling for Internet advertising, and to show that it is beneficial for both sellers (publishers of Internet content, who sell advertising space) and buyers (advertisers purchasing the right to display their ad adjacent to the content). Traditionally, display advertising is sold in two ways: First, through *reservation contracts* sold in advance, where an advertiser enters into an agreement with a publisher, paying a fixed price for its ads to be shown to a specified volume of visitors to the publisher's website, perhaps satisfying certain additional criteria. For example, Nike may pay $50,000 to have its ads shown to 5 million espn.com website users who are based in the US and frequently

[1] The idea of buyers paying premiums for higher *levels / quality* of service has been studied in the literature, but as we shall see, our work differs from this in important ways.

visit the basketball section of the website. Reservation contracts have guaranteed spend from the advertiser, and guaranteed number of impressions (an impression or ad view occurs when one ad is shown to a user) by the publisher. Second, display advertising may be sold through *real-time bidding*, in which advertisers and publishers meet through an exchange platform (such as Google's DoubleClick Ad Exchange). When a user visits the publisher's website, the exchange may request real-time bids from multiple advertisers and run an auction, awarding the ad slot to the highest bidder. This is usually a second-price auction for each individual impression, with no guarantees to either publisher or advertiser; in some cases, instead of an auction, the seller posts a fixed price.

Reservations and Market Maker: Typically, costs per impression are several times higher for reservation contracts than for auction purchases, even though auctions allow advertisers more fine-grained tracking and targeting of their ads to individual users. Why is this the case? One of the main insights of this chapter is an explanation for this difference; we model buyers as risk-averse rational agents that follow commonly used utility models, and show that they are willing to pay a premium for the guaranteed impression volume and guaranteed prices offered by reservations, providing higher revenue to publishers.[2] Recent independent work [37] in the context of pricing for Cloud Computing has also posited risk aversion as the reason for the existence of a guaranteed option at a higher price relative to the expected clearing price of an auction; see our discussion of related work. Further, we go beyond this qualitative insight, mathematically characterizing the appropriate premiums. In particular, we propose a new type of contract, referred to as *Market-Maker* contracts,

[2]One might note that there is a large and ever-increasing supply of Internet content; in such a supply-rich world, why should advertisers pay reservation premiums, since they can always buy ads on other websites? In line with our reasoning, the large mass of small publishers barely sell reservation contracts, instead using auction-based platforms like Google's AdSense. However, there is a limited supply of high-quality content; large publishers with such differentiated content and audiences (such as The New York Times, The Economist, or YouTube) are indeed the most likely to sell reservation contracts and charge high premiums.

that can *replace* or complement reservations: For risk-averse buyers concerned about having to pay an unpredictable high price, or possibly not receiving impressions at all, the Market-Maker is a system that quotes a price higher than the expected price of an impression. Buyers can choose whether to pay this higher price or take their chances in the open auction. The Market-Maker contract guarantees (like reservation contracts) that buyers who purchase it will receive their impressions. It is then the responsibility of the Market-Maker to purchase these impressions on behalf of its buyers, even if it has to pay a price higher than it charged the buyers.

Advantages of Market-Maker: We first claim that compared to a world with no contracts or guaranteed sales, the addition of the Market-Maker benefits *both* buyers and sellers:

- Since the risk-averse buyers obtain guarantees, they may derive higher utility even though they pay the Market-Maker a higher price.
- This higher price obtained from buyers can be passed on to the seller as additional revenue (after deducting a share for the Market-Maker's assumption of risk).

In settings that currently offer both contracts and auctions, such as Internet display advertising, the Market-Maker can replace existing forms of contracts, by *automating* reservations. There are multiple advantages to Market-Maker as an alternative to reservations:

- The current sales process for reservations involves considerable manual effort and long-term human negotiations; some publishers report up to $10,000 in costs to service a single guaranteed reservation advertising campaign [61], which can be a sizable fraction of the total campaign spend. Offering publishers an automated option that provides a reservation-like premium can allow them to obtain more revenue at lower costs.

- Currently, advertisers can only practically sign contracts with a small number of publishers, and vice versa. An automated system like the Market-Maker can scale better to a larger number of buyers and sellers, reducing search and transaction costs for such contracts, and extend to more complex contracts with finer targeting.

We next describe our model and contributions in more detail.

5.1.1 Contributions

At a high level, we make the following contributions:

- *Framework to study guarantee-based premiums.* We introduce a framework for mathematically analyzing the benefits of guarantees for risk-averse Internet advertisers. We consider the addition of guaranteed-delivery sales to the two predominant existing modes of real-time-bidding based sales. The first is fixed-price deals (commonly called *preferred deals*, a part of *private marketplaces*), in which the seller invites buyers to participate at a posted price determined by the seller. The other is the classic second-price auction setting. We study the benefits of adding a Market-Maker purchase option which provides buyer guarantees in each of these settings.

- *Equilibrium buyer behavior analysis.* The introduction of a new option for buyers changes the equilibrium that would exist without this option. We assume that buyers choose their preferred option to maximize their utilities, and study several common utility models. We use an envy-based utility model in the fixed-price / preferred deals setting and two widely used utility models, namely CARA and Standard deviation models, for the auction setting. For each of these settings, and each of these utility models, we characterize how to set the

Market-Maker price, and analyze the equilibrium buyer behavior in the presence of this additional Market-Maker option.

- *Reducing allocation and price uncertainties.* In the new equilibrium after the addition of a Market-Maker option, we show that buyers who opt for the Market-Maker reduce both their allocation uncertainty and price uncertainty. In particular, compared to the equilibrium clearing price that would have existed in the absence of the Market-Maker, the Market-Maker charges a premium above this price as a fee that buyers pay for reducing their allocation and price uncertainties.

- *Pareto improvement in seller's revenue and sum of buyer utilities.* We show that in all the settings we consider, adding the Market-Maker contract can only increase both the seller's revenue and the sum of buyer utilities. For some commonly used buyer value distributions, we further prove that this increase is significant.

We now describe these contributions more fully, including some of the challenges faced, and surprising observations.

Throughout this chapter, we assume for simplicity that there is a large inventory I of identical, indivisible items. Each buyer is interested in exactly one unit of inventory, and has a private value drawn independently from a common distribution F.[3]

In *preferred deals* sold on Ad Exchanges, sellers and buyers agree on a fixed (posted) price per impression, but there is no guarantee that buyers will bid in any volume, nor does the seller guarantee that there will be any inventory available to bid on, let alone sufficient inventory for all the buyers interested in purchasing at

[3]In practice, a seller's inventory may be segmented into different sections with different values, but each segment can be treated independently. Similarly, buyers typically want to buy a large number of impressions d, but one can break up each such buyer into d single-unit buyers. In reality, buyers may have different valuation distributions, overlapping targeting, etc., and extending our model to some of these more realistic cases is an interesting direction for future work.

the posted price. When multiple buyers bid at this price, the impression is allocated randomly to one of them. This motivates an envy-based utility model, where a buyer receives negative utility for not being allocated even though he was willing to pay the price at which another buyer was. In the absence of the Market-Maker, we show examples where the total welfare can be significantly lower than optimal, but flexibility in adding a second pricing option (i.e. the Market-Maker option) can result in near-optimal welfare. Further, adding the Market-Maker option can result in a Pareto improvement in both the seller revenue and the utility for each buyer.

In the auction setting, we assume a standard multi-unit auction (that is, the I items are sold to the I highest bidders at a price equal to the the $(I+1)^{th}$-highest bid).[4] However, there is an additional layer of complexity in analyzing the equilibrium both with and without the Market-Maker. When utilities are not quasi-linear, it is not clear how buyers should bid at all, since it is not immediately obvious that a second-price auction (or its generalization to the multi-unit case) is truthful. For example, buyers may choose to shade their bids in order to decrease their pricing uncertainty by avoiding a small probability of a very high price. We prove that the auction is indeed truthful in the standard risk-aversion models we consider, demonstrating the robustness of the second-price auction even in the presence of risk aversion. Armed with these results, we can analyze outcomes in the auction setting.

In the auction setting, we first consider the case in which all buyers exhibit the same degree of risk aversion (though this is perhaps unrealistic, it provides useful insights for heterogenous degrees of risk aversion, which we consider later, and is interesting in its own right). We show that there is a *unique* Market-Maker price such that (a) the Market-Maker runs no risk of defaulting (which could be a possibility if more than I buyers choose this option) and (b) at least one buyer opts for the Market-

[4] In reality, there are I repeated second-price auctions, but if buyers are paying a price higher than the $(I+1)^{th}$ bid, they have an incentive to lower their bids. Therefore, we assume that all I items are sold at this price.

Maker (if no buyer opts for it, it serves no purpose). We prove this by showing that for any price p quoted by the Market-Maker, there is a threshold value $v(p)$ such that any buyer with value at least $v(p)$ will choose the Market-Maker, and no buyer with lower value will choose it. This also results in the somewhat counter-intuitive fact that even the buyers with very high values (who have zero allocation uncertainty, since they know they will always win in the auction) will choose the Market-Maker; this is because in the homogenous risk-aversion case, the Market-Maker only changes prices, and not the allocation, meaning that the premium is paid purely to reduce pricing uncertainty. At this unique Market-Maker price, we show that there is a Pareto increase in both the seller revenue and the sum of buyer utilities.

We then consider the more realistic case of heterogenous risk aversion; here, the Market-Maker can change the allocation in addition to reducing pricing uncertainty. In particular, a more risk-averse buyer may buy the Market-Maker contract and win an impression, while a less risk-averse buyer with a higher value who decided to take a chance may be left unallocated. Now, the Market-Maker price is no longer unique, and we characterize the range of feasible Market-Maker prices. We prove that there exists at least one price that gives a Pareto improvement in the seller revenue and the sum of buyer utilities; choosing other points in the range allows trading off these two objectives.

We believe that a significant strength of our paper is that *all* the results described above for the auction setting hold in both the models we consider (though the proofs are quite different), showing that our results are robust and not tied to a particular model for risk aversion.

5.1.2 Related Literature

As we discussed, this chapter is related to several streams of literature.

In the Marketing community, the topic of advance selling has received great attention in the last two decades (see, e.g., [69], [70] and [28]). In [69], the authors show that advance selling allows sellers to improve profits. In particular, they prove that when buyers are homogeneous in the advance period purchase, advance selling can attain the profits from first degree price discrimination (even when the seller cannot price discriminate in the consumption period). Subsequently, [70] extend the treatment to competitive environments and show that the relative profit advantage from advance selling in a competitive market can be higher or the same relative to a setting with a monopolist.

In the Operations Management community, advance selling was also studied (see, e.g., [63], [19], [15] and [12]). The work by [63] study advance selling in a newsvendor setting. The authors examine the advance selling price and inventory decisions in a two-period setting, and conclude that advance selling is not always optimal. In [19], the authors study a supply chain setting with a manufacturer who produces and sells a seasonal product to a retailer under uncertain supply and demand. They model the problem as a Stackelberg game and study the impact of advance selling on both the manufacturer and the retailer. In [15], the author studies how the allocation of inventory risk impacts the supply chain efficiency under advance-purchase discount contracts. It is shown that if firms consider advance-purchase discounts, then the coordination of the supply chain and the arbitrary allocation of its profit is possible. In [12], the authors study a model that uses the acquired advance sales information to decide the capacity. They derive a threshold policy that determines when to stop acquiring advance sales information and show that advance selling can improve profit significantly. Finally, the recent article [16] studies a similar problem as this chapter in the context of an online multi-unit auction. In their model, the seller faces a Poisson arrival stream of consumers who can get the product from the auction or from a list price channel. Each consumer maximizes his own surplus, and must decide either to buy at the posted price and get the item at no risk, or to join the auction and wait until its end. This chapter differs by explicitly modeling the risk aversion of buyers in the utility function, instead of assuming different arrival times that are Poisson distributed. In addition, we focus on studying how to design and set the price of the advance selling option and we study the impact on the buyers and the sellers. We note that in both the Marketing and Operations Management literatures, most of the previous works on advance selling aim to mitigate the uncertainty in the buyer's valuation (or the consumption level). One of the key messages is that advance selling helps the seller to increase its profit by offering a discounted price to the buyers who can commit to make the purchase in advance. In this chapter, however, the motivation is different in nature as our goal is to capture the risk aversion of buyers (or advertisers) and to offer a premium price for the Market-Maker contract. In this paper, we show that in addition to the benefit for the buyers, it also increase the revenue of the seller.

In the Finance literature, futures contracts are a very well studied area of research (see, [41], [40], [38] and the references therein). As we previously mentioned, the financial contract exists to protect both buyers and sellers from uncertainty in future prices. Since both parties benefit from advance selling, the price is not necessarily discounted and usually depends on risk factors that leads to a no-arbitrage pricing policy. A large number of strategies that aim to price such contracts were developed and implemented but this is beyond the scope of this chapter.

Finally, several relatively recent papers in the CS/Econ community are also related. Notably, [37] independently considered the problem of guaranteed and spot markets coexisting, specifically in the cloud computing market. One important difference is that their model assumes that the seller is the agent offering the two options, and explicitly sets aside inventory for the purchasers of the guarantee. In contrast, our market maker cannot set aside inventory, as it is not the seller;[5] it bids in the auc-

[5] Individual publishers in the display advertising context are unlikely to have the scale and/or

tion to obtain inventory for its buyers, and hence makes a loss with some probability (though it makes a profit in expectation). As such, this service can be offered by any arbitrary third party wiling to accept the arbitrage risk, though it is likely to be the exchange, passing on a large portion of the profits to publishers. A few other papers consider settings in which buyers can pay more to get a higher chance of winning an item, but they are not motivated by risk aversion: [74] describes how publisher can sell an options contract that gives advertisers the right to buy ads later at a particular price; their work differs from ours by focusing on reducing *seller* revenue volatility by accepting lower average revenue. [56] studies how publishers can increase revenue by *bundling* impressions and offer advertisers a fixed take-it-or-leave-it price. In this work, advertisers who buy a bundle do receive a guarantee, but they are not motivated to do so by risk aversion; instead, if they reject the bundle, they are barred from the auction. Finally, [18] describes a modified auction where for each item, buyers are offered the choice between paying a high fixed price, or taking their chances in a lottery; the authors show that this tool can extract additional revenue particularly in thin auctions where only a single buyer is likely to have a high value.

Structure of the chapter. We first study the case of a posted price in Section 5.2. In Section 5.3, we consider the case where ads are sold via auctions and extend our analysis to this setting. In Section 5.4, we run computational experiments on commonly used distributions to illustrate the lift in revenue and buyer utilities that we get from adding the Market-Maker contract.

5.2 Posted price

In this section, we consider a setting with a posted price contract which corresponds to the fixed-price cpm deals in the online advertising world. More precisely, online adver-

technological sophistication to offer such a service.

tisers can sign contracts at a pre-determined posted price set by the seller. Typically, these types of contracts do not provide any guarantee of delivery to the advertisers; they only provide a price guarantee if allocated. Our goal is to study the benefits of introducing a new additional type of contract, called the *Market Maker contract*, that provides a delivery guarantee. The seller will offer both types of contracts to advertisers, who can then choose between the two options.

5.2.1 Model

Consider a single seller or publisher (e.g., a website such as the New York Times selling online ads). Let I be the amount of inventory (number of ads available), and N be the number of buyers or advertisers interested in purchasing this inventory. For simplicity, we assume that all the inventory units are identical,[6] that each buyer is interested in a single unit, and each unit is equally valuable to all N buyers. For example, the seller wants to sell I ads slots for a specific day next month or for a special event (e.g., Valentine's day). We assume that I and N are deterministic and known to the seller[7] (for instance, the publisher has a reasonably accurate estimate based on the number of users that visited his website in the past, and similarly the number of buyers with whom he has long-term relationships with). Each buyer has a private valuation v drawn i.i.d. from some discrete distribution F. We focus specifically on discrete distributions because several aspects of the buyer population like re-marketing buyers etc. make the distribution bimodal or multi-modal and discrete distributions model this well[8]. The seller decides upon a posted price p

[6]This is indeed the case if we partition the inventory/user-eyeballs that the seller has available into several segments based on the feature-list of each user, and focus on each segment separately. Within each segment, buyers have the same value for any unit.

[7]Even with I and N deterministic we show that an additional market-maker price can significantly increase seller revenue, buyer utilities, and efficiency of allocation.

[8]For the continuous distribution case where even a single price is enough to optimize welfare and efficiency.

and then, the inventory is randomly allocated among all the buyers with valuation that exceeds p. The random allocation mechanism is motivated by the fact that the buyers arrive in a random order, and are served on a first-come-first-serve basis. Consequently, depending on I, N and the buyers' value distribution, some buyers may not be served even though their valuations are above p. (Indeed this happens with real world fixed-price cpm deals, and is part of the reason why these deals don't promise allocation guarantees). This phenomenon makes such buyers experience envy: the inability to purchase a good that was priced below their value, whereas another buyer could. We consider a utility model for the buyer that captures her envy when a good is not allocated. More precisely, we consider the following utility model:

$$U(v,p,\beta) = \begin{cases} v - p & \text{if } v \geq p \text{ and allocated an item at price } p \\ -\beta \cdot (v-p) & \text{if } v \geq p \text{ and not allocated an item} \\ 0 & \text{otherwise.} \end{cases}$$

If the valuation v is less than the posted price p, the buyer perceives 0 utility. If $v \geq p$, the buyer is interested in purchasing the item. If the buyer is allocated, her utility is equal to the quasilinear utility $v - p$, and if she is not allocated (due to the random allocation mechanism), she perceives a negative utility equal to $-\beta \cdot (v - p)$, where β represents the *envy parameter* of the buyer, and could be different for different buyers. In this case, the buyer can afford the item, and seeing others receiving the item, makes her perceive a disutility/envy.

Welfare vs Efficiency. Efficiency of an outcome is the sum of valuations of buyers who were allocated. Welfare of an outcome is the sum of utilities of all buyers and sellers. With quasilinear utilities, welfare = efficiency. But with the envy-based utility we study, some buyers could experience disutility due to envy and therefore, we have welfare $\leq$ efficiency.

Market-Maker price. We study the effect of adding a Market Maker contract here, namely, we provide an additional price p^M along with the guarantee that buyers who pay p^M are guaranteed to get an item, and those who pay p will enter a uniformly random lottery along with other buyers who also paid p, and the remaining inventory after serving the market-maker confirmed buyers will be distributed in this random fashion.

Fluid assumption. We assume N is large enough such that for each point x in the support of F, the number of buyers with value less than x is exactly $N(1-F(x))$. I.e., when N is large, concentration bounds puts this number very close to $N(1-F(x))$, and the fluid assumption makes this exactly at $N(1-F(x))$. This assumption is to simplify exposition and avoid notational clutter from concentration bounds that doesn't add any insight.

Further, we assume that the distribution F is *non-trivial*, namely, there is at least one point x in the support of F such that $N(1-F(x)) < I$. In words, there is at least point in the support of F where the supply is not exhausted completely. If F is trivial, by definition of triviality, no collection of prices is enough to give allocation guarantees, and we therefore ignore this case.

5.2.2 Results

The main message is that adding a market-maker price will simultaneously (sometimes only weakly) increase seller's revenue and sum of buyer's utilities. We now establish this result in the posted-price setting. In fact in this setting, we also show that adding the market-maker price also increases efficiency of allocation, namely the sum of valuations of the buyers who get allocated.

We analyze the effect of adding a market-maker price to an existing posted price. If the posted-price was already not exhausting supply, it does not make sense to add a market-maker at a higher price. Assume therefore that the prior posted price exhausts the supply. Among all supply exhausting prices, the chosen one could have optimized revenue or welfare or efficiency etc.. But we show in the following theorem that there exists a single price that simultaneously optimizes seller's revenue, efficiency of allocation and total welfare in the system (sum of utilities of seller and buyers). It therefore unambiguously establishes that the single posted price that existed before Market-Maker was added should have been p^-. We then show that offering a market maker price of p^M simultaneously with p^- Pareto improves all quantities of interest. The definitions of p^- and p^M are given in the statement of Theorem 5.1

Notation. We will often be interested in the quantity $F_<(x) = Pr[v < x]$, as opposed to the regular cdf $F(x) = Pr[v \leq x]$.

Theorem 5.1. *For any I, N and non-trivial value distribution F, and arbitrarily heterogeneous risk averse parameters (β_i for buyer i), the following are true.*

1. *Let $S = \{p \text{ in support of } F : N(1 - F_<(p)) \geq I\}$ be the finite set of prices that exhaust supply. Let $p^- = \max_{p \in S} p$. In the domain S, revenue, welfare and efficiency are all optimized simultaneously at p^-.*

2. *There exists a market-maker price p^M such that offering prices p^M and p^- to buyers (with p^M guaranteeing allocation), will:*

   a) *attain optimal efficiency: $E(p^M, p^-)$ is exactly the sum of the I highest values;*

   b) *attain optimal welfare, and no buyer experiences envy: $W(p^M, p^-) = E(p^M, p^-)$;*

   c) *strictly increase revenue: $R(p^M, p^-) > R(p^-)$, except, when $N(1-F_<(p^-)) = I$ we have $R(p^M, p^-) = R(p^-)$;*

d) strictly increase at least one buyer's utility: there exists an i for which $U(v_i, (p^M, p^-), \beta) > U(v_i, p^-, \beta_i)$, except when $N(1 - F_<(p^-)) = I$ we have $U(v_i, (p^M, p^-), \beta) = U(v_i, p^-, \beta_i)$ for all i.

e) weakly increase each buyer's utility: for each i, we have $U(v_i, (p^M, p^-), \beta_i) \geq U(v_i, p^-, \beta_i)$.

Proof. We prove the theorem in two parts.

Proof of part-1 Consider the smallest point larger than p^- in the support of F. Call it p^+. Note that $N(1 - F_<(p^-)) \geq I$ by definition of set S (note that $S \ni p^-$) and $N(1 - F_<(p^+)) < I$.

1. Revenue: Since the market clears for all prices in S, the revenue $R(p)$ for any $p \in S$ is exactly $I \cdot p$. Since p^- is the maximum price in S, among all $p \in S$, the revenue $R(p)$ is maximized at p^-.

2. Efficiency: Consider any price $p \leq p^-$. Let I_p be the number of buyers with value at least p. Note that $I_p > I$ whenever $p \leq p^-$. Let $v_1 \geq \cdots \geq v_{I_p}$ be the values of all the buyers with value at least p. The efficiency at any at p is:.

$$E(p) = \frac{I}{I_p} \sum_{i=1}^{I_p} v_i.$$

As we decrease p, I_p increases, making the inventory open to more and more lower value buyers, at the cost of decreased probability of allocation for high value buyers. It immediately follows that $E(p)$ increases with p and the optimal efficiency among points in S is obtained at p^-.

3. Welfare: Let $p \in S, I_p$ and $v_1 \geq \ldots v_{I_p}$ be defined as before. Welfare at p, which is the sum of all buyer's utilities and seller's revenue, is defined as follows.

$$W(p) = \frac{I}{I_p} \sum_{i=1}^{I_p} v_i - \left(1 - \frac{I}{I_p}\right) \sum_{i=1}^{I_p} \beta_i \cdot (v_i - p).$$

The first term constitutes the quasi-linear utility of the buyers plus the seller's revenue (that's why the prices cancel each other and don't appear). The second term constitutes the envy experienced by buyers. Note that since I_p decreases as p increases, the first term clearly increases with p. In the second term, the factor $(1 - \frac{I}{I_p})$ clearly decreases with p, and so does the factor $v_i - p$. Thus the second term decreases with p. Thus $W(p)$ increases with p, showing that $W(p)$ is maximized at p^- among all prices $p \in S$.

Proof of part-2 Consider posting a Market-maker price of $p^M = p^- + \epsilon$ for a tiny ϵ. We analyze the buyer behavior equilibrium, namely, which set of buyers will purchase at Market-Maker price of p^M, and which set of buyers will buy at p^-.

Claim 5.1. *The unique equilibrium at prices (p^M, p^-) is as follows. When $N(1 - F_<(p^-)) = I$, the top I buyers get allocated at the price of p^-, and no buyers will every choose the Market-Maker price, and no buyer experiences envy, and optimal welfare and efficiency are already achieved without Market-Maker. But if $N(1 - F_<(p^-)) > I$, buyers with $v \geq p^+$ will purchase at Market-Maker price of $p^M = p^- + \epsilon$, and buyers with $v = p^-$ will opt for a price of p^- and enter a lottery to randomly share the left over $I - I_{p^+}$ items (left over after Market-Maker serves I_{p^+} buyers with values at least p^+). Buyers with $v < p^-$ do not get allocated.*

Proof of Claim. The proof for $N(1 - F_<(p^-)) = I$ is immediate. We prove the other case in the claim in three parts:

1. Buyers with $v < p^-$ do not face any envy even if unallocated since both prices are strictly larger than p^-, and hence will go unallocated.

2. Buyers with $v = p^-$ will also face no envy because their quasi-linear utility when opting for a price of p^- is exactly $v - p^- = 0$. Thus they will prefer to get a 0 utility than opting for p^M and getting negative utility.

3. To see that all the I_{p^+} buyers with $v \geq p^+$ will opt for the Market-Maker price of p^M: suppose on the contrary only $0 \leq d < I_{p^+}$ buyers with $v \geq p^+$ opt for Market-Maker price of p^M. Under this equilibrium, consider the utility of a buyer i with value $v_i \geq p^+$ who has not opted for the Market-Maker. There are only $I - d$ units available for sale after serving Market-Maker buyers.

$$U(v_i, p^-, \beta_i) = \frac{I-d}{I_{p_-}}(v_i - p^-) - \beta_i \left(1 - \frac{I-d}{I_{p_-}}\right)(v_i - p^-).$$

Clearly, $U(v_i, p^-, \beta_i) < v_i - p^-$ because $I - d \leq I < I_{p^-}$. Thus, there exists a sufficiently small ϵ such that $U(v_i, p^-, \beta_i) < v_i - (p^- + \epsilon) = v_i - p^M$. Thus it was strictly sub-optimal for the buyer to have not opted for the Market-Maker price. This proves the claim.

□

Armed with the claim, we now show prove the theorem.

1. <u>Optimal efficiency achieved:</u> $E(p^M, p^-)$ is the optimal efficiency achievable, i.e., it is the sum of the highest I values. This is immediate because the highest I_{p^+} values pick Market-Maker price and are guaranteed to get allocated. All the remaining buyers who get allocated are at the next highest possible value after p^+, namely p^-. It does not matter which among the buyers with $v = p^-$ are getting allocated, so randomness in allocation for those buyers will not affect efficiency.

2. <u>Optimal welfare achieved:</u> We show that $W(p^M, p^-) = E(p^M, p^-)$. As explained in Section 5.2.1, welfare is always at most efficiency. Since we have already shown that $E(p^M, p^-)$ is optimal, if we now show that $W(p^M, p^-) = E(p^M, p^-)$, it follows that welfare is also optimal at pair (p^M, p^-). To establish that, all we have to show is that no buyers experience envy in the equilibrium allocation (i.e., there is no loss in utility due to envy), and then it immediately follows

that welfare = efficiency. Clearly buyers with value $v \geq p^+$ don't experience envy because they choose the Market-Maker price and are guaranteed to get allocated. Buyers with value $v = p^-$ have 0 quasi-linear utility, and therefore even if they don't get allocated due to the randomness in allocation, they don't experience envy. This proves the optimality of welfare.

3. Strictly increase revenue: The fact that $R(p^M, p^-) > R(p^-)$ (in the case when $N(1 - F_<(p^-)) > I$) immediately follows from noting that $R(p^-) = I \cdot p^-$, and $R(p^M, p^-) = I_{p^+}(p^- + \epsilon) + (I - I_{p^+})p^- > R(p^-)$ (our equilibrium behavior analysis says that I_{p^+} buyers purchase at a price of p^M when $N(1-F_<(p^-)) > I$) . When $N(1 - F_<(p^-)) = I$, no buyer chooses p^M, and revenue is $R(p^M, p^-) = R(p^-)$.

4. Strictly increase at least one buyer's utility: Buyers with value $v \geq p^+$ get strictly higher utility after adding Market-Maker option, i.e., if $v_i \geq p^+$, we have $v_i - p^M > U(v_i, p^-, \beta_i)$ (we proved this while deriving the equilibrium).

5. Weakly increase each buyer's utility: To show that for all buyers i, $U(v_i, (p^M, p^-), \beta_i) \geq U(v_i, p^-, \beta_i)$, note that buyers with values $v \leq p^-$ get 0 utility before and after adding Market-Maker price. Buyers with value $v \geq p^+$ experience a strict increase in utility as was just discussed in point 4 above.

□

In Theorem 5.1, we show that the price pair ($p^M = p^- + \epsilon,\ p^-$) optimizes welfare and efficiency, and Pareto improves other quantities of interest. While the ϵ was to just show existence, in practice one could significantly increase the Market-Maker price beyond p^- and significantly increase revenue. We demonstrate this for a few distributions in Section 5.4

While Theorem 5.1 establishes that Market-Maker price can simultaneously improve all quantities of interest, while also achieving optimal welfare and optimal efficiency, it does not quantify the extent of improvement that Market-Maker can provide. To do this, we show in Theorem 5.2 that having just a single posted price can lead to really bad welfare, i.e., we show that there exist distributions for which the single posted-price's welfare can be arbitrarily small as β gets very large.

Theorem 5.2. *Even when the risk aversion parameter β is the same across all buyers, there exists I, F, N such that the welfare from any single price is at most $\frac{1}{2+\beta}$ of the optimal welfare.*

Proof. Consider a setting where the support of F has two values v_2 and $v_1 > v_2$. We pick our v_1, v_2, F so that Let $N\big[1 - F_<(v_1)\big] < I$, i.e., the number of buyers I_{v_1} with value at least v_1 is strictly smaller than I, namely at a price of v_1, we don't exhaust supply. We further ensure that $v_2 \cdot \big[I - I_{v_1}\big] = (1+\beta) \cdot v_1 \cdot I_{v_1}$. (Note that we just have imposed 2 constraints so far, and we have 3 degrees of freedom namely v_1, v_2 and $F(v_2)$.). In this case, the optimal welfare is given by:

$$\begin{aligned} W^* &= v_1 \cdot I_{v_1} + v_2 \cdot \big[I - I_{v_1}\big] \\ &= (2+\beta) \cdot v_1 \cdot I_{v_1} \\ &= \frac{2+\beta}{1+\beta} \cdot v_2 \cdot \big[I - I_{v_1}\big]. \end{aligned}$$

We next compute the welfare evaluated at both values v_1 and v_2. Note that the values v_1 and v_2 are the only two relevant candidates for the posted price.

When the price is set to v_1, we have: $W(v_1) = v_1 \cdot I_{v_1}$ and therefore:

$$\frac{W(v_1)}{W^*} = \frac{1}{2+\beta}.$$

When the price is set to v_2, we have:

$$W(v_2) = \frac{I}{N} \cdot v_1 \cdot I_{v_1} - \beta(1 - \frac{I}{N})(v_1 - v_2)I_{v_1} + \frac{I}{N} \cdot v_2 \cdot (N - I_{v_1}).$$

Note that we have included seller's revenue also in this welfare, and that's why the first term has just v_1 instead of $v_1 - v_2$, and similarly that's why the third term has just v_2 instead of $v_2 - v_2$. Now, set $N >> I >> I_{v_1}$, and ignore the term $\frac{I}{N}$ in the above expression (except for the $\frac{I}{N}$ in the last term because it has a large multiplier, namely $N - I_{v_1}$). This gives us

$$\begin{aligned}
W(v_2) &\simeq 0 - \beta(v_1 - v_2)I_{v_1} + \frac{I}{N} \cdot v_2 \cdot (N - I_{v_1}) \\
&\leq v_2 I + \beta v_2 I_{v_1} - \beta v_1 I_{v_1} \qquad \text{(by ignoring the term } \frac{I}{N} \cdot v_2 \cdot I_{v_1}) \\
&= v_2 I + \beta v_2 I_{v_1} - \frac{\beta}{1+\beta} v_2 (I - I_{v_1}) \\
&= v_2 I \frac{1}{1+\beta} + \beta v_2 I_{v_1} \frac{2+\beta}{1+\beta}
\end{aligned}$$

Now consider $\frac{W(v_2)}{W^*}$, and substitute for W^* from the above derivation. We have

$$\begin{aligned}
\frac{W(v_2)}{W^*} &= \frac{1}{2+\beta} \frac{I}{I - I_{v_1}} + \beta \frac{I_{v_1}}{I - I_{v_1}} \\
&\simeq \frac{1}{2+\beta}.
\end{aligned}$$

This completes the proof. □

As β grows large, the welfare grows arbitrarily bad with a single price, i.e., there is a lot of disutility in the system because of the envy that arises out of random allocation hurting high value buyers. Note that even at $\beta = 0$, the welfare approximation by a single price is at least a factor 2. Where as, like we saw in Theorem 5.1, the addition of a single Market-Maker price, gives us the optimal welfare and efficiency for all β.

While we skip the proof here, it turns out that the factor $2 + \beta$ is tight, i.e., there is a single posted price that can give a $2 + \beta$ approximation to optimal welfare.

5.3 Auctions

In this section, we consider the most popular setting for Internet advertising, where the publisher sells the items via an auction mechanism, instead of a posted price. Note that a large fraction of real-time bidding exchanges operate under such a mechanism.

Buyers can post a bid for the item in real-time, and the I inventory units are allocated to the I highest bidders. More precisely, the I highest bidders win the auction, and they all pay the $(I+1)$-th bid, also called the *auction clearing price*. This mechanism is called a standard multi-unit auction and generalizes the second price auction, and is very common in both the academic literature and in practice. Our goal is to study the benefit of adding the Market Maker contract in such an auction setting. In particular, we show that it allows the seller to significantly increase both seller revenue and sum of buyer utilities.

5.3.1 Model

As discussed, we consider a setting where the items are allocated by running a second price auction. Note that buyers can suffer from two types of uncertainty: (i) allocation uncertainty, and (ii) price uncertainty. In other words, the buyers are not guaranteed to be allocated and if they are, the clearing price is also uncertain.

Fluid assumption. As in the posted-price setting, we make the fluid assumption that N is large, and consequently, the number of buyers with each value in the support of F is deterministic (i.e., it is so concentrated that it is effectively deterministic).

A consequence of the fluid assumption is that when N is deterministic, the auction clearing price, which is just the $I+1$-th order statistic among N draws from N is also deterministic. Thus, there is no uncertainty in allocation or pricing when N is deterministic. Note that this is unlike in the posted-prices setting where even at deterministic N there was significant allocation uncertainty due to the uniformly random allocation used when demand exceeded supply. The auction on the other hand is efficient, and doesn't have this issue. Therefore, adding the Market Maker contract when N is deterministic is not relevant. In reality, N is very often unknown and we study the stochastic N case in this section.

N **from discrete distribution,** v **from continuous or discrete distribution** F**.** In particular, we assume that N has a finite support distribution supported in $[N_{min}, N_{max}]$. The buyer values could be drawn from either discrete or continuous distribution. Note that the utility model used in Section 5.2 is not relevant here. In particular, the auction is efficient by nature, and no disutility/envy is perceived by the buyers. As a result, we consider two different utility models that are commonly used in the literature and capture the risk-aversion of the buyers.

Note that for each realization of N, we obtain a corresponding clearing price that is simply a deterministic function of the number of buyers N, due to the fluid assumption. Under this assumption, there exists a deterministic mapping between the uncertainty in N and the uncertainty in the clearing price p. Therefore, instead of considering the uncertainty in N, we use the uncertainty in the clearing price.

In order to model the buyer's risk aversion, we consider the two following utility models:

$$U^A(v,b) = \begin{cases} \mathbf{E}_p\left[(v-p)I_x\right] - \beta \cdot \sqrt{\mathrm{Var}_p\left[(v-p)I_x\right]} & \text{(Standard Deviation (SD) model),} \\ \mathbf{E}_p\left[1 - e^{-\alpha\cdot(v-p)I_x}\right] & \text{(CARA model).} \end{cases} \tag{5.1}$$

Here, $\alpha \geq 0$ and $\beta \geq 0$ are the parameters of each model that capture the risk aversion of the buyers, and I_x is the indicator that the buyer is allocated the item at his bid of b. In addition, $\mathbf{E}_p$ and Var_p denote the expectation and variance operators over the distribution of the auction clearing price. These two classes of utility models are commonly used in the literature and aim to capture the risk aversion of buyers. The SD model is used in finance (e.g., portfolio optimization) and in various Marketing applications. The Constant Absolute Risk Aversion (CARA) model is a very commonly used risk aversion model.

Truthfulness. Before introducing the Market-Maker option, is the $I+1$-th price auction even truthful with the above two utilities? Is it clear that the buyer doesn't stand to benefit by over-bidding or under-bidding? It is straight-forward to see this in the CARA model, but the proof is involved in the SD model. We skip the truthfulness proof here, and provide it in the full version of the paper.

Lemma 5.1. *The $I+1$-th price auction is truthful under the above two utility models, i.e.:*

1. *In the CARA model, $\forall \alpha \geq 0$, $\forall v' \neq v$, $U^A(v, v) \geq U^A(v, v')$.*
2. *In the SD model, $\forall \beta \geq 1$ $\forall v' \neq v$:*
   a) *If $U^A(v, v) \geq 0$, then $U^A(v, v) \geq U^A(v, v')$.*
   b) *If $U^A(v, v) < 0$, then $U^A(v, v') < 0$.*

We note the subtlety in Lemma 5.1; we don't claim that utility never increases by misreporting one's bid. We show that if the true utility is positive, one's misreported bid never yields anything more than the true utility. If the true utility is negative, one's misreported bid could yield higher than the true utility, but it is still negative as well. An agent with negative utility simply does not participate in the auction, and gets 0 utility instead.

Is a negative utility at one's true value meaningful? We claim that it is meaningful. It means that the agent is so risk averse that any amount of uncertainty in outcome is enough to cause net disutility to her. Such agents simply don't participate in the auction at all. Importantly, Market-Maker helps such buyers by providing a risk-free option that they will consider purchasing. Namely, the Market-Maker increases the net buyer participation.

Now that we said that the auction is truthful, we rewrite the utilities by dropping the bid (bid = value) as:

$$U^A(v) = \begin{cases} \mathbf{E}_p\left[(v-p)^+\right] - \beta \cdot \sqrt{\mathrm{Var}_p\left[(v-p)^+\right]} & \text{(Standard Deviation (SD) model),} \\ \mathbf{E}_p\left[1 - e^{-\alpha\cdot(v-p)^+}\right] & \text{(CARA model).} \end{cases} \tag{5.2}$$

Market-Maker and auction. Assume that we now introduce the Market Maker option at a price p_M. The buyer's utility with valuation v for selecting the Market Maker contract is given by:

$$U^M(v, p^M) = \begin{cases} (v - p^M)^+ & \text{(SD model),} \\ 1 - e^{-\alpha\cdot(v-p^M)^+} & \text{(CARA model).} \end{cases}$$

Note that in the SD model, the standard deviation term disappears as the value of p_M is deterministic. The Market-Maker bids on behalf of its buyers in the auction. It bids a very high number (essentially ∞) so that it is guaranteed to get allocated. The only way a Market-Maker can default on its allocation promise is when more than I buyers choose to buy Market-Maker. The Market-Maker price has to be designed to avoid this, and yet should not be too high to get 0 or tiny incremental revenue.

Equilibrium behavior. If provided both the market-maker and the auction, how will the equilibrium buyer behavior be? Which buyer values will choose which option? The buyers while making this decision know their own values, but do not know the total number of buyers N who are entering the system. This causes some allocation and pricing uncertainty in the auction. In the next section, we analyze this setting. We use our equilibrium behavior analysis to guide how the market-maker price should be set so that the Market-Maker never defaults, i.e., never gets into a demand-more-than-supply situation. What will happen to total revenue? What will happen to total utility? All these in next section.

5.3.2 Homogeneous risk-aversion

In this section, we consider the setting where all the buyers have the same risk-aversion parameter (β and α for the SD and CARA models respectively). We will extend our results for the case of heterogeneous risk-aversion in Section 5.3.3. Our first result characterizes the equilibrium induced by the coexistence of both the auction mechanism and the Market Maker contract. In particular, we show that the buyers with high valuations choose the Market Maker.

Theorem 5.3. *For both CARA and SD utility models, and for a given price p_M, if there exists a value $\tilde{v}$ such that $U^M(\tilde{v}, p^M) \geq U^A(\tilde{v})$, then for all $v \geq \tilde{v}$, $U^M(v, p^M) \geq U^A(v)$.*

Proof. We give separate analyses for the two utility models. For ease of exposition, we assume that the distribution of values is continuous for this proof.

<u>**Proof for SD model**</u> : We show that the difference of the Market-Maker and Auction utilities is increasing, thereby establishing the threshold property. We first rewrite the utility derived from the auction. Note that if N has a discrete distribution, then the auction clearing price is discrete as well. Let $G(\cdot)$ denote its cdf and let $g(\cdot)$ be the probability mass function of the clearing price. A clearing price only exists only when the game has an equilibrium; we begin by assuming that a clearing price exists, and later prove that because equilibrium exists, our assumption is valid.

$$U^A(v) = \int_0^v (v-p) \cdot g(p) \cdot dp - \beta \cdot \sqrt{\mathrm{Var}_p[(v-p)^+]}.$$

We compute the derivative of each term separately. For the first term:

$$\frac{\partial}{\partial v}\left(\int_0^v (v-p) \cdot g(p) \cdot dp\right) = \int_0^v g(p) \cdot dp = G(v).$$

The derivative of the second term can be written as:

$$\begin{aligned}
\frac{\partial}{\partial v}\left(\mathrm{Var}_p[(v-p)^+]\right) &= \frac{\partial}{\partial v}\left(\mathbf{E}_p[((v-p)^+)^2] - \mathbf{E}_p[(v-p)^+]^2\right) \\
&= \frac{\partial}{\partial v}\left(\int_0^v (v-p)^2 \cdot g(p)\cdot dp\right) - 2\cdot \mathbf{E}_p[(v-p)^+]\cdot G(v) \\
&= 2\cdot \int_0^v (v-p)\cdot g(p)\cdot dp - 2\cdot \mathbf{E}_p[(v-p)^+]\cdot G(v) \\
&= 2\cdot \mathbf{E}_p[(v-p)^+] - 2\cdot \mathbf{E}_p[(v-p)^+]\cdot G(v) \\
&= 2\cdot \mathbf{E}_p[(v-p)^+]\cdot \overline{G}(v).
\end{aligned}$$

Here, $\overline{G}(v) = 1 - G(v)$ denotes the complementary cdf of the auction clearing price. Note that the variance is an increasing function of v. Putting the two terms together, we obtain:

$$\frac{\partial}{\partial v}U^A(v) = G(v) - \beta \cdot \frac{\mathbf{E}_p[(v-p)^+]\cdot \overline{G}(v)}{2\sqrt{\mathrm{Var}_p[(v-p)^+]}}.$$

Since $U^M(\cdot)$ is a linear function of v for $v \geq p_M$, its derivative is equal to 1 for those values. Consequently,

$$\frac{\partial}{\partial v}(U^M(v) - U^A(v)) = \overline{G}(v)\cdot\left(1 + \beta\cdot\frac{\mathbf{E}_p[(v-p)^+]}{2\sqrt{\mathrm{Var}_p[(v-p)^+]}}\right) \geq 0.$$

Consequently, the difference in the utility functions is increasing with v, which shows the desired threshold property.

<u>CARA model</u> Unlike the SD model's proof, where we studied the derivative of the difference of utilities, here we study the derivative of the ratio of utilities. As before, we begin by computing the derivatives of the utility functions. For $v \geq p^M$, we have:

$$\begin{aligned}
\frac{\partial}{\partial v}U^M(v, p^M) &= e^{-\alpha\cdot(v-p^M)} \geq 0, \\
\frac{\partial^2}{\partial^2 v}U^M(v, p^M) &= -\alpha\cdot e^{-\alpha\cdot(v-p^M)} \leq 0.
\end{aligned}$$

In addition, one can write:

$$\frac{\partial}{\partial v}U^A(v) = \int_0^v e^{-\alpha\cdot(v-p)}\cdot g(p)\cdot dp \geq 0.$$

This shows that $U^M(\cdot, p^M)$ and $U^A(\cdot)$ are both increasing functions in v. In addition, $U^M(\cdot, p^M)$ is concave in v. Using the expressions for the derivatives, we have:

$$\frac{\partial U^A}{\partial v}(v) \Big/ \frac{\partial U^M}{\partial v}(v) = \int_0^v e^{-\alpha \cdot (p_M - p)} \cdot g(p)dp. \tag{5.3}$$

Note that the ratio in (5.3) is an increasing function of v. Furthermore, we can rewrite (for any $v \geq p^M$):

$$\begin{aligned} \frac{\partial}{\partial v} U^M(v, p^M)\Big|_{v=v_{max}} &= 1 - U^M(v, p^M), \\ \frac{\partial}{\partial v} U^A(v)\Big|_{v=v_{max}} &= 1 - U^A(v). \end{aligned}$$

Therefore, we obtain:

$$\frac{\partial U^A}{\partial v}(v) \Big/ \frac{\partial U^M}{\partial v}(v, p^M)\Big|_{v=v_{max}} = \frac{1 - U^A(v_{max})}{1 - U^M(v_{max}, p^M)}. \tag{5.4}$$

Here, v_{max} denotes the maximal value of the valuation v (if the support of v is unbounded, one can take $v_{max} = \infty$ and use a limit argument).

Note that for $v \leq p^M$, $U^M(v, p^M) = 0$. Consequently, $U^A(p^M) \geq U^M(p^M, p^M) = 0$. We consider 2 different cases. First, assume that $U^A(v_{max}) \leq U^M(v_{max}, p^M)$ so that the ratio in (5.4) is larger than 1. Since $U^A(p^M) \geq U^M(p^M, p^M) = 0$, the two functions $U^A(\cdot)$ and $U^M(\cdot, p^M)$ have to intersect at least once. In other words, there must exist at least one value $\bar{v} > p^M$ such that $U^M(\bar{v}, p^M) = U^A(\bar{v})$. In addition, it's not possible to have an even number of crossing points. Otherwise, it will contradict that $U^A(v_{max}) \leq U^M(v_{max}, p^M)$. Assume by contradiction that the number of crossing points is at least 3. In this case, this contradicts the fact the ratio of derivatives is increasing in v (using equation (5.3)). Therefore, there exists a single value $\bar{v}$ such that for all $v \geq \bar{v}$, $U^M(v, p^M) \geq U^A(v)$.

In the second case, we assume that $U^A(v_{max}) > U^M(v_{max}, p^M)$ so that the ratio in (5.4) is smaller than 1. Since the ratio of derivatives is an increasing function of v, the two functions $U^A(\cdot)$ and $U^M(\cdot, p^M)$ will not intersect. As a result, there cannot exist any value v such that $U^M(v, p^M) = U^A(v)$. Indeed, since the ratio of derivatives always remains strictly less than 1, this would contradict $U^A(v_{max}) > U^M(v_{max}, p^M)$. This concludes the proof. □

Corollary 5.1. *From Theorem 5.3 it follows that for any given p_M, there exists a threshold value $\bar{v}(p^M)$ such that for all $v \geq \bar{v}(p^M)$, $U^M(v, p^M) \geq U^A(v)$, and $U^M(v, p^M) < U^A(v)$ otherwise.*

Note that the threshold result is not immediately intuitive. One would expect that buyers with very high value feel certain about their allocations, and hence don't go for the Market-Maker option. But the opposite is true, as Theorem 5.3 shows. The reason for this is that although the allocation uncertainty is tiny for high value buyers, there is significant pricing uncertainty that stems from unknown number of buyers $N \in [N_{min}, N_{max}]$ (recall that buyers don't know N when they make the auction vs. Market-Maker decision).

Engineering the Market-Maker price. Armed with Theorem 5.3, we now consider the question of how the designer should design the Market-Maker price. The price should be such that the Market-Maker is never over demanded (i.e., never more than I buyers ask for it), and at the same time, increase revenue for seller and utility for buyers.

Theorem 5.4. *Let $p_{\max}$ be the auction clearing price (without the presence of Market-Maker) when $N = N_{max}$. Let p^M_* be the Market Maker price such that $\bar{v}(p^M_*) = p_{max}$. Then, we have:*

1. *A closed form formula for p^M_* is given by:*

$$p^M_* = \begin{cases} \min(\mu_A + \beta \cdot \sigma_A,\ p_{max}) & \textit{(SD model)}, \\ \min\left(\frac{1}{\alpha} \log\left(\mathbf{E}_p[e^{\alpha \cdot p}]\right),\ p_{max}\right) & \textit{(CARA model)}. \end{cases} \tag{5.5}$$

   Here μ_A and σ_A are the mean and standard deviation of the auction clearing price (without the Market-Maker option).

2. *Equilibrium buyer behavior: Buyers with value* $v \geq p_{max}$ *purchase the Market-Maker option, and buyers with* $v < p_{max}$ *buy in the auction. In addition, for any value* $v \geq p_{max}$*, we have* $U^A(v) = U^M(v, p_*^M)$*, i.e., buyers who buy Market-Maker are indifferent between Market-Maker and auction.*

3. $Rev(auction, p_*^M) > Rev(auction)$.

4. $Welfare(auction, p_*^M) > Welfare(auction)$.

5. *The unique feasible Market-Maker price is* p_*^M*. Any price above* p_*^M *will not be chosen by any buyer, and any price below* p_*^M *will make the Market-Maker default on his promise when* N_{max} *buyers arrive.*

Proof. Note that for $v \geq p_{\max}$, we have $(v-p)^+ = v - p$ for all p. This implies that (after realizing that since v is fixed, the variance of $(v-p)$ is simply the variance of p which is σ_A):

$$\begin{aligned} U^A(v) &= \begin{cases} v - \mu_A - \beta \cdot \sigma_A & \text{(SD model)} \\ 1 - e^{-\alpha \cdot v}\mathbf{E}_p[e^{\alpha \cdot p}] & \text{(CARA model)} \end{cases} \\ &= \begin{cases} v - p_M^* & \text{(SD model)} \\ 1 - e^{-\alpha \cdot (v - p_M^*)} & \text{(CARA model)} \end{cases} \\ &= U^M(v, p_M^*). \end{aligned}$$

Note that this means that all the buyers with value $v \geq p_{max}$ are indifferent between the Market Maker contract and the auction mechanism. We claim that such indifferent buyers still choose Market-Maker because the price is always fixed at p_*^M whereas the auction could at times be p_{max}, where as $p_*^M \leq p_{max}$. Note that $p_*^M > \mu_A$, i.e., the Market-Maker marks up the price above the mean auction clearing price, but it is smaller than the largest value that the auction clearing price can take.

For the revenue claim, note that since only buyers with value $v \geq p_{max}$ buy via Market-Maker the auction clearing price after Market-Maker is introduced never changes. I.e., only previously winning buyers continue to win now. Therefore, revenue from buyers with $v < p_{max}$ remains the same. Whereas the revenue from buyers with value more than p_{max} has increased from μ_A to $\mu_A + \beta\sigma_A$. We can show this for the CARA model as well.

For welfare, note that since the auction clearing price was unaffected, the utility of auction buyers was not affected. Market-Maker buyers were also unaffected as they were indifferent. Revenue strictly increases. Thus welfare strictly increases.

For unique feasible price, note that any price above p_*^M is rejected by all buyers. This is clear because already at p_*^M, buyers were indifferent between auction and market-maker. When Market-Maker puts a price below p_*^M, when N_{max} buyers are realized Market-Maker will default. Note that currently Market-Maker sells all I units when N_{max} is realized (the threshold p_{max} by definition has I people above it because it is obtained as I item auction's clearing price). If Market-Maker goes any smaller, it will have more than I demand and will immediately default. □

Some remarks.

1. The Market-Maker price of p_*^M strictly increases with risk-parameter α and β (see Theorem 5.5 below). It also strictly increases with the variance of auction clearing price σ_A. This is true for both utility models.

2. Market-Maker clearing price is higher than the average auction clearing price of μ_A, but smaller than the highest possible auction clearing price of p_{max}.

3. Only buyers with value above p_{max} buy Market-Maker. The mass of these buyers is $(1 - F_<(p_{max}))$. And each of these buyers pays $p_*^M - \mu_A$ additional money in expectation. For the SD model, it is immediate to see this works out to $(1 - F_<(p_{max}))\beta\sigma_A$ as incremental revenue per buyer. Similarly for the CARA model, we can derive a closed form using the formula for p_*^M that we give.

We next show that the optimal Market Maker price p_M^* increases with respect to the risk aversion parameter (α or β) and with respect to the variance of the auction clearing price (σ_A).

Theorem 5.5. *p_M^* is an increasing function of the risk aversion parameter (α in CARA and β in SD). In addition, p_M^* is an increasing function of σ_A.*

Proof. Note that for the SD model, the results directly follows from the expression in (5.5). We next prove the result for the CARA model. The first order derivative with respect to α is given by:

$$(p_M^*)'(\alpha) = -\frac{1}{\alpha^2} \cdot \log(\mathbf{E}_p[e^{\alpha \cdot p}]) + \frac{1}{\alpha} \cdot \frac{\mathbf{E}_p[p \cdot e^{\alpha \cdot p}]}{\mathbf{E}_p[e^{\alpha \cdot p}]} = -\frac{1}{\alpha^2} \cdot h(\alpha),$$

where we denote:

$$h(\alpha) = \log(\mathbf{E}_p[e^{\alpha \cdot p}]) - \alpha \cdot \frac{\mathbf{E}_p[p \cdot e^{\alpha \cdot p}]}{\mathbf{E}_p[e^{\alpha \cdot p}]}.$$

Note that $h(0) = 0$ and therefore, it suffices to show that $h'(\alpha) \leq 0$ for all $\alpha \geq 0$. The first derivative of $h(\alpha)$ is given by:

$$\begin{aligned} h'(\alpha) &= \frac{\mathbf{E}_p[p \cdot e^{\alpha \cdot p}]}{\mathbf{E}_p[e^{\alpha \cdot p}]} - \frac{\mathbf{E}_p[p \cdot e^{\alpha \cdot p}]}{\mathbf{E}_p[e^{\alpha \cdot p}]} - \alpha \cdot \frac{\mathbf{E}_p[p^2 \cdot e^{\alpha \cdot p}] \cdot \mathbf{E}_p[e^{\alpha \cdot p}] - \mathbf{E}_p[p \cdot e^{\alpha \cdot p}]^2}{\mathbf{E}_p[e^{\alpha \cdot p}]^2} \\ &= -\alpha \cdot \frac{\mathbf{E}_p[p^2 \cdot e^{\alpha \cdot p}] \cdot \mathbf{E}_p[e^{\alpha \cdot p}] - \mathbf{E}_p[p \cdot e^{\alpha \cdot p}]^2}{\mathbf{E}_p[e^{\alpha \cdot p}]^2} \leq 0, \end{aligned}$$

where the last inequality follows from the Cauchy-Schwarz inequality. Since $h'(\alpha) \leq 0$ for all $\alpha \geq 0$ and $h(0) = 0$, we have: $h(\alpha) \leq 0$ for all $\alpha \geq 0$ and therefore, $(p_M^*)'(\alpha) \geq 0$ for all $\alpha \geq 0$. □

In conclusion, we have shown that adding the Market Maker contract allows to reduce the price uncertainty for high valuation buyers ($v > p_{max}$), without changing the allocation. We also demonstrated that there exists a unique Market Maker price that increases with both the risk aversion and the variability. Therefore, by assuming that indifferent buyers choose the Market Maker, the seller can strictly increase its revenue without changing the surplus of the buyers. For example, using the SD model, the revenue increase amounts to $\bar{F}(p_{max})\beta\sigma_A$. In Section 5.4 in Appendix, we show computationally that for realistic instances, this revenue increase can be significant.

5.3.3 Heterogeneous risk aversion

So far, we consider that all the buyers have the same risk aversion parameter (either α or β depending on the utility model). In practice, different buyers may behave differently with regard to risk. We study the setting where the buyers have heterogeneous risk aversion, and show the benefit of adding the Market Maker contract in such a setting. Let there be k different populations, with ρ_i mass in population i.

Theorem 5.6. *In the heterogeneous risk-averse buyers setting, the Market-Maker price is not necessarily unique. There exists a range of possible prices depending on the range of the risk-parameters. But there always exists a single price that strictly increases seller revenue and system welfare, i.e., offers a Pareto improvement.*

Proof. Let $\beta_1 < \ldots \beta_k$ be the risk-aversion parameters. Clearly β_k is the most risk-averse buyer. Set the Market-Maker price assuming that the entire population mass is at β_k. I.e., set a price of $p_*^M = \mu_A + \beta_k \sigma_A$ (and similarly for the CARA model). By our argument in the homogeneous case, buyers in the k-th population with value at least p_{max} will be indifferent between auction and Market-Maker and will purchase Market-Maker. But the less risk-averse buyers are strictly preferring auction at this price. At this price, the auction clearing price is unaffected, and revenue strictly increases, just like the homogeneous setting. This proves the Pareto-improvement part.

For the range of prices, note that as we keep decreasing the Market-Maker price, one-by-one the less risk-averse population will switch to Market-Maker till Market-

Maker defaults. As Market-Maker price falls, and as more agents choose Market-Maker, the auction clearing price rises. □

Homogeneous vs Heterogeneous: Change in allocation A notable aspect of the heterogeneous Market-Maker dynamics is that a lower value buyer from a higher risk-averse population gets allocated when a higher value buyer from a lower risk-averse population doesn't. To see this, let N_{max} buyers be realized, and at Market-Maker clearing price of p_*^M, the threshold for Market-Maker is exactly p_{max} for β_k population. At this point there are exactly I buyers with value above p_{max} and only $I\rho_k$ of them take Market-Maker and rest win in auction as they strictly prefer auction. The rest of the population is fully composed of non-β_k buyers. As the Market-Maker price decreases by a tiny ϵ, population-k buyers with value just below p_{max}, who would have no chance in a pure auction, will now switch to Market-Maker and get allocated. For every population-k buyer that moves to Market-Maker with value below p_{max}, some non-population-k buyer with value above p_{max} loses in the auction after Market-Maker because there are only I units available! The natural question is, why does the buyer who gets edged out not buy the Market-Maker herself instead of getting 0 utility, given that she can afford Market-Maker? The answer is that such a buyer doesn't get 0 utility — she loses her allocation only when N_{max} realizes. For much smaller values of N, she will get allocated in auction and get a much higher utility. As the Market-Maker price keeps dropping, the auction clearing price keeps increasing because of migration of low value population-k buyers to Market-Maker. At some point buyers from population-$k-1$ make the switch to Market-Maker and so on.

While our theorems establish structure of Market-Maker and show the Pareto-improvement provided by it, we show in the next section that for several commonly used distributions the lift in revenue and buyer utilities is significant.

5.4 Computational experiments

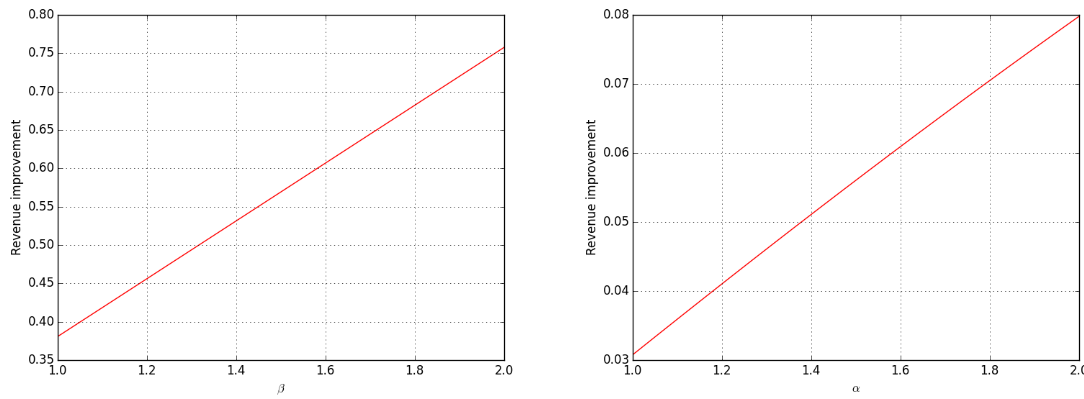
We first consider the setting with the auctions and study computationally the benefit of adding the Market-Maker. Then, we also consider the posted price environment.

5.4.1 Auctions

We first consider the setting with auctions. Our goal is to illustrate and quantify the results developed in Section 5.3.

5.4.1.1 Homogeneous risk aversion

In Section 5.3.2, we have shown that there exists a unique Market-Maker price p_M^*. In addition, we characterized this optimal price in closed form for both the SD and the CARA utility models. We also demonstrated that adding the Market-Maker contract increases the seller revenue without changing the utility of the buyers (as it does not modify the allocation). More precisely, the revenue increase per buyer that chooses the Market-Maker contract amounts to $\bar{F}(p_{max})\beta(p_M - \mu_A)$. Our goal is to show that this revenue improvement is significant relative to the revenue generated without the presence of the Market-Maker contract. We consider a setting with uniform valuations between 0 and 1 and two possible values for the number of buyers N (with equal probability). We assume that these two different values induce two distinct values of the auction clearing price: $p_L = 0.2$ and $p_H = 0.8$ (each with probability 0.5). In Figure 5.1, we vary the risk aversion parameter (β for the SD model and α for the CARA model) and compute the relative revenue improvement obtained by adding the Market-Maker contract. Note that in both utility models, we obtain a similar behavior as well as a potentially significant increase in revenue. In this example, the relative revenue improvement for the SD model when β is between 1 and 2 is between 38% and 76%, whereas for the CARA model, the relative revenue improvement is between 3% and 8% (when α is between 1 and 2).



(a) SD model (b) CARA model

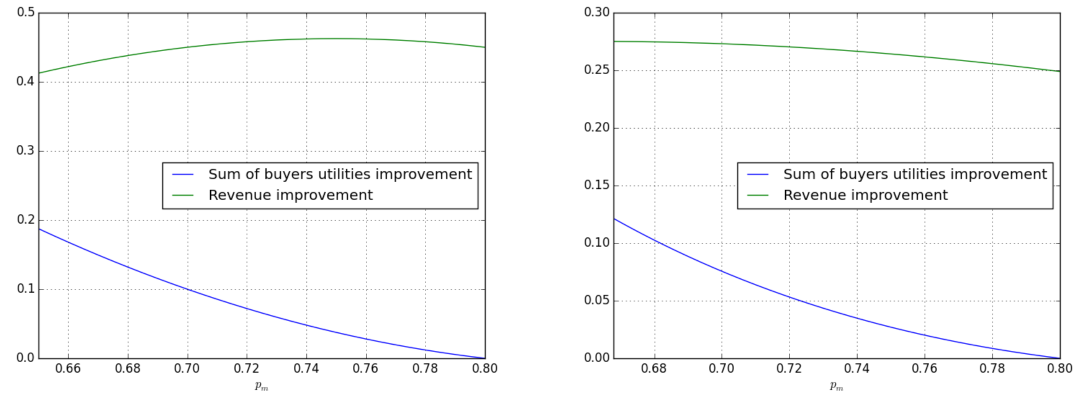
Figure 5.1: Relative improvement in the seller revenue by adding the Market-Maker contract for for the setting with auctions and homogeneous risk aversion.

5.4.1.2 Heterogeneous risk aversion

In Section 5.3.3, we studied the setting where buyers have heterogeneous risk aversion parameters. Next, we computationally illustrate and quantify the impact of adding the Market-Maker contract. We consider a setting with two populations of buyers: $\beta_1 = 1$ and $\beta_2 = 2$, i.e., population 2 is more risk averse. We assume that the proportions are equal, i.e., $\rho_1 = \rho_2 = 0.5$ and consider two different valuation distributions: uniform between 0 and 1 and exponential with mean 0.5. For each realization of the number of buyers N, we independently draw a split of the N buyers into the two populations. In addition, we consider a setting with two different values of N that induce two distinct auction clearing prices: $p_L = 0.2$ with probability 0.8 and $p_H = 0.8$ with probability 0.2.

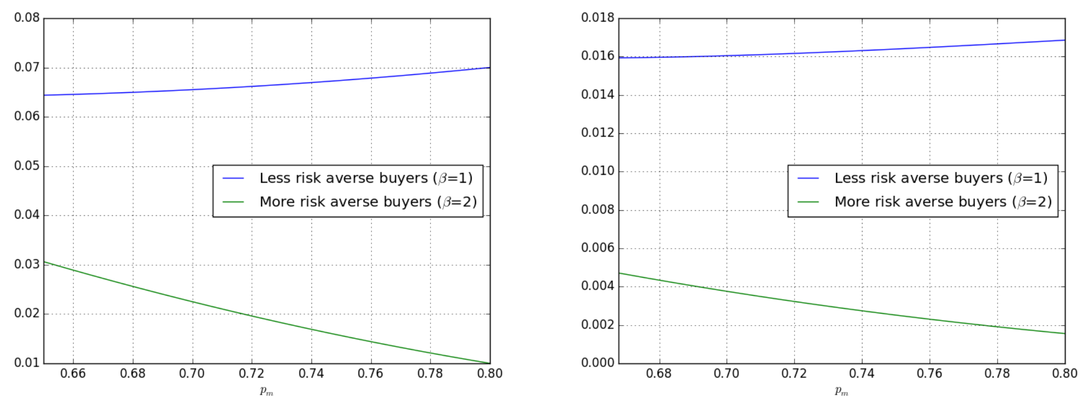
In Figure 5.2, we plot the relative improvement in the seller revenue and in the sum of buyer utilities as a function of p_M for both uniform and exponential valuation distributions. One can see that adding the Market-Maker contract yields a clear Pareto improvement. In other words, both the buyers and the seller will benefit from adding the Market-Maker contract. Note that the minimal meaningful value of p_M is such that the buyers in population 2 (i.e., the more risk averse buyers) are indifferent between the auction and the Market-Maker contract. As we saw in Section 5.3, this value is given by: $\min(\mu_A + \beta_2\sigma_A, p_H)$ and is denoted by Γ. In this example, we have: $\mu_A = 0.32$, $\beta_2 = 2$ and $\sigma_A = 0.48$ so that $\Gamma = 0.8$ (this corresponds to the rightest point on the x-axis in Figure 5.2). Then, when we start decreasing p_M below Γ, some buyers from population 2 will strictly prefer the Market-Maker contract. In this case, some buyers from population 2 will choose the Market-Maker contract, and some buyers from population 1 will be allocated via the auction. As p_M decreases, additional buyers from population 2 will choose the Market-Maker contract. Consequently, the auction clearing price increases, and some buyers from population 1 are not allocated anymore. In other words, items are secured through the Market-Maker contract from risk averse buyers at the expense of buyers from population 1 that now lose the auction. We continue decreasing p_M until the point where the Market-Maker defaults. Note that when $p_M = \Gamma$, the buyers from population 2 are indifferent between the two options so that the sum of buyers utilities stay the same. As we start decreasing p_M, it yields a Pareto improvement for both the sellers and the buyers.

In Figure 5.3, we plot the utilities of the buyers for each population separately as a function of p_M. As we decrease p_M, we increase the utilities of population 2 (more risk averse) and decrease the utilities of population 1. This follows from the fact that additional buyers from population 2 choose the Market-Maker in order to secure an allocation. At the same time, since the auction clearing price increases, less buyers from population 1 are allocated so that the utilities of population 1 reduce.



(a) Uniform valuations (b) Exponential valuations

Figure 5.2: Relative improvement in the seller revenue and the sum of buyer utilities by adding the Market-Maker contract for the setting with auctions and heterogeneous risk aversions.



(a) Uniform valuations (b) Exponential valuations

Figure 5.3: Relative changes in the buyer utilities by adding the Market-Maker contract for the setting with auctions and heterogeneous risk aversions.

5.4.2 Posted price

We now consider the setting with a posted price mechanism which is studied in Section 5.2. We next illustrate the fact that having only a posted price mechanism can suffer from a poor welfare performance, as we have shown in Theorem 5.2. We consider the setting where the number of buyers N is deterministic and equal to $2I$, and the valuations follow a 2-point distribution ($v = 1$ with probability 0.1 and $v = 0.1$

with probability 0.9). This setting can capture the realistic situation of retargeting in Internet display advertising. In Figure 5.4, we consider different values of the envy parameter β ($\beta = 0, 0.3, 0.6$) and plot the welfare attained by the posted price mechanism relative to the optimal welfare W^* (for more details, see Section 5.2). One can see that in this example, the welfare loss is quite significant (about 18%). Note that in Figure 5.4, the best fixed price yields the same welfare independent of the value of β. Indeed, when the price is high enough, no buyer experiences any envy and therefore, the welfare is independent of the parameter β. Recall that we have shown in Theorem 5.1 that adding the Market-Maker contract recovers the optimal welfare and at the same time, increases both the seller revenue and the sum of the buyer utilities.

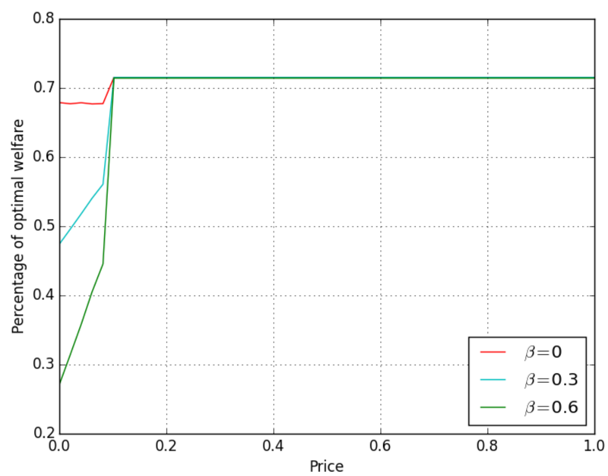
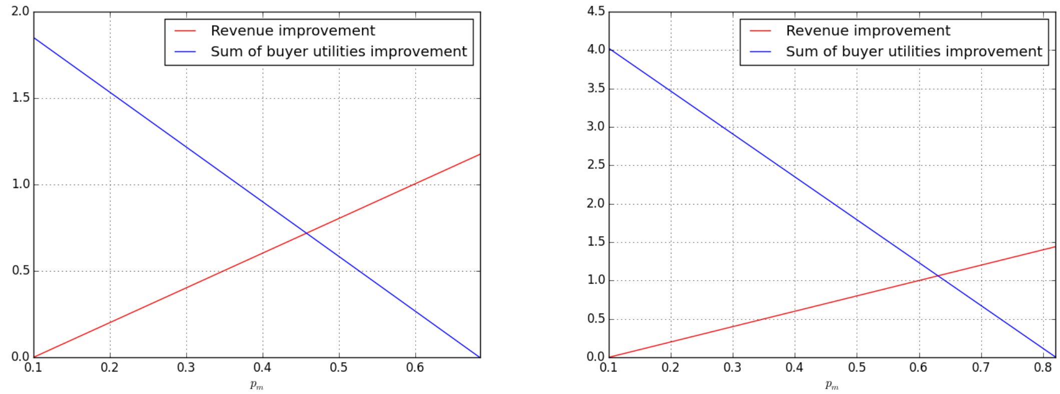


Figure 5.4: Welfare performance of the posted price for different values of β.

In Figure 5.5, we set the posted price to 0.1 and vary the price of the Market-Maker contract p^M between 0.1 and the value at which the Market-Maker defaults (i.e., more than I buyers choose this option). We consider two values of β ($\beta = 0.3$ and $\beta = 0.6$). One can see that for a wide range values of p^M, adding the Market-Maker contract yields a Pareto improvement in both the seller revenue and the sum of the buyer utilities relative to the case with only a posted price. Note that these relative improvements can be very significant (in this example, more than 70% in both metrics) and their magnitude increase with β.



(a) Setting with $\beta = 0.3$

(b) Setting with $\beta = 0.6$

Figure 5.5: Relative improvement in the seller revenue and the sum of buyer utilities by adding the Market-Maker contract.

This illustrates the fact that adding the Market-Maker contract to the existing posted price mechanism allows to attain a significant Pareto improvement for both the seller and the buyers.

In conclusion, we saw that the Market-Maker contract improves both the seller revenue and the sum of buyer utilities. It allows to reduce both the price and allocation uncertainties for risk averse buyers who are willing to pay a premium over the expected auction clearing price. In addition, when the buyers have heterogeneous risk aversions, the Market-Maker contract allows the buyers with a higher risk aversion to secure a higher probability of being allocated.

Bibliography

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

[2] Aydın Alptekinoğlu and John H. Semple. The exponomial choice model: A new alternative for assortment and price optimization. *Operations Research*, 64(1):79–93, 2016.

[3] Ali Aouad, Vivek Farias, Retsef Levi, and Danny Segev. The approximability of assortment optimization under ranking preferences. *Available at SSRN 2612947*, 2015.

[4] Pranjal Awasthi, Avrim Blum, Or Sheffet, and Aravindan Vijayaraghavan. Learning mixtures of ranking models. In *Advances in Neural Information Processing Systems*, pages 2609–2617, 2014.

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

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

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

[8] Moshe Ben-Akiva and Steven Lerman. *Discrete Choice Analysis: Theory and Application to Travel Demand*, volume 9. MIT press, 1985.

[9] Dimitris Bertsimas and Velibor V. Mišic. Data-driven assortment optimization. Technical report, Working paper, MIT Sloan School, 2015.

[10] Jose Blanchet, Guillermo Gallego, and Vineet Goyal. A markov chain approximation to choice modeling. *Operations research*, 64(4):886–905, 2016.

[11] Axel Börsch-Supan. On the compatibility of nested logit models with utility maximization. *Journal of Econometrics*, 43(3):373–388, 1990.

[12] Tamer Boyacı and Özalp Özer. Information acquisition for capacity planning via pricing and advance selling: When to stop and act? *Operations Research*, 58(5):1328–1349, 2010.

[13] Graham Brightwell and Peter Winkler. Counting linear extensions is #P-complete. In *STOC '91 Proceedings of the twenty-third annual ACM Symposium on Theory of Computing*, pages 175–181, 1991.

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

[15] Gérard P. Cachon. The allocation of inventory risk in a supply chain: Push, pull, and advance-purchase discount contracts. *Management Science*, 50(2):222–238, 2004.

[16] René Caldentey and Gustavo Vulcano. Online auction and list price revenue management. *Management Science*, 53(5):795–813, 2007.

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

[18] L. Elisa Celis, Gregory Lewis, Markus Mobius, and Hamid Nazerzadeh. Buy-it-now or take-a-chance: Price discrimination through randomized auctions. *Management Science*, 60(12):2927–2948, 2014.

[19] Soo-Haeng Cho and Christopher S. Tang. Advance selling in a supply chain under uncertain supply and demand. *Manufacturing & Service Operations Management*, 15(2):305–319, 2013.

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

[21] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.

[22] Dwork Cynthia, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In ACM, editor, *Proceedings of the 10th international conference on World Wide Web*, pages 613–622, 2001.

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

[24] James Davis, Guillermo Gallego, and Huseyin Topaloglu. Assortment optimization under variants of the nested logit model. *Operations Research*, 62(2):250–273, 2014.

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

[26] Vivek Farias, Srikanth Jagabathula, and Devavrat Shah. A nonparametric approach to modeling choice with limited data. *Management Science*, 59(2):305–322, 2013.

[27] Vivek Farias, Srikanth Jagabathula, and Devavrat Shah. Building optimized and hyperlocal product assortments: A nonparametric choice approach. 2017.

[28] Scott Fay and Jinhong Xie. The economics of buyer uncertainty: Advance selling vs. probabilistic selling. *Marketing Science*, 29(6):1040–1057, 2010.

[29] Uriel Feige, Shafi Goldwasser, Laszlo Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM (JACM)*, 43(2):268–292, 1996.

[30] Jacob B. Feldman and Huseyin Topaloglu. Revenue management under the markov chain choice model, 2014.

[31] Jacob B. Feldman and Huseyin Topaloglu. Capacity constraints across nests in assortment optimization under the nested logit model. *Operations Research*, 63(4):812–822, 2015.

[32] Alan M. Frieze and M.R.B. Clarke. Approximation algorithms for the m-dimensional 0–1 knapsack problem: Worst-case and probabilistic analyses. *European Journal of Operational Research*, 15(1):100–109, 1984.

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

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

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

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

[37] Darrell Hoy, Nicole Immorlica, and Brendan Lucier. On-demand or spot? selling the cloud to risk-averse customers. In *International Conference on Web and Internet Economics*, pages 73–86. Springer, 2016.

[38] John Hull. *Introduction to futures and options markets*. Englewood Cliffs, NJ: Prentice Hall, 1991.

[39] Srikanth Jagabathula and Paat Rusmevichientong. A nonparametric joint assortment and price choice model. *Management Science*, 2016.

[40] Robert A. Jarrow and George S. Oldfield. Forward contracts and futures contracts. *Journal of Financial Economics*, 9(4):373–382, 1981.

[41] Leland L. Johnson. The theory of hedging and speculation in commodity futures. *The Review of Economic Studies*, 27(3):139–151, 1960.

[42] Toshihiro Kamishima, Hideto Kazawa, and Shotaro Akaho. Supervised ordering-an empirical survey. In *Data Mining, Fifth IEEE International Conference on*, pages 4–pp. IEEE, 2005.

[43] A. Gürhan Kök and Marshall L. Fisher. Demand estimation and assortment optimization under substitution: Methodology and application. *Operations Research*, 55(6):1001–1021, 2007.

[44] A.G. Kok and M.L. Fisher. Demand estimation and assortment optimization under substitution: Methodology and application. *Operations Research*, 55(6):1001–1021, 2007.

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

[46] Guy Lebanon and Yi Mao. Non-parametric modeling of partially ranked data. *Journal of Machine Learning Research*, 9:2401–2429, 2008.

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

[48] Guang Li, Paat Rusmevichientong, and Huseyin Topaloglu. The d-level nested logit model: Assortment and price optimization problems. *Operations Research*, 62(2):325–342, 2015.

[49] Tyler Lu and Craig Boutilier. Learning mallows models with pairwise preferences. In *Proceedings of the 28th International Conference on Machine Learning*, pages 145–152, 2011.

[50] R. Duncan Luce. *Individual choice behavior: A theoretical analysis*. Wiley, 1959.

[51] Colin L. Mallows. Non-null ranking models. i. *Biometrika*, pages 114–130, 1957.

[52] John Marden. *Analyzing and Modeling Rank Data*. Chapman and Hall, 1995.

[53] Daniel McFadden. Modeling the choice of residential location. *Transportation Research Record*, 1978.

[54] Daniel McFadden. Econometric models for probabilistic choice among products. *Journal of Business*, pages 13–29, 1980.

[55] Daniel McFadden and Kenneth Train. Mixed mnl models for discrete response. *Journal of applied Econometrics*, 15(5):447–470, 2000.

[56] Vahab Mirrokni and Hamid Nazerzadeh. Deals or no deals: Contract design for online advertising. 2015.

[57] Shashi Mittal and Andreas S Schulz. A general framework for designing approximation schemes for combinatorial optimization problems with many objectives combined into one. *Operations Research*, 61(2):386–397, 2013.

[58] Thomas Murphy and Donal Martin. Mixtures of distance-based models for ranking data. *Computational Statistics & Data Analysis*, 41(3):645–655, 2003.

[59] George L. Nemhauser and Laurence A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research*, 3(3):177–188, 1978.

[60] Christos H Papadimitriou and Mihalis Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *In Proceedings of 41st Annual IEEE Symposium on Foundations of Computer Science*, pages 86–92, 2000.

[61] Roy Pereira. The MakeGood, `http://www.the-makegood.com/2013/01/21/defining-programmatic-guaranteed-the-problems-with-rtb-for-premium-inventory/`, 2013.

[62] R.L. Plackett. The analysis of permutations. *Applied Statistics*, pages 193–202, 1975.

[63] Ashutosh Prasad, Kathryn E. Stecke, and Xuying Zhao. Advance selling by a newsvendor retailer. *Production and Operations Management*, 20(1):129–142, 2011.

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

[65] Paat Rusmevichientong, Zuo-Jun Max Shen, and D.B. Shmoys. A ptas for capacitated sum-of-ratios optimization. *Operations Research Letters*, 37(4):230–238, 2009.

[66] Paat Rusmevichientong, David Shmoys, Chaoxu Tong, and Huseyin Topaloglu. Assortment optimization under the multinomial logit model with random choice parameters. *Production and Operations Management*, 2014.

[67] Paat Rusmevichientong and Huseyin Topaloglu. Robust assortment optimization in revenue management under the multinomial logit choice model. *Operations Research*, 60(4):865–882, 2012.

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

[69] Steven M. Shugan and Jinhong Xie. Advance pricing of services and other implications of separating purchase and consumption. *Journal of Service Research*, 2(3):227–239, 2000.

[70] Steven M. Shugan and Jinhong Xie. Advance-selling as a competitive marketing tool. *International Journal of Research in Marketing*, 22(3):351–373, 2005.

[71] Kalyan Talluri and Garret Van Ryzin. Revenue management under a general discrete choice model of consumer behavior. *Management Science*, 50(1):15–33, 2004.

[72] Kenneth Train. *Discrete choice methods with simulation*. Cambridge University Press, New York, NY, 2003.

[73] Garret van Ryzin and Gustavo Vulcano. A market discovery algorithm to estimate a general class of nonparametric choice models. *Management Science*, 61(2):281–300, 2014.

[74] Jun Wang and Bowei Chen. Selling futures online advertising slots via option contracts. In *Proceedings of the 21st International Conference on World Wide Web*, pages 627–628. ACM, 2012.

[75] Huw Williams. On the formation of travel demand models and economic evaluation measures of user benefit. *Environment and Planning A*, 3(9):285–344, 1977.

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

Appendices

Appendix A

Near optimal algorithms for capacity constrained assortment under random utility models

A.1 FPTAS for mMNL-Capa

High-level description. Let p (resp. P) be the minimum (resp. maximum) revenue and u (resp. U) be the minimum (resp. maximum) value of the utility parameters over all segments. We assume wlog. that $u_{j,k} > 0$ for all j, k. Otherwise, we can replace $u_{j,k}$ by $\hat{u}_{j,k} = \epsilon up/(nR)$ for all j, k such that $u_{j,k} = 0$ where $u = \min\{u_{i,k} \mid u_{i,k} > 0\}$. This only changes the objective function by a factor of $(1+\epsilon)$ (see Appendix A.2). For a given $\epsilon > 0$, we use the following set of guesses.

$$\Gamma_{\epsilon,K} = (\Gamma_\epsilon)^K \quad \text{and} \quad \Delta_{\epsilon,K} = (\Delta_\epsilon)^K,$$

where

$$\Gamma_\epsilon = \{pu(1+\epsilon)^\ell, \ell = 0, \ldots, L_1\} \text{ and } \Delta_\epsilon = \{u(1+\epsilon)^\ell, \ell = 0, \ldots, L_2\}, \tag{A.1}$$

and $L_1 = O\left(\log\left(nPU/p\right)/\epsilon\right)$ and $L_2 = O\left(\log\left((n+1)U/p\right)/\epsilon\right)$. Note that for constant K, the number of guesses is polynomial in the input size and $1/\epsilon$. For a given guess $(\boldsymbol{h}, \boldsymbol{g}) \in \Gamma_{\epsilon,K} \times \Delta_{\epsilon,K}$, we discretize the coefficients as follows,

$$\tilde{p}_{i,k} = \left\lfloor \frac{p_i u_{i,k}}{\epsilon h_k/n} \right\rfloor \quad \text{and} \quad \tilde{u}_{i,k} = \left\lceil \frac{u_{i,k}}{\epsilon g_k/(n+1)} \right\rceil. \tag{A.2}$$

We use a dynamic program to find a feasible assortment S such that for all $k \in [K]$

$$\sum_{j\in S} p_j u_{j,k} \geq h_k \quad \text{and} \quad \sum_{j\in S_+} u_{j,k} \leq g_k. \tag{A.3}$$

Let us now present the dynamic program. Let $I = \lfloor n/\epsilon \rfloor - n$ and $J = \lceil (n+1)/\epsilon \rceil + (n+1)$. For each $(\boldsymbol{i}, \boldsymbol{j}, \ell) \in [I]^K \times [J]^K \times [n]$, let $F(\boldsymbol{i}, \boldsymbol{j}, \ell)$ be the minimum weight of any subset $S \subseteq \{1, \dots, \ell\}$ such that for all $k \in [K]$,

$$\sum_{s \in S} \tilde{p}_{s,k} \geq i_k \quad \text{and} \quad \sum_{s \in S_+} \tilde{u}_{s,k} \leq j_k.$$

We can compute $F(\boldsymbol{i}, \boldsymbol{j}, \ell)$ for $(\boldsymbol{i}, \boldsymbol{j}, \ell) \in [I]^K \times [J]^K \times [n]$ using the following recursion.

$$F(\boldsymbol{i}, \boldsymbol{j}, 1) = \begin{cases} w_1 & \text{if } \mathbf{0} \leq \boldsymbol{i} \leq \tilde{\boldsymbol{p}}_1 \text{ and } \boldsymbol{j} \geq \tilde{\boldsymbol{u}}_0 + \tilde{\boldsymbol{u}}_1 \\ 0 & \text{if } \boldsymbol{i} \leq \mathbf{0} \text{ and } \boldsymbol{j} \geq \tilde{\boldsymbol{u}}_0 \\ \infty & \text{otherwise} \end{cases} \tag{A.4}$$

$$F(\boldsymbol{i}, \boldsymbol{j}, \ell+1) = \min\{F(\boldsymbol{i}, \boldsymbol{j}, \ell), w_{\ell+1} + F(\boldsymbol{i} - \tilde{\boldsymbol{p}}_{\ell+1}, \boldsymbol{j} - \tilde{\boldsymbol{u}}_{\ell+1}, \ell)\}$$

Let $\boldsymbol{I}$ (resp. $\boldsymbol{J}$) be the vector with all components being I (resp. J). In order to show that (A.4) correctly finds a subset satisfying (A.3), we have the following lemma.

Lemma A.1. *For any guess $\boldsymbol{h}$, $\boldsymbol{g}$, if there exists a feasible S such that (A.3) is satisfied, then $F(\boldsymbol{I}, \boldsymbol{J}, n) \leq W$. Moreover, if $F(\boldsymbol{I}, \boldsymbol{J}, n) \leq W$, then the DP finds a subset $\tilde{S}$ such that for all $k \in [K]$,*

$$\sum_{j \in S} p_{j,k} u_{j,k} \geq h_k(1 - 2\epsilon) \quad \textit{and} \quad \sum_{j \in S_+} u_{j,k} \leq g_k(1 + 2\epsilon).$$

Proof. Consider S satisfying (A.3) for given guesses $\boldsymbol{h}, \boldsymbol{g}$. Scaling the inequalities yields for all $k \in [K]$

$$\sum_{j \in S} \frac{p_j u_{j,k}}{\epsilon h_k / n} \geq \frac{h_k}{\epsilon h_k / n} \quad \text{and} \quad \sum_{j \in S_+} \frac{u_{j,k}}{\epsilon g_k/(n+1)} \leq \frac{g_k}{\epsilon g_k/(n+1)}.$$

Rounding down and up the previous inequalities gives for all k

$$\sum_{j \in S} \tilde{p}_{j,k} \geq \lfloor n/\epsilon \rfloor - n = I \quad \text{and} \quad \sum_{j \in S_+} \tilde{u}_{j,k} \leq \left\lceil \frac{(n+1)}{\epsilon} \right\rceil + (n+1) = J,$$

which implies that $F(\boldsymbol{I}, \boldsymbol{J}, n) \leq W$.

Conversely, suppose $F(\boldsymbol{I}, \boldsymbol{J}, n) \leq W$ and let $\tilde{S}$ be the corresponding subset. We have

$$\sum_{j \in \tilde{S}} p_j u_{j,k} \geq I \frac{\epsilon h_k}{n} \geq h_k(1-2\epsilon) \quad \text{and} \quad \sum_{j \in \tilde{S}_+} u_{j,k} \leq J \frac{\epsilon g_k}{n+1} \leq g_k(1+2\epsilon).$$

□

We can now present the FPTAS for mMNL-Capa.

Algorithm 15 FPTAS for mMNL-Capa

```
1: procedure FPTAS(ϵ)
2:     for (h, g) ∈ Γ_{ϵ,K} × Δ_{ϵ,K} do
3:         Compute discretization of coefficient r̃_{i,k} and ũ_{i,k} using (A.2)
4:         Compute F(i, j, ℓ) for all (i, j, ℓ) ∈ [I]^K × [J]^K × [n] using (A.4)
5:         If F(I, J, n) ≤ W, then let S̃_{h,g} be a the corresponding subset
6:     end for
7:     return S that maximizes the expected revenue over {S̃_{h,g}, (h, g) ∈ Γ_{ϵ,K} × Δ_{ϵ,K}}
8: end procedure
```

Theorem A.1. *Algorithm 15 returns an $(1-\epsilon)$-optimal solution to mMNL-Capa. Moreover, the running time is $O\left(\log(nPU)^K \log(nU)^K n^{2K+1}/\epsilon^{4K}\right)$.*

Proof. Let S^* be the optimal solution to mMNL-Capa and $(\hat{\boldsymbol{\ell}}_1, \hat{\boldsymbol{\ell}}_2)$ such that for all $k \in [K]$

$$pu\,(1+\epsilon)^{\hat{\ell}_{1,k}} \leq \sum_{i \in S^*} p_i u_{i,k} \leq ru\,(1+\epsilon)^{\hat{\ell}_{1,k}+1} \quad \text{and} \quad u\,(1+\epsilon)^{\hat{\ell}_{2,k}-1} \leq \sum_{i \in S^*_+} u_{i,k} \leq u\,(1+\epsilon)^{\hat{\ell}_{2,k}}.$$

From Lemma A.1, we know that for $(\boldsymbol{h}, \boldsymbol{g}) = (pu\,(1+\epsilon)^{\hat{\boldsymbol{\ell}}_1}, u\,(1+\epsilon)^{\hat{\boldsymbol{\ell}}_2})$, Algorithm 15 returns $\tilde{S}$ such that for all $k \in [K]$

$$\sum_{i \in \tilde{S}} p_i u_{i,k} \geq pu\,(1+\epsilon)^{\hat{\ell}_{1,k}}(1-2\epsilon) \quad \text{and} \quad \sum_{i \in \tilde{S}_+} u_{i,k} \leq u\,(1+\epsilon)^{\hat{\ell}_{2,k}}(1+2\epsilon).$$

Consequently,

$$f(\tilde{S}) = \sum_{k=1}^{K} \theta_k \frac{\sum_{i \in \tilde{S}} p_i u_{i,k}}{\sum_{i \in \tilde{S}_+} u_{i,k}} \geq \frac{1-2\epsilon}{1+2\epsilon} f(S^*) \geq (1-4\epsilon) f(S^*).$$

Running Time. We try $L_1^K \cdot L_2^K$ guesses for the numerators and denominators values, $(\boldsymbol{h}, \boldsymbol{g})$, of the optimal solution. For each guess, we formulate a dynamic program with $O\left(n^{2K+1}/\epsilon^{2K}\right)$ states. Therefore, the running time of Algorithm 15 is $O\left(L_1^K L_2^K n^3/\epsilon^2\right) = O\left(\log(nPU)\log(nU)n^{2K+1}/\epsilon^{4K}\right)$ which is polynomial in input size and $1/\epsilon$. □

A.2 Assumption of $u_{i,k} > 0$ in mMNL-Capa

We show that wlog. we can assume $u_{i,k} > 0$ for all $i \in [n], k \in [K]$ in the mMNL-Capa problem. Let $u = \min\{u_{i,k} \mid u_{i,k} > 0\}$. Suppose $u_{j,k} = 0$ for some j, k. Then, consider the following modified utility parameters for all j, k.

$$\hat{u}_{j,k} = \begin{cases} \epsilon up/(nP) & \text{if } u_{j,k} = 0 \\ u_{j,k} & \text{otherwise} \end{cases}$$

We show that replacing $u_{j,k}$ by $\hat{u}_{j,k}$ in mMNL-Capa changes the expected revenue of any subset by a factor of $[1-\epsilon, 1+\epsilon]$. In particular, for any $x \in \{0,1\}^n$, for all $k \in [K]$,

$$\sum_{j=1}^{n} p_j u_{j,k} x_j \le \left(\sum_{j=1}^{n} p_j \hat{u}_{j,k} x_j\right) \le \sum_{j=1}^{n} r_j u_{j,k} x_j + \frac{r_{j,k}}{R} \cdot \epsilon pu \le (1+\epsilon) \cdot \sum_{j=1}^{n} p_j u_{j,k} x_j.$$

Similarly for all $k \in [K]$,

$$u_{0,k} + \sum_{j=1}^{n} u_{j,k} x_j \le \left(\hat{u}_{0,k} + \sum_{j=1}^{n} \hat{u}_{j,k} x_j\right) \le (1+\epsilon) \cdot \left(u_{0,k} + \sum_{j=1}^{n} u_{j,k} x_j\right).$$

Therefore, for each rational terms in the expression for the expected revenue, both the numerator and denominator increase by a factor of at most $(1+\epsilon)$. Let z^* be the optimal value of mMNL-Capa and $\hat{z}$ be the optimal value of the modified problem with parameters, $\hat{u}_{j,k}$. Using the previous set of inequalities, we have $(1-\epsilon)\hat{z} \le z^* \le (1+\epsilon)\hat{z}$ and we can equivalently approximate the modified problem.

A.3 Proof of Theorem 2.8

As in Theorem 2.7, we prove this by a reduction from the independent set problem where we are given an undirected graph $G = (V, E)$ and the goal is to find a maximum cardinality subset of vertices that are independent. Let $V = \{v_1, \ldots, v_n\}$.

We construct an instance of MMNL-Assort similar to the proof of Theorem 2.7. We have one product and one MNL segment corresponding to each vertex in G. Therefore, $n = K = |V|$ and we consider the following utility parameters:

$$u_{j,k} = \begin{cases} 1 & \text{if } j = k \text{ or } j = 0 \\ n^3 & \text{if } (v_j, v_k) \in E \text{ and } j < k \\ 0 & \text{otherwise} \end{cases} \tag{A.5}$$

$$p_i = n^{3(i-1)},\ i \in [n]$$

$$\theta_k = \frac{\theta}{n^{3(k-1)}},\ k \in [n]$$

where $\theta \in [1/2, 1]$ is an appropriate normalizing constant.

Consider an optimal independent set, $\mathcal{I}^*$ of size t^*. Consider the following assortment

$$S = \{j \mid v_j \in \mathcal{I}^*\}.$$

It is easy to observe that the expected revenue of S is exactly $\theta t^*/2$.

Conversely, consider an optimal fractional assortment $\boldsymbol{x}^* \in [0,1]^n$ with revenue z^*. Then we show that there exists an independent set of size $\lfloor 2z^*/\theta \rfloor$. Let $\epsilon = 1/4n$. Consider a modified solution $\tilde{\boldsymbol{x}}$ defined as follows. For all $k \in [K]$,

$$\tilde{x}_k = \begin{cases} 0 & \text{if } x_k^* \leq \epsilon \\ x_k^* & \text{otherwise.} \end{cases}$$

Also, let $\tilde{z}$ be the revenue associated with solution $\tilde{\boldsymbol{x}}$. It is easy to observe that the revenue of each nest only decreases by at most $\theta\epsilon$. Consequently,

$$\tilde{z} \geq z^* - n\theta\epsilon \geq z^* - \frac{\theta}{4} \geq \frac{z^*}{2},$$

where the last inequality follows as $z^* \geq \theta/2$. For any $k \in [K]$, let

$$\tilde{z}_k = \theta_k \cdot \frac{\sum_{j=1}^n p_j u_{j,k} \tilde{x}_k}{u_{0,k} + \sum_{j=1}^n u_{j,k} \tilde{x}_j}, \quad \text{and } \tilde{z} = \sum_{k=1}^K \tilde{z}_k.$$

We show that for any $k \in [K]$, $\tilde{z}_k \geq \theta/(5n)$ or $\tilde{z}_k \leq \theta/n^2$. Let

$$N(k) = \{j \mid j < k,\ (v_j, v_k) \in E,\ \tilde{x}_j \geq \epsilon\}.$$

Case 1 ($N(k) = \emptyset$): In this case

$$\tilde{z}_k = \frac{\theta \tilde{x}_k}{1 + \tilde{x}_k} \leq \frac{\theta}{2}.$$

Therefore, if $\tilde{x}_k < \epsilon$, it implies $\tilde{x}_k = 0$ (by construction) and $\tilde{z}_k = 0$.

Case 2 ($N(k) \neq \emptyset$): In this case,

$$\begin{aligned}
\tilde{z}_k &\leq \frac{\theta}{n^{3(k-1)}(1 + n^3 \sum_{j \in N(k)} \tilde{x}_j)} \left(n^{3(k-1)} + n^3 \sum_{j \in N(k)} n^{3(j-1)} \tilde{x}_j \right) \\
&\leq \frac{\theta}{n^{3(k-1)}(2 + n^3 \epsilon)} \left(n^{3(k-1)} + n^3 \sum_{j=1}^{k-1} n^{3(j-1)} \right) \\
&\leq \frac{2\theta}{n^2},
\end{aligned}$$

where the second inequality follows as $N(k) \neq \emptyset$ and there exists $j \in N(k)$ such that $\tilde{x}_j \geq \epsilon$. Now, we construct an independent set, $\mathcal{I}$ as follows.

$$\mathcal{I} = \{v_k \in V \mid \tilde{x}_k \geq \epsilon,\ N(k) = \emptyset\}.$$

Since for all k such that $v_k \in \mathcal{I}$, $N(k) = \emptyset$, we know that $\mathcal{I}$ is an independent set (using an argument similar to proof of Theorem 2.7). From the above case analysis, we know

$$\sum_{k: v_k \in \mathcal{I}} \tilde{z}_k \leq \tilde{z} \leq \sum_{k: v_k \in \mathcal{I}} \tilde{z}_k + \frac{2\theta}{n}, \tag{A.6}$$

where the second inequality follows from the fact that $\tilde{z}_k \leq 2\theta/n^2$ if $v_k \notin \mathcal{I}$. We also know that $\tilde{z} \geq z^*/2 \geq \theta/4$ and $\tilde{z}_k \leq \theta/2$ for all $k : v_k \in \mathcal{I}$. Therefore,

$$z^* \leq 2\tilde{z} \leq 2 \left(\sum_{k: v_k \in \mathcal{I}} \frac{\theta}{2} \right) + \frac{4\theta}{n} \leq |\mathcal{I}| \cdot \theta + \frac{8z^*}{n},$$

which implies

$$|\mathcal{I}| \geq \frac{\left(1 - \frac{8}{n}\right)}{\theta} \cdot z^* \geq \frac{1}{2\theta} \cdot z^*.$$

Therefore,

$$\frac{1}{2\theta} \cdot z^* \leq t^* \leq \frac{2}{\theta} \cdot z^*.$$

Recall that θ is a constant in $[1/2, 1]$. Therefore, an α-approximation for the continuous relaxation of MMNL-Assort implies an $O(\alpha)$-approximation for the maximum independent set problem. Since the maximum independent set is hard to approximation within a factor better than $O(1/n^{1-\delta})$ (where $|V| = n = K$) for any constant $\delta > 0$ (see [29]), so must be the continuous relaxation of MMNL-Assort. This concludes the proof.

Appendix B

Approximation algorithms for assortment optimization problems under a Markov chain based choice model

B.1 Proof of Theorem 3.2

In an instance of the assortment optimization problem over the distribution over rankings model, we are given a collection of products $\mathcal{N} = \{1, \ldots, n\}$ with prices $p_1 \leq \cdots \leq p_n$, respectively. In addition, we are given an arbitrary (known) distribution on K preference lists, $L_1, \ldots, L_K$, each of which specifies a subset of the products listed in decreasing order of preference. A customer with a given preference list selects the most preferred product that is offered (possibly the no-purchase option) according to his/her list. The goal is to find an assortment such that the expected revenue is maximized. [**?**] show that unconstrained assortment optimization over the distribution over permutations model is hard to approximate within factor $O(n^{1-\epsilon})$ for any fixed $\epsilon > 0$ even for the case where the number of preference lists is equal to the number of items, i.e., $K = n$.

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

Therefore, the set of states is:

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

The transition probabilities between these states are given by:

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

In other words, for each list there is a directed path (with transition probabilities 1) over its corresponding states in decreasing order of preference, ending at the no-purchase option. This is illustrated in Figure B.1. Finally, the arrival rates are defined by

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

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

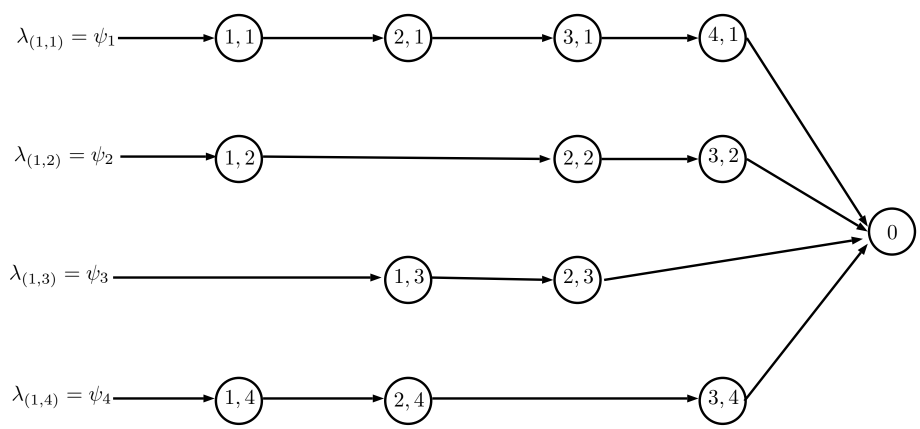


Figure B.1: Sketch of our construction for an instance on 4 items, where $L_1 = (1 \succ 2 \succ 3 \succ 4)$, $L_2 = (1 \succ 3 \succ 4)$, $L_3 = (2 \succ 3)$, and $L_4 = (1 \succ 2 \succ 4)$. Note, for example, that the state $(2,2)$ corresponds to the second item of L_2, but actually corresponds to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

In order to obtain a one-to-one correspondence between the solutions to $\mathcal{I}$ and $\mathcal{M}(\mathcal{I})$, it remains to ensure that, when item i is offered in $\mathcal{I}$, all of its corresponding copies (appearing in the same column) are offered in $\mathcal{M}(\mathcal{I})$, and vice versa. This restriction can be captured by the constraints $x_{(j,i)} = x_{(k,\ell)}$, for every $i, \ell \in \{1, \ldots, n\}$ such that $j \leq |L_i|, k \leq |L_\ell|$ and such that the j^{th} item in L_i is the k^{th} item in L_ℓ. This way, we guarantee that each column is either completely picked or completely unpicked in the instance $\mathcal{M}(\mathcal{I})$. The resulting set of inequalities specifies a constraint matrix with a single appearance of $+1$ and -1 in each row, where all other entries are 0. Such matrices are well-known to be totally-unimodular (see, for example, [68]).

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

B.2 Proof of Lemma 3.2

This result is an immediate corollary of the following (more general) claim: Let S^g be the solution returned by Algorithm 4, and let S be any subset of states. Then,

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

To prove this claim, let g be the first item selected by Algorithm 4, which necessarily exists as long as there is an item i with $p_i > 0$. Then, by definition of the greedy algorithm, we have $R(\{g\}) \geq R(\{i\})$ for every item $i \in S$. Therefore,

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

where the last inequality follows from the sublinearity of the revenue function (Lemma 3.9).

B.3 Proof of Lemma 3.4

Let S^{gu} be the set of states selected by Algorithm 5. Note that for every $i \in S^{gu}$, we have that $\mathbb{P}(i \prec S^{gu}_{+} \backslash \{i\}) \geq \mathbb{P}(i \prec U^{*}_{+} \backslash \{i\})$ since S^{gu} is a subset of U^{*}. Thus,

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

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

B.4 Proof of Lemma 3.6

It suffices to verify that $(p_i^{S_1})^{S_2} = p_i^{S_1 \cup S_2}$ for all S_1,S_2 and $i \notin S_1 \cup S_2$, as the above identity clearly hold for the transition matrix updates. We have

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

Using the definition of the updated prices,

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

We can now combine A and C,

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

Putting everything together, we get

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

B.5 Application of Algorithm 6 to MNL

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

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

making the expected revenue

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

[10] prove that the MNL choice model is a special case of the Markov chain model. More precisely, when $\rho_{ij} = u_j$ for all j and $\lambda_i = u_i$ for all i, the choice probabilities

of the two models are identical. In this special case, our local ratio updates can be written as

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

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

B.6 FPTAS for MC-Capa under rank one assumption

Recall that MC-Capa can be formulated as

$$\max_{S \subseteq [n]} \left\{ \sum_{i \in S} p_i \left(\lambda_i + v_i \left(\sum_{j \notin S} u_j \lambda_j \right) \left(\sum_{m=0}^{\infty} \left(\sum_{j \notin S} u_j v_j \right)^m \right) \right) \;\middle|\; \sum_{j \in S} w_j \leq W \right\}.$$

Without loss of generality, we assume that $u_0, \lambda_0, v_0 > 0$. As before, let p (resp. P) be the minimum (resp. maximum) revenue and u, v (resp. U, V) be the minimum (resp. maximum) MNL parameter. We can assume wlog. that $p, u, v > 0$; otherwise, we can clearly remove the corresponding item from our collection and continue. For any given $\epsilon > 0$, we use the following set of guesses.

$$\Gamma_\epsilon = \{\lambda_0 u (1+\epsilon)^\ell, \ell = 0, \ldots, L_1\},$$

$$\Delta_\epsilon = \{uv(1+\epsilon)^\ell, \ell = 0, \ldots, L_2\},$$

$$\Lambda_\epsilon = \{\lambda_0 pvu(1+\epsilon)^\ell, \ell = 0, \ldots, L_3\},$$

where $L_1 = O(\log(nU/\lambda_0 u)/\epsilon)$, $L_2 = O(\log(nUV/uv)/\epsilon)$ and $L_3 = O(\log(n^2 PVU/\lambda_0 pvu)/\epsilon)$. The number of guesses is polynomial in the input size and $1/\epsilon$. For a given guess $h \in \Gamma_\epsilon, g \in \Delta_\epsilon$ and $t \in \Lambda_\epsilon$, we try to find a feasible assortment S with

$$\sum_{j \in S} p_i \left(\lambda_i + v_i \frac{h}{1-g}\right) \geq t, \sum_{j \notin S} u_j \lambda_j \geq h \text{ and } \sum_{j \notin S} u_j v_j \geq g, \tag{B.1}$$

using a dynamic program. In particular, we consider the following discretized values,

$$\tilde{p}_j = \left\lfloor \frac{p_j(\lambda_j + v_j(h/1-g))}{\epsilon t/n} \right\rfloor, \; \tilde{u}_j = \left\lfloor \frac{u_j \lambda_j}{\epsilon h/n} \right\rfloor \text{ and } \tilde{v}_j = \left\lfloor \frac{u_j v_j}{\epsilon g/n} \right\rfloor, \forall j. \tag{B.2}$$

Let $I = \lfloor n/\epsilon \rfloor - n, \; J = \lfloor (n+1)/\epsilon \rfloor - (n+1)$. We can now present our dynamic program. For each $(i,j,k,\ell) \in [I] \times [J] \times [J] \times [n]$, let $F(i,j,k,\ell)$ be the minimum weight of any subset $S \subseteq \{1,\ldots,\ell\}$ such that

$$\sum_{s \in S} \tilde{p}_s \geq i, \sum_{s \notin S} \tilde{u}_s \geq j \text{ and } \sum_{s \notin S} \tilde{v}_s \geq k. \tag{B.3}$$

We compute $F(i,j,k,\ell)$ for $(i,j,k,\ell) \in [I]\times[J]\times[J]\times[n]$ using the following recursion.

$$F(i,j,1) = \begin{cases} w_1 & \text{if } 0 \leq i \leq \tilde{p}_1, j \leq \tilde{u}_0, \text{ and } k \leq \tilde{v}_0 \\ 0 & \text{if } i \leq 0 \text{ and } j \geq \tilde{u}_0 \\ \infty & \text{otherwise} \end{cases}$$
$$F(i,j,k,\ell+1) = \min\{F(i,j-\tilde{u}_{\ell+1},k-\tilde{v}_{\ell+1},\ell), w_{\ell+1} + F(i-\tilde{p}_{\ell+1},j,k,\ell)\} \tag{B.4}$$

Using this dynamic program, we construct a set of candidate assortments $S_{h,g,\ell}$ for all guesses $(h,g,t) \in \Gamma_\epsilon \times \Delta_\epsilon \times \Lambda_\epsilon$. Algorithm 16 details the procedure to construct the set of candidate assortments.

Algorithm 16 Construct Candidate Assortments

1: For $(h,g,t) \in \Gamma_\epsilon \times \Delta_\epsilon \times \Lambda_\epsilon$,
  (a) Compute discretization of coefficients $\tilde{p}_i$, $\tilde{u}_i$ and $\tilde{v}_i$ using (B.2).
  (b) Compute $F(i,j,k,\ell)$ for all $(i,j,k,\ell) \in [I] \times [J] \times [K] \times [n]$ using (B.4).
  (c) Let $S_{h,g,t}$ be the subset corresponding to $F(I,J,J,n)$.
2: Return $\mathcal{A} = \cup_{(h,g,t)\in\Gamma_\epsilon\times\Delta_\epsilon\times\Lambda_\epsilon} S_{h,g,t}$.

Let us show that Algorithm 16 correctly finds a subset satisfying (B.1). In particular, we have the following lemma.

Lemma B.1. *Let $\mathcal{A}$ be the set of candidate assortment returned by Algorithm 16. For any guess $(h, g, t) \in \Gamma_\epsilon \times \Delta_\epsilon \times \Lambda_\epsilon$, if there exists S such that $W(S) \leq W$ and* (B.1) *is satisfied, then $W(S_{h,g,t}) \leq W$. Moreover, $S_{h,g,t}$ satisfies* (B.1) *approximately, i.e.*

$$\sum_{j \in S_{h,g,t}} p_j \left(\lambda_i + v_i \frac{h}{1-g}\right) \geq t(1-2\epsilon), \quad \sum_{j \notin S_{h,g,t}} u_j \lambda_j \geq h(1-2\epsilon) \quad \textit{and} \quad \sum_{j \notin S_{h,g,t}} u_j v_j \geq g(1-2\epsilon).$$

Proof. Consider S satisfying (B.1) for given guesses h, g, t. Scaling the three inequalities yield

$$\sum_{j \in S} \frac{1}{\epsilon t/n} p_j \left(\lambda_i + v_i \frac{h}{1-g}\right) \geq \frac{t}{\epsilon t/n},$$
$$\sum_{j \notin S} \frac{u_j \lambda_j}{\epsilon h/(n+1)} \geq \frac{h}{\epsilon h/(n+1)},$$
$$\sum_{j \notin S} \frac{u_j v_j}{\epsilon g/(n+1)} \geq \frac{g}{\epsilon g/(n+1)}.$$

Rounding down and up the previous inequalities gives

$$\sum_{j \in S} \tilde{p}_j \geq I, \ \sum_{j \notin S} \tilde{u}_j \geq J \ \text{ and } \ \sum_{j \notin S} \tilde{v}_j \geq J,$$

which implies that $F(I, J, J, n) \leq W$. Moreover, let $S_{h,g,t}$ be the corresponding subset. We have

$$\sum_{j \in S_{h,g,t}} p_j \left(\lambda_i + v_i \frac{h}{1-g}\right) \geq I \frac{\epsilon t}{n} \geq t(1-2\epsilon),$$
$$\sum_{j \notin S_{h,g,t}} u_j \lambda_j \geq J \frac{\epsilon h}{n+1} \geq h(1-2\epsilon),$$
$$\sum_{j \notin S_{h,g,t}} u_j v_j \geq J \frac{\epsilon g}{n+1} \geq g(1-2\epsilon).$$

□

Now that we have constructed a set of candidate assortment, the second part of the algorithm consist of returning the best possible feasible assortment. We can therefore present in Algorithm 17 a complete description of the algorithm.

Algorithm 17 FPTAS for MC-Capa

1: Construct a set of candidate assortment $\mathcal{A}$ using Algorithm 16.
2: Return the best feasible solution to MC-Capa from $\mathcal{A}$.

Theorem B.1. *Algorithm 17 returns an* $(1-\epsilon)$*-optimal solution to MC-Capa. Moreover, the running time is* $O\left(\log(nU)\log(nUV)\log(nPVU)n^4/\epsilon^6\right)$.

Proof. Let S^* be the optimal solution to MC-Capa and $(\hat{\ell}_1, \hat{\ell}_2, \hat{\ell}_3)$ such that

$$\lambda_0 pvu\,(1+\epsilon)^{\hat{\ell}_1} \le \sum_{i\in S^*} p_i\left(\lambda_i + v_i\frac{\lambda_0 u(1+\epsilon)^{\hat{\ell}_2}}{1-uv(1+\epsilon)^{\hat{\ell}_3}}\right) \le \lambda_0 pvu\,(1+\epsilon)^{\hat{\ell}_1+1},$$

$$\lambda_0 u\,(1+\epsilon)^{\hat{\ell}_2} \le \sum_{i\notin S^*} u_i\lambda_i \le \lambda_0 u\,(1+\epsilon)^{\hat{\ell}_2+1},$$

$$uv\,(1+\epsilon)^{\hat{\ell}_3} \le \sum_{i\notin S^*} u_i v_i \le uv\,(1+\epsilon)^{\hat{\ell}_3+1}.$$

From Lemma B.1, we know that for $(h,g,t) = (\lambda_0 u\,(1+\epsilon)^{\hat{\ell}_2}, uv\,(1+\epsilon)^{\hat{\ell}_3}, \lambda_0 pvu(1+\epsilon)^{\hat{\ell}_1})$, $\mathcal{A}$ contains an assortment $\tilde{S}$ such that

$$\sum_{i\in\tilde{S}} p_i\left(\lambda_i + v_i\frac{h}{1-g}\right) \ge t(1-2\epsilon),$$

$$\sum_{i\notin\tilde{S}} u_i\lambda_i \ge h(1-2\epsilon)$$

$$\sum_{i\notin\tilde{S}} u_i v_i \ge g(1-2\epsilon).$$

Consequently,

$$\begin{aligned} f(\tilde{S}) &= \sum_{i\in\tilde{S}} p_i\left(\lambda_i + v_i\left(\sum_{j\notin\tilde{S}} u_j\lambda_j\right)\left(\sum_{m=0}^{\infty}\left(\sum_{j\notin\tilde{S}} u_j v_j\right)^m\right)\right) \\ &\ge \sum_{i\in\tilde{S}} p_i\left(\lambda_i + v_i\frac{h(1-2\epsilon)}{1-g(1-2\epsilon)}\right) \\ &\ge \sum_{i\in\tilde{S}} p_i\left(\lambda_i + v_i\frac{h}{1-g}\right) \\ &\ge \ell(1-2\epsilon) \ge f(S^*)\frac{1-2\epsilon}{1+\epsilon} \end{aligned}$$

Running Time. The running time analysis is similar than for the previous algorithms. □

Appendix C

Mallows-smoothed distribution over rankings approach for modeling choice

C.1 Proof of Theorem 4.2 (continued)

In this section, we prove that for a fixed R, $\sum_{\sigma \in h(R)} e^{-\theta . C_3(\sigma)}$ is equal to

$$\psi(|G| - m_0, \theta) \cdot \psi(|S| + m_0, \theta) \cdot \frac{e^{-\theta \cdot (k-1-\sum_{m=1}^{\ell-1} r_m)}}{1 + \cdots + e^{-\theta \cdot (|S|+m_0-1)}} \cdot \prod_{m=1}^{M} \frac{\psi(|G_m|, \theta)}{\psi(r_m, \theta) \cdot \psi(|G|_m - r_m, \theta)}.$$

We use a similar approach than in the first part of the proof. Let Γ be the set of $(\tilde{G}_1, \ldots, \tilde{G}_M) \subseteq (G_1, \ldots, G_M)$ such that $|\tilde{G}_m| = r_m$ for all $m \in [M]$. For all $\gamma = (\tilde{G}_1, \ldots, \tilde{G}_M) \in \Gamma$, let $t(\gamma)$ be the set of permutations σ which satisfy the following two conditions:

- $\sigma \in h(R)$.
- for all $m \in [M]$, the subset of products from G_m which is preferred to a_k is exactly $\tilde{G}_m$.

With this notation, we can write

$$\sum_{\sigma \in h(R)} e^{-\theta . C_3(\sigma)} = \sum_{\gamma \in \Gamma} \sum_{\sigma \in t(\gamma)} e^{-\theta \cdot (D_1(\sigma) + D_2(\sigma) + \sum_{m \in [M]} D_3(\sigma, m))},$$

where,

- $D_1(\sigma)$ is the sum of disagreements $\xi(\sigma, i, j)$ over pairs of products (i, j) such that either $i = k$ and $a_k \succ_\sigma a_j$ or $a_k \succ_\sigma a_i$ and $a_k \succ_\sigma a_j$.

- $D_2(\sigma)$ is the sum of disagreements $\xi(\sigma, i, j)$ over pairs of products (i, j) such that $a_i \succ_\sigma a_k$ and $a_j \succ_\sigma a_k$.

- for all $m \in [M]$, $D_3(\sigma, m)$ is the sum of disagreements $\xi(\sigma, i, j)$ over pairs of products (i, j) such that $a_i \in \tilde{G}_m$ and $a_j \in G_m \backslash \tilde{G}_m$.

Using the definition of $D_1(\sigma)$ and $D_2(\sigma)$ together with Theorem 4.1, we have that $\sum_{\sigma \in t(\gamma)} e^{-\theta \cdot (D_1(\sigma) + D_2(\sigma) + \sum_{m \in [M]} D_3(\sigma, m))}$ is equal to

$$\psi(|G| - m_0, \theta) \cdot \psi(|S| + m_0, \theta) \cdot \frac{e^{-\theta \cdot (k - 1 - \sum_{m=1}^{\ell - 1} r_m)}}{1 + \cdots + e^{-\theta \cdot (|S| + m_0 - 1)}} \cdot \sum_{\sigma \in t(\gamma)} e^{-\theta \cdot \sum_{m \in [M]} D_3(\sigma, m))}.$$

To complete the proof, it remains to compute $\sum_{\gamma \in \Gamma} \sum_{\sigma \in t(\gamma)} e^{-\theta \cdot \sum_{m \in [M]} D_3(\sigma, m)}$. Using the definition of the normalization constant, we have for all $m \in [M]$,

$$\psi(|G_m|, \theta) = \psi(r_m, \theta) \cdot \psi(|G_m| - r_m, \theta) \cdot \sum_{\gamma \in \Gamma} \sum_{\sigma \in t(\gamma)} e^{-\theta \cdot D_3(\sigma, m)},$$

which implies that

$$\sum_{\gamma \in \Gamma} \sum_{\sigma \in t(\gamma)} e^{-\theta \cdot \sum_{m \in [M]} D_3(\sigma, m)} = \prod_{m=1}^{M} \frac{\psi(|G_m|, \theta)}{\psi(r_m, \theta) \cdot \psi(|G_m| - r_m, \theta)},$$

and concludes the proof.