Display Optimization for Vertically Differentiated Locations under Multinomial Logit Preferences

Ali Aouad* Danny Segev†

31/07/2019

Abstract

We introduce a new optimization model, dubbed the display optimization problem, that captures a common aspect of choice behavior, known as the framing bias. In this setting, the objective is to optimize how distinct items (corresponding to products, web links, ads, etc.) are being displayed to a heterogeneous audience, whose choice preferences are influenced by the relative locations of items. Once items are assigned to vertically differentiated locations, customers consider a subset of the items displayed in the most favorable locations, before picking an alternative through Multinomial Logit choice probabilities.

The main contribution of this paper is to derive a polynomial-time approximation scheme for the display optimization problem. Our algorithm is based on an approximate dynamic programming formulation that exploits various structural properties to derive a compact state space representation of provably near-optimal item-to-position assignment decisions. As a by-product, our results improve on existing constant-factor approximations for closely-related models, and apply to general distributions over consideration sets. We develop the notion of "approximate assortments", that may be of independent interest and applicable in additional revenue management settings.

Lastly, we conduct extensive numerical studies to validate the proposed modeling approach and algorithm. Experiments on a public hotel booking data set demonstrate the superior predictive accuracy of our choice model vis-a-vis the Multinomial Logit choice model with location bias, proposed in earlier literature. In synthetic computational experiments, our approximation scheme dominates various benchmarks, including natural heuristics – greedy methods, local-search, priority rules – as well as state-of-the-art algorithms developed for closely-related models.

Keywords: Choice Models, Display Optimization, Approximation Schemes, Revenue Management.

*Department of Management Science and Operations, London Business School, Regent's Park, London, United Kingdom (NW14SA). Email: aaouad@london.edu.

†Department of Statistics, University of Haifa, Haifa 31905, Israel. Email: segevd@stat.haifa.ac.il

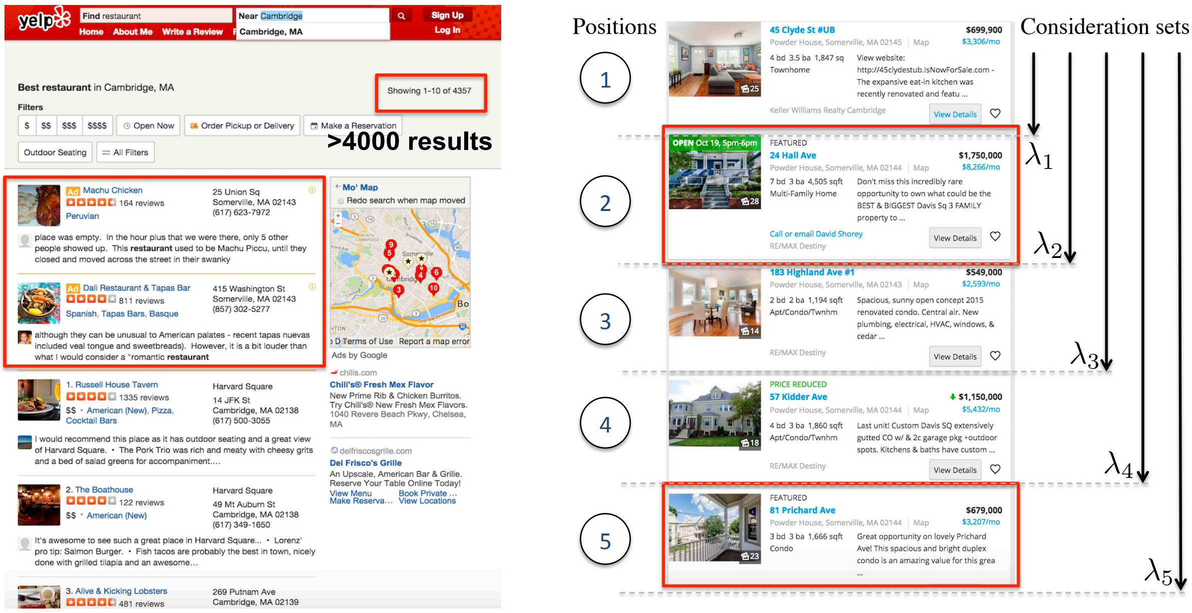
1 Introduction

A well-known behavioral bias in choice preferences, known as the 'framing effect', asserts that the choice outcomes over precisely the same set of alternatives are highly variable with changes in framing, perspective, and display (Tversky and Kahneman 1981, 1986). For example, in the context of retailing, typical customers have limited attention span, and do not thoroughly consider all alternatives available before selecting their preferred item (Payne 1976, Silk and Urban 1978, Bettman et al. 1998); in such settings, the framing bias favors the most prominent product alternatives on display. Hence, the main computational challenge for the retailer in online and offline environments is that of designing an effective display configuration, where heterogeneous and substitutable products are assigned to various display locations, each associated with a different propensity to be considered by end customers. At the same time, these products differ by their attractiveness to customers, as well as by the revenue they generate for retailers, and potentially by other metrics of interest, e.g., price margins, inventory positions, or perishability. As such, we are facing a typical revenue management tradeoff between demand generation and price cannibalization. On the one hand, the display allocation of products aims to encourage customers to buy, through conveniently displaying the best sales products; on the other hand, retailers wish to highlight the products that are more valuable from their perspective. As a result, the design of an effective display configuration strikes a delicate balance between satisfying the customers' utility and fulfilling the retailer's objective.

The above-mentioned tradeoff arises in applications that span across many industries, among which we discuss some of the more notable examples. In the context of in-store merchandising, retailers have limited display space with highly differentiated quality, such as vertical shelf position, which has a significant impact on whether a product is considered by shoppers (Chandon et al. 2009, Frank and Massy 1970, Dreze et al. 1995). A similar tradeoff appears in online retailing. Indeed, for each search query, unless end users specify their desired sorting rule, the retailer chooses a ranking order according to which relevant products are displayed across multiple web pages. There is strong empirical evidence that the display position on a web page is a key driver of purchase decisions and conversion (Ghose et al. 2014, Agarwal et al. 2011, Ghose and Yang 2009, Ursu 2018). In vertical search (e.g, Yelp, Trip Advisor, Trulia), the links displayed to end users are ranked by the search engine, as demonstrated in Figure 1a. The monetization of such platforms through sponsored results requires to jointly optimize user affinity and revenue generation, as both organic and sponsored listings are competing for users' attention (Yang and Ghose 2010, Jerath et al. 2011). More generally, in online advertising, the audience has a limited attention span for which multiple ads are competing, and the outcome of this competition is highly dependent on the ads display hierarchy (Jeziorski and Segal 2015, Narayanan and Kalyanam 2015).

1.1 Modeling approach

As further elaborated in Section 1.4, the interaction between display location effects and choice preferences has been overlooked in most computational problems studied in the revenue management literature thus far, in spite of its substantial impact on the choice outcomes (Chandon



(a) Display of search results (b) Consideration sets arising from location hierarchy

Figure 1: Examples of display configurations on Yelp (a) and Trulia (b), over large sets of alternatives, combining organic listings and sponsored results (marked by red rectangles).

et al. 2009, Agarwal et al. 2011, Ghose and Yang 2009). In this paper, we introduce a new optimization model, dubbed the *display optimization* problem, that captures the interplay between display configuration and choice preferences, and leverages well-established behavioral premises. For this purpose, we follow one of the prominent approaches to modeling choice in psychology and marketing, which assumes that customers consider only a subset of the alternatives available, named the consideration set, prior to picking one particular alternative (Bettman et al. 1998, Hauser 2014, Shocker et al. 1991). Screening procedures of this nature have been proposed to capture the search and exploration effort that customers are willing to make prior to their final decision (Hauser and Wernerfelt 1990, Mehta et al. 2003, Wu and Rangaswamy 2003).

The influence of the display locations on choice preferences is captured through the consideration sets structure. Specifically, motivated by concrete applications in retailing and vertical search (Breugelmans et al. 2007, Ursu 2018), we assume that these sets have a nested structure over vertically differentiated locations. In other words, the relative quality of locations at which products are placed creates an inclusion order between the consideration sets, as shown in Figure 1b. Our modeling approach captures the following natural effects: items placed at higher quality locations are more likely to appear in the consideration set of a customer in comparison to those at lower quality locations, which require additional search efforts. It is worth noting that this nested sequential structure has been validated in the context of vertical search by a recent empirical study, involving the Expedia Kaggle dataset (Ursu 2018). Specifically, the latter paper develops a model where consideration sets are formed endogenously by agents who minimize their search cost.

Once a customer forms his restricted consideration set, we assume that the choice mechanism through which a given item is picked within this set is described by the Multinomial Logit choice model (MNL). This model, whose specifics are described below, has extensively been studied and exploited by operations practitioners and researchers (see Section 1.4). For example, using data from Alibaba's online retail platforms, a recent study by Feldman et al. (2018) shows that assortment decisions made from MNL choice models have the potential to be significantly more profitable than sophisticated machine-learning based product recommendation algorithms, thus providing ample evidence that the standard MNL model is practically-relevant in online applications.

Instance parameters. We are given a collection of n items (corresponding to products, web links, ads, etc.), where each item i is associated with a revenue r_i as well as with a preference weight w_i. The display space for these items is represented by an ordered array comprised of n positions (or locations), numbered $1, \ldots, n$. Here, each position $k \in [n]$ is associated with a probability λ_k, whose precise meaning will be explained shortly. A position-to-item assignment $\mathcal{A} : [n] \to [n]$ is a bijective function mapping each position to a distinct item; throughout the paper, we use the shorthand notation $[n] = \{1, \ldots, n\}$. It is worth noting that, in practical applications, the number of items is not necessarily equal to the number of positions. However, such settings can easily be captured by our model, since by augmenting the original instance with either dummy items or dummy positions, we may assume without loss of generality that the number of items matches the number of positions.

The choice model. For a given assignment $\mathcal{A}$, the probabilistic choice outcomes are described through a representative customer taking random actions. Initially, this customer picks at random a subset of positions he is willing to consider, out of the collection of consideration sets $\mathcal{A}[1], \ldots, \mathcal{A}[n]$, where $\mathcal{A}[k]$ stands for the set of items assigned to the top-k positions $1, \ldots, k$. Technically speaking, this choice is represented by the random variable K, which stands for the endpoint of the array considered, with the convention that $\Pr[K = k] = \lambda_k$. This nested structure of consideration sets implies that top positions, with lower indices, are more likely to be considered than bottom positions, having higher indices. Unlike closely-related models (Davis et al. 2015, Gallego et al. 2016) that will be discussed in Section 1.2, we make no assumption whatsoever on the distribution $(\lambda_1, \ldots, \lambda_n)$. For this reason, it is easy to verify that our model actually subsumes a more general setting, where there are m positions, with arbitrary capacities $c_1, \ldots, c_m$, such that each position k may hold either exactly or at most c_k items. For example, in online settings each position could be interpreted as a web page with a predetermined capacity of precisely c_k search results, whereas in offline settings each position could correspond to a display shelf, with a capacity of at most c_k items. Additionally, we can easily capture settings in which another potential outcome is that the customer's consideration set is empty, by rescaling the probabilities $(\lambda_1, \ldots, \lambda_n)$.

Finally, conditional on the event $\{K = k\}$, where the chosen consideration set is $\mathcal{A}[k]$, the customer picks a specific item out of this selection, based on the MNL choice model, according to the preference weights $\{w_i : i \in \mathcal{A}[k]\}$. For ease of notation, for every subset $S \subseteq [n]$, we denote the combined weights of all items in S by $w(S) = \sum_{i \in S} w_i$. With this notation, the conditional probability of picking item $i \in \mathcal{A}[k]$ is $\frac{w_i}{1+w(\mathcal{A}[k])}$, noting that the customer may also decide to leave without purchasing any item, which happens with probability $\frac{1}{1+w(\mathcal{A}[k])}$.

Consequently, the conditional expected revenue in this case is

$$R_k(\mathcal{A}) = \sum_{i \in \mathcal{A}[k]} \frac{r_i w_i}{1 + w(\mathcal{A}[k])} \ .$$

Note that, due to the stochasticity in how the representative customer picks his random consideration set $\mathcal{A}[K]$, the expected revenue $R_K(\mathcal{A})$ of the assignment $\mathcal{A}$ is clearly a random variable.

Objective. The display optimization problem asks to compute a position-to-item assignment $\mathcal{A} : [n] \to [n]$ that maximizes the expected revenue, where the expectation is taken with respect to K as well as the MNL choice outcomes, which can be written as

$$\mathcal{R}(\mathcal{A}) = \sum_{k \in [n]} \lambda_k \cdot R_k(\mathcal{A}) \ . \tag{1}$$

1.2 Closely-related models

From a thematic perspective, the display optimization problem constitutes in particular a strict generalization of two closely-related problems, both aiming to determine a sequencing strategy (or prioritization) over different items, facing customers with heterogeneous preferences. Specifically, Davis et al. (2015) have recently introduced the *assortment over time* problem, where retailers build their assortment incrementally, by prioritizing their product launches over a discrete planning horizon. For monotone choice models, they derived a $\frac{1}{2}$-approximation algorithm, assuming uniform arrival rates (i.e., $\lambda_k = \frac{1}{n}$). Due to the analogy between time periods and positions, assortment over time under Multinomial Logit preferences is in fact a highly-structured special case of the display optimization problem. Furthermore, what makes the former significantly easier to approximate is that retailers are allowed to leave certain positions vacant, while still garnering an expected revenue from these positions, as well as to avoid displaying certain items. Subsequent to our work, Gallego et al. (2016) studied the *product framing* model, which generalizes the assortment over time problem by allowing products to be introduced in limited-size batches, and by considering non-uniform customer arrival rates. Under certain technical conditions, they devised an elegant algorithm attaining an approximation guarantee of $6/\pi^2 \approx 0.607$. However, this model still operates under the assumption that positions can be left empty or blank, which could be unrealistic for the display optimization applications we consider.

In general, enabling retailers to leave certain positions vacant, while still garnering an expected revenue from these positions, leads to a simpler problem formulation. For example, to maximize his expected revenue, the retailer may decide to display a very limited selection of items – potentially with a single high-priced item. In contrast, in our setting, there is a fixed number of locations that need to be assigned with an item, thereby retailers decide on a sequencing strategy over the entire collection of items. This feature eliminates numerous well-behaved properties of the objective function and leads to technical difficulties of greater magnitude.

From a practical point of view, identifying a complete position-to-item assignment is particularly relevant in vertical search applications, where an online platform is required to display the full search results, and cannot "hide" certain items. For example, most vertical search platforms, such as Yelp or Trulia, rank all listings available for a given search query; some of these listings are sponsored (paid results) and others are organic (non-paid results). Even though it seems intuitive that the platform should favor sponsored listings, organic ones cannot be discarded, even if this strategy is revenue-wise beneficial. The objective is thus to allocate all listings across existing display positions, some of which are more prominent than others. Similarly, large online retailers are also marketplace channels that display products managed by third party-providers, such as Amazon, that only manages around $\frac{1}{10}$-th of the assortment on display. Due to contractual relationships with third-party sellers, Amazon cannot choose to hide certain items (in particular, those that cannibalize Amazon's personal revenue), and to exclusively promote its own products.

By focusing on the Multinomial Logit choice model, our results significantly improve on the existing approximation factors for both assortment over time and product framing, and avoid additional assumptions made in previous literature on the distribution over consideration sets. It is worth mentioning that we are not aware of any way to employ the algorithmic ideas of Davis et al. (2015) and Gallego et al. (2016) in order to obtain any non-trivial performance guarantee for the display optimization problem.

1.3 Our results

Performance guarantees. The main contribution of this paper is to introduce a fundamental optimization model that accounts for display location effects in the face of heterogeneous customer segments, for which we devise a polynomial-time approximation scheme (PTAS). In other words, for any desired level of accuracy $\epsilon > 0$, we prove that the optimal expected revenue of any display optimization instance can be efficiently approximated within a factor of $1 - \epsilon$, as formally stated in the following theorem.

Theorem 1.1. *Under Multinomial Logit preferences, the display optimization problem admits a polynomial-time approximation scheme. Specifically, for any $\epsilon \in (0, \frac{1}{75})$, our dynamic programming algorithm approximates the optimal expected revenue within factor $1 - \epsilon$ in time $O(|\mathcal{I}|^{O(\frac{1}{\epsilon^2} \cdot \log \frac{1}{\epsilon})})$, where $|\mathcal{I}|$ denotes the size of the instance.*

It is worth mentioning that our performance guarantees are essentially best-possible. As explained in Section 1.2, under Multinomial Logit preferences, the display optimization problem captures as a special case the product framing model of Gallego et al. (2016), who proved that the latter problem is NP-hard. As an interesting side note, simple counter-examples demonstrate that natural heuristics to address the display optimization problem – such as local-search or greedy procedures – have arbitrarily large optimality gaps, generating only an $O(\frac{1}{n})$ fraction of the optimal expected revenue in the worst case. These findings are formally established in Appendix A.

Technical outline. From a technical perspective, our algorithmic approach is based on synthesizing methodologies related to approximate dynamic programming, efficient guessing, enumeration ideas, and various function relaxations. These methods are brought to fruition through the derivation of fundamental structural properties, shown to be inherent to near-optimal assignments. In Section 2, we perform preliminary transformations of the display optimization instance, whereby key input parameters are rounded. Relatedly, we develop a notion of "approximate assortments", which is a crucial ingredient of our algorithm. Consequently, in Section 3, we devise a value approximation for the display optimization problem, that is, a method that approximately computes the conditional expected revenues $R_1(\mathcal{A}), \ldots, R_n(\mathcal{A})$ using limited information about any given assignment $\mathcal{A}$. Building on this value approximation, in Section 4, we formulate an efficient dynamic program for the display optimization problem, and explain how its actions are converted into analogous assignment decisions. Finally, the performance guarantees of our algorithm are formally proven in Section 5. We show that our dynamic programming algorithm is an approximation scheme for the display optimization problem, thereby completing the proof of Theorem 1.1.

Case study. In Section 6, we demonstrate the predictive power of our modeling approach on real-world choice data, that exhibits significant location effects. Specifically, we utilize a publicly-available data set that describes search results and bookings for hotel rooms on Expedia's online platform. Our consideration set-based approach, described in Section 1.1, is benchmarked against an MNL choice model with position bias, which stands as the most natural alternative proposed in related literature (Davis et al. 2013, Abeliuk et al. 2016). In both cases, users' relative preferences over hotel listings are described through similar feature-based MNL models. However, the choice models differentiate themselves through the incorporation of location effects. While these choice models have precisely the same number of parameters, we find that our consideration set-based approach leads to superior predictive accuracy, with up to 2% relative improvements of out-of-sample log-likelihood.

Computational study. Through extensive numerical experiments, we demonstrate in Section 7 the practical efficiency of our algorithmic approach. Specifically, our algorithm is thoroughly tested against various heuristics, including a local search method, a greedy algorithm, and common-sense ranking rules, as well as against a suitable adaptation of the constant-factor approximation of Gallego et al. (2016), originally suggested for the product framing problem. To this end, we propose and implement a simplified version of our approximation scheme, which is shown to improve on the best heuristic by up to 3%, and to provide better trade-offs between performance and speed.

Implications to related models. As an immediate by-product of Theorem 1.1, it follows that our approach provides a PTAS for the assortment over time model studied by Davis et al. (2015) as well as for the product framing problem considered by Gallego et al. (2016), when customers' preferences are captured by the MNL choice model. To our knowledge, this result cannot be inferred from the existing analysis presented in these papers, where a constant-factor loss in optimality is inevitable.

Theorem 1.2. *Under Multinomial Logit preferences, both assortment over time and product framing admit a polynomial-time approximation scheme.*

Further comments. From a conceptual perspective, our results generally show that a more realistic approach to modeling choice, taking into account display location effects, does not necessarily come at the expense of creating a strongly-inapproximable problem. Specifically, we prove that near-optimal approximations can still be obtained despite enlarging the dimensionality of the search space in comparison to optimization problems of similar flavor, such as standard assortment planning formulations (see Section 1.4). In addition, the model we propose deviates from some of the common ingredients allowing for the tractability of such formulations. In this regard, we consider a choice model not subject to the Independence of Irrelevant Alternatives property, which is likely to be violated in practical settings (McFadden et al. 1977, Ben-Akiva and Lerman 1985). Our choice model can be viewed as a solution-dependent Mixture of Multinomial Logit choice model (MMNL) with a parametric number of customer types (Bront et al. 2009, Rusmevichientong et al. 2014, Méndez-Diaz et al. 2014, Feldman and Topaloglu 2015). It is worth noting that, under an MMNL choice model with n customer types, even the seemingly-simple assortment planning problem with a solution-independent choice model is NP-hard to approximate within factor $O(n^{1-\epsilon})$, for any fixed $\epsilon > 0$ (Désir et al. 2014).

1.4 Directly-related work

Online advertising with externalities. A recent line of work in algorithmic mechanism design has studied the substitution effects between competing ads, named externality effects (Xiong et al. 2012, Xu et al. 2010, Jerath et al. 2011). Here, instead of being exogenous, the click-through-rates are expressed as a function of the display configuration. Kempe and Mahdian (2008) proposed the so-called 'cascade' model to describe the inter-dependency of choice over different ads placed in vertically differentiated locations. In this model, customers sequentially examine the ads according to the linear order of locations, a structural property that allows the authors to propose efficient solution methods. The cascade model has several advantages over standard choice models: for example, this model can be extended to capture multiple clicks and leverage contextual information, solving its related assortment and pricing optimization problems is relatively easy, etc. However, this approach for modeling users' choices is stylized: the probability of clicking on a given ad does not depend on any of the alternatives placed below. This assumption simplifies the substitution effects, but does not agree with experimental studies (Breugelmans et al. 2007). It is worth pointing out that this stream of literature focuses on the design of algorithms allowing for truthful mechanisms, at the expense of more realistic modeling of the choice behavior.

Assortment optimization. Closer to the present setting are assortment optimization problems, which have received a great deal of attention in the revenue management literature. Here, a retailer wishes to select a subset of products to maximize his expected revenue in the face of heterogeneous customers, given a choice model that provides a fine-grained description of the substitution effects between competing products. Problems of this nature have recently been studied in various shapes and forms, depending on the probabilistic structure of choice preferences, and additional constraints on the set of products offered. Since an inclusive overview of existing work is beyond the scope of this paper, we refer the reader to directly-related papers (Li et al. 2015, Davis et al. 2014, Blanchet et al. 2016, Farias et al. 2013, Aouad et al. 2015, Talluri and Van Ryzin 2004, Rusmevichientong et al. 2010) for a comprehensive literature review.

Quite surprisingly, beyond the work of Davis et al. (2015) and Gallego et al. (2016), which is discussed in Section 1.2, display configuration effects are largely overlooked in assortment optimization problems, other than a handful of exceptions. Feldman and Topaloglu (2017) studied a choice model that combines MNL preferences with nested consideration sets, for which they devised a fully polynomial-time approximation scheme. However, the structure of the consideration sets thereof is given a priori, independently of the display configuration. Generally speaking, our problem formulation contends with searching over position-to-item assignments, instead of identifying an optimal assortment out of all subsets of items. To our knowledge, Davis, Gallego, and Topaloglu (2013) were the first to introduce location effects in an assortment optimization model with MNL choice preferences. Assuming that these effects are captured by modified product weights, the problem is recast as a linear program with totally unimodular constraints, rendering it polynomial-time solvable. Similarly, Abeliuk et al. (2016) devised an efficient solution method for an assortment optimization model where the MNL preference weights are biased by an exogenous multiplicative factor, based on their location. However, the modeling approaches of these papers assume that customers consider all offered products, and only the relative weights of products are affected by the display location. Hence, such models impose a strong structure on choice outcomes, best exemplified by the Independence of Irrelevant Alternatives property, which limits its practical applicability (McFadden et al. 1977, Ben-Akiva and Lerman 1985). Namely, the relative choice probabilities of products placed at fixed locations are invariant to the display configuration of all other products. In contrast, our model endogenously captures the interplay between location and choice, in the same spirit as the aforementioned ad-externality models.

The Multinomial Logit model. The MNL choice model is arguably the most widespread approach for modeling choice among practitioners (McFadden 1980, Guadagni and Little 1983, Ben-Akiva and Lerman 1985, Grover and Vriens 2006, Chandukala et al. 2008). This model, independently proposed by Luce (1959) and Plackett (1975), is grounded in the economic theory of utility maximization, as it describes the probabilistic choice outcomes of a representative agent who maximizes his utility over different alternatives, through a noisy evaluation of the utility they procure. The far-reaching popularity of this model was notably driven by its efficient estimation procedures (McFadden 1973, Hausman and McFadden 1984, Talluri and Van Ryzin 2004), even with limited data (Ford 1957, Negahban et al. 2012), as well as by its computational tractability in numerous decision-making problems. Indeed, in the context of static assortment planning, MNL choice preferences are well-understood. In particular, exact polynomial-time algorithms have been devised for the unconstrained assortment optimization problem (Talluri and Van Ryzin 2004), where any subset of products can be offered, and later on for its capacitated variant (Rusmevichientong et al. 2010), where an upper bound is imposed on the number of offered products. Recent contributions to this line of work incorporate totally-unimodular constraints (Davis et al. 2013), random choice parameters (Rusmevichientong et al. 2014), and robust optimization settings (Rusmevichientong and Topaloglu 2012).

2 Preliminaries

In this section, we present a number of preprocessing operations, that will be instrumental in deriving our approximation scheme. These operations are motivated by the following basic observation, referred to as a *priority rule*: among any two items having identical MNL weights, it is revenue-wise preferable to display the higher-priced item at a higher-ranked position. At first glance, this priority rule may not be useful for arbitrary instances, since items generally have distinct MNL weights. However, broader algorithmic implications can be derived when these weights are slightly perturbed. More specifically, at a negligible loss of optimality, by rounding the original MNL weights, we obtain a more structured instance, where the priority rule is applicable within each class of items of uniform MNL weights (Section 2.1). Consequently, this operation will enable us to develop a computationally-efficient notion of *approximate assortments* (Section 2.2). That is, we construct a polynomially-sized collection of assortments, such that any assortment that satisfies the priority rules has a suitable approximate counterpart within this collection.

Additional notation and terminology. Before describing these algorithmic ideas, we introduce additional notation and terminology that will be used throughout the paper. For an assortment $S \subseteq [n]$, we denote the expected revenue generated by this assortment under the MNL model by $R(S) = \sum_{i \in S} \frac{r_i w_i}{1+w(S)} = \sum_{i \in S} \frac{\rho_i}{1+w(S)}$, where $\rho_i = r_i w_i$; the latter term is referred to as the ρ-quantity of item i. Noting that the expected revenue $R(S)$ depends on two quantities, the ρ-quantities appearing at the numerator and the weights appearing at the denominator, we define the function $\rho(S) = \sum_{i \in S} \rho_i$, so that the expected revenue function can be written as $R(S) = \frac{\rho(S)}{1+w(S)}$.

As in the objective function representation (1), the expected revenue generated by a position-to-item assignment $\mathcal{A} : [n] \to [n]$ can be specified as:

$$\mathcal{R}(\mathcal{A}) = \sum_{k=1}^{n} \lambda_k \cdot R(\mathcal{A}[k]) \ .$$

We also use the shorthand notation $R_k(\mathcal{A}) = R(\mathcal{A}[k])$ to designate the expected revenue conditional on picking position $k \in [n]$. In addition, we use $\mathcal{A}^{-1}(i)$ to denote the position occupied by item i. Finally, $\mathcal{A}^*$ designates a fixed optimal assignment, i.e., one that maximizes $\mathcal{R}(\mathcal{A})$ over all possible assignments.

2.1 Priority rules

Given an error parameter $\delta > 0$, we first describe three structural assumptions on the instance parameters, which are essential for the development of an efficient approximation scheme. We show in Appendix B that the conjunction of these assumptions leads to a negligible loss in optimality.

Assumption 2.1. *The preference weight of every item $i \in [n]$ is of the form $w_i = \frac{\delta}{n} \cdot (1+\delta)^{\tau}$, for some integer $\tau \geq 0$.*

Assumption 2.2. *The ρ-quantity of every item $i \in [n]$ is of the form $\rho_i = \rho_{\min} \cdot (1+\delta)^\tau$ for some integer $\tau \geq 0$, where $\rho_{\min} = \min_{i \in [n]} \rho_i$.*

These assumptions are by no means required, and their sole purpose is to simplify the exposition of our algorithm. Indeed, Assumptions 2.1 and 2.2 can be enforced by "rounding" the MNL preference weights and prices to their nearest power of $(1+\delta)$. As shown in Appendix B, these operations result in losing only an $O(\delta)$-fraction of the optimal revenue. Similarly, any problem instance can be altered to satisfy the next assumption while incurring a negligible loss in optimality.

Assumption 2.3. $\lambda_k > 0$ *for every position* $k \in [n]$.

It is important to note that, from this point on, we no longer treat $(\lambda_1, \ldots, \lambda_n)$ as a probability distribution, and in particular, $\sum_{k=1}^n \lambda_k$ could be different from 1, due to the alterations required to enforce Assumption 2.3. By a slight abuse of terminology, we will still refer to $\lambda_1, \ldots, \lambda_n$ as the "probability weights" of the positions $1, \ldots, n$, while $\mathcal{R}(\mathcal{A})$ is the "expected revenue" of a given assignment $\mathcal{A}$.

Priority rules. Now, by taking advantage of Assumptions 2.1, 2.2, and 2.3, we show that there exists so-called *priority rules* satisfied by any optimal assignment. More specifically, given an assignment $\mathcal{A}$, the priority rules refer to the following two properties:

1. For any pair of distinct items $i_1 \neq i_2$, if $w_{i_1} = w_{i_2}$ and $\rho_{i_1} > \rho_{i_2}$, then $\mathcal{A}^{-1}(i_1) < \mathcal{A}^{-1}(i_2)$.

2. For any pair of distinct items $i_1 \neq i_2$, if $\rho_{i_1} = \rho_{i_2}$ and $w_{i_1} < w_{i_2}$, then $\mathcal{A}^{-1}(i_1) < \mathcal{A}^{-1}(i_2)$.

The next claim shows that any optimal assignment follows these priority rules. Hereafter, $\mathcal{A}^*$ designates a fixed optimal assignment of the display optimization problem.

Claim 2.4. *The optimal assignment $\mathcal{A}^*$ satisfies the priority rules 1 and 2.*

Proof. To verify that priority rule 1 is met, we show that, if our original optimal assignment $\mathcal{A}^*$ does not satisfy this property, swapping between the positions of a violated pair (i_1, i_2) would strictly improve the objective function, contradicting the optimality of $\mathcal{A}^*$. Indeed, letting $\vec{\mathcal{A}}$ be the assignment obtained from $\mathcal{A}^*$ by swapping the positions of the items i_1 and i_2, we have:

$$\mathcal{R}\left(\mathcal{A}^*\right) - \mathcal{R}\left(\vec{\mathcal{A}}\right) = \sum_{k=\mathcal{A}^{*-1}(i_2)}^{\mathcal{A}^{*-1}(i_1)} \lambda_k \cdot \left(\frac{\rho(\mathcal{A}^*[k])}{1+w(\mathcal{A}^*[k])} - \frac{\rho(\vec{\mathcal{A}}[k])}{1+w(\vec{\mathcal{A}}[k])} \right) = \sum_{k=\mathcal{A}^{*-1}(i_2)}^{\mathcal{A}^{*-1}(i_1)} \lambda_k \cdot \frac{\rho_{i_2} - \rho_{i_1}}{1+w(\mathcal{A}^*[k])} < 0\,,$$

where the inequality follows since $\rho_{i_1} > \rho_{i_2}$ by hypothesis, and $\lambda_k > 0$ for every $k \in [n]$ in light of Assumption 2.3.

Using a similar proof by contradiction, one can easily verify that priority rule 2 is satisfied by the optimal assignment $\mathcal{A}^*$. Given a pair of items $i_1 \neq i_2$ having identical ρ-quantities, i.e., $\rho_{i_1} = \rho_{i_2}$, if $w_{i_1} < w_{i_2}$, then $\mathcal{A}^{*-1}(i_1) < \mathcal{A}^{*-1}(i_2)$; otherwise we would strictly increase the expected revenue by swapping the violated pair. ■

■

In an analogous fashion, we say that an assortment $\mathcal{U}$ satisfies the priority rules when the following two properties hold:

1. For every $i_1 \neq i_2 \in [n]$ with $w_{i_1} = w_{i_2}$ and $\rho_{i_1} > \rho_{i_2}$, if $i_2 \in \mathcal{U}$ then $i_1 \in \mathcal{U}$.

2. For every $i_1 \neq i_2 \in [n]$ with $\rho_{i_1} = \rho_{i_2}$ and $w_{i_1} < w_{i_2}$, if $i_2 \in \mathcal{U}$ then $i_1 \in \mathcal{U}$.

Consequently, we let $\mathcal{P}$ designate the family of assortments that satisfy the priority rules. Based on the preceding discussion, we have $\mathcal{A}^*[k] \in \mathcal{P}$ for every position $k \in [n]$.

2.2 Approximate assortments

In this section, we develop an "approximation" procedure for every assortment $\mathcal{U} \in \mathcal{P}$, as defined in Section 2.1. Before formally describing our notion of approximate assortments, we highlight certain intuitive properties that this approximation method should possess in order to develop an approximation scheme for the display optimization problem. An assortment $\hat{\mathcal{U}}$ qualifies as a "good" approximation of $\mathcal{U}$ when the two subsets $\hat{\mathcal{U}}$ and $\mathcal{U}$ share a significant fraction of their items; in other words, the cardinality of $\mathcal{U} \cap \hat{\mathcal{U}}$ nearly matches that of $\mathcal{U}$ and $\hat{\mathcal{U}}$. Moreover, since an assortment $\mathcal{U}$ is evaluated on the basis of its expected revenue $R(\mathcal{U}) = \frac{\rho(\mathcal{U})}{1+w(\mathcal{U})}$, it follows that $\mathcal{U}$ is "well-approximated" by $\hat{\mathcal{U}}$ when $\rho(\mathcal{U})$ and $\rho(\hat{\mathcal{U}})$ are nearly equal, and similarly, $w(\mathcal{U})$ and $w(\hat{\mathcal{U}})$ are nearly equal as well. Lastly, while the family of assortments $\mathcal{P}$ is exponential in size, we would like to ensure that the number of distinct approximate assortments $\hat{\mathcal{U}}$ is polynomial, so that an exhaustive enumeration over all approximate assortments can be carried out efficiently.

In what follows, we construct an approximation procedure that achieves the above-mentioned properties, as stated by the next lemma.

Lemma 2.5. *For any accuracy level $\delta > 0$, under Assumptions 2.1-2.3, there exist collections of assortments $\mathcal{S}^-(\delta)$ and $\mathcal{S}^+(\delta)$, where $|\mathcal{S}^-(\delta)| = O(|\mathcal{I}|^{O(\frac{1}{\delta^2} \cdot \log \frac{1}{\delta})})$ and $|\mathcal{S}^+(\delta)| = O(|\mathcal{I}|^{O(\frac{1}{\delta^2} \cdot \log \frac{1}{\delta})})$, such that for every assortment $\mathcal{U} \in \mathcal{P}$, there exist approximate assortments $\mathcal{U}^- \in \mathcal{S}^+(\delta)$ and $\mathcal{U}^+ \in \mathcal{S}^-(\delta)$ satisfying the following properties:*

1. *$\mathcal{U}^- \subseteq \mathcal{U} \subseteq \mathcal{U}^+$.*

2. *$\rho(\mathcal{U}^-) \geq (1-\delta) \cdot \rho(\mathcal{U})$ and $w(\mathcal{U}^-) \geq (1-\delta) \cdot w(\mathcal{U})$.*

3. *$\rho(\mathcal{U}^+) \leq (1+\delta) \cdot \rho(\mathcal{U})$ and $w(\mathcal{U}^+) \leq (1+\delta) \cdot w(\mathcal{U})$.*

Furthermore, the sets $\mathcal{S}^-(\delta)$ and $\mathcal{S}^+(\delta)$ can be constructed in time $O(|\mathcal{I}|^{O(\frac{1}{\epsilon^2} \cdot \log \frac{1}{\epsilon})})$.

Due to lengthy technical details, the proof of Lemma 2.5 is deferred to Appendix C.1. In what follows, given the chain of inclusions stated by Property 1 above, $\mathcal{S}^-(\delta)$ will be referred to as the collection of *sub-assortments*, while $\mathcal{S}^+(\delta)$ will be that of *super-assortments*. In particular, when $\delta \in (0, \frac{1}{2})$, Properties 1-3 of Lemma 2.5 imply the following relationships:

$$\rho\left(\mathcal{U}^+ \setminus \mathcal{U}^-\right) = \rho\left(\hat{\mathcal{U}}^+\right) - \rho\left(\mathcal{U}\right) + \rho\left(\mathcal{U}\right) - \rho\left(\hat{\mathcal{U}}^-\right) \leq 2\delta \cdot \rho\left(\mathcal{U}\right) \leq 4\delta \cdot \rho\left(\mathcal{U}^-\right) \ , \qquad (2)$$

Similarly, we obtain:

$$w\left(\mathcal{U}^+ \setminus \mathcal{U}^-\right) \leq 4\delta \cdot w\left(\mathcal{U}^-\right) \ . \qquad (3)$$

It is worth remarking that, in contrast with Properties 2 and 3, none of inequalities (2) and (3) involve the underlying assortment $\mathcal{U}$.

Set inclusion properties. Algorithmically-speaking, the notion of approximate assortments developed by Lemma 2.5 will be utilized to approximate the consideration sets induced by assignments that satisfy the priority rules. As highlighted in Section 1.1, for any such assignment $\mathcal{A}$, the sequence of consideration sets $(\mathcal{A}[1], \ldots, \mathcal{A}[n])$ is nested, namely, $\mathcal{A}[1] \subseteq \ldots \subseteq \mathcal{A}[n]$. Hence, it is important to uncover whether our approximation method of Lemma 2.5 preserves similar set inclusion properties. To this end, given an accuracy level of $\delta > 0$, for every position $k \in [n]$, we denote by $(S_k^-, S_k^+) \in \mathcal{S}^-(\delta) \times \mathcal{S}^+(\delta)$ the pair of approximate assortments associated with the assortment $\mathcal{A}[k]$. As shown by the next claim, the sequence of super-assortments $(S_1^+, \ldots, S_n^+)$ is nested. The proof is presented in Appendix C.2.

Lemma 2.6. *For every pair of assortments $\mathcal{U}_1, \mathcal{U}_2 \in \mathcal{P}$ and accuracy level $\delta > 0$, let $\hat{\mathcal{U}}_1^+, \hat{\mathcal{U}}_2^+ \in \mathcal{S}^+(\delta)$ be their respective super-assortments. If $\mathcal{U}_1 \subseteq \mathcal{U}_2$, then $\hat{\mathcal{U}}_1^+ \subseteq \hat{\mathcal{U}}_2^+$.*

On the other hand, unlike the sequence of super-assortments $(S_1^+ \ldots, S_n^+)$, the sequence of sub-assortments $(S_1^-, \ldots, S_n^-)$ is not nested. For example, suppose that $n = 2$, $r_1 = \frac{1}{3}$, $w_1 = \frac{1}{2}$, $r_2 = \frac{1}{9}$ and $w_2 = 100$. In this case, it is easy to verify that the optimal assignment $\mathcal{A}^*$ is characterized by $\mathcal{A}^*(1) = 1$ and $\mathcal{A}^*(2) = 2$. Now, let S_1^- and S_2^- be the sub-assortments associated with $\mathcal{A}^*[1] = \{1\}$ and $\mathcal{A}^*[2] = \{1, 2\}$, respectively, with an accuracy level $\delta = 0.9$. By noting that $\mathcal{S}^-(0.9) = \{\{1\}, \{2\}\}$, it follows that $S_1^- = \{1\} \not\subseteq \{2\} = S_2^-$, meaning that the sub-assortments are not nested. This observation is a first indication of the structural alterations entailed by the approximation method of Lemma 2.5.

3 Value Approximation

In this section, we construct a value approximation for the display optimization problem. Given an assignment $\mathcal{A}$, we introduce the *revenue function* $k \mapsto R_k(\mathcal{A})$, that maps each position $k \in [n]$ to the expected revenue conditional on choosing the consideration set $\mathcal{A}[k]$. Our goal is to approximate the revenue function $k \mapsto R_k(\mathcal{A})$ using limited information about the assignment $\mathcal{A}$. We begin by showing in Section 3.1 that the revenue function $k \mapsto R_k(\mathcal{A}^*)$ induced by the optimal assignment $\mathcal{A}^*$ is unimodal. Focusing on assignments that satisfy the priority rules defined in Section 2.1 and induce a unimodal revenue function, we develop a functional approximation of the revenue function, incurring an $O(\epsilon)$-error for any accuracy level $\epsilon > 0$, in a sense made precise in Section 3.2. Rather than having access to the assortment $\mathcal{A}[k]$ for all positions $k \in [n]$, our approximation method utilizes their approximate counterparts, as defined by Lemma 2.5, only for a small subset of positions $k \in [n]$. This "loss of information" will enable the formulation of an efficient approximate dynamic program for the display optimization problem, presented in Section 4.

3.1 Unimodality of $k \mapsto R_k(\mathcal{A}^*)$

Here, we characterize the variations of the revenue function $k \mapsto R_k(\mathcal{A}^*)$ induced by the optimal assignment $\mathcal{A}^*$. Specifically, in the next claim, we establish the unimodality of this revenue function.

Lemma 3.1. *The revenue function $k \mapsto R_k(\mathcal{A}^*)$ is unimodal. That is, there exists $k_{\text{mid}} \in [n]$ such that $k \mapsto R_k(\mathcal{A}^*)$ is non-decreasing over $[1, k_{\text{mid}}]$ and non-increasing over $[k_{\text{mid}}, n]$.*

Proof. To arrive at a contradiction, suppose that for some $k \in [n]$ we have $R_k(\mathcal{A}^*) > R_{k+1}(\mathcal{A}^*)$ and $R_{k+1}(\mathcal{A}^*) < R_{k+2}(\mathcal{A}^*)$. In the MNL model, adding a single item $i \in [n]$ to an assortment $S \subseteq [n]$ generates an expected revenue that can be written as a convex combination between the expected revenue of S and the price of item i. Specifically, it is easy to verify that

$$R\left(S \cup \{i\}\right) = R\left(S\right) \cdot \frac{1 + w(S)}{1 + w(S) + w_i} + r_i \cdot \frac{w_i}{1 + w(S) + w_i} \ .$$

Hence, $R_k(\mathcal{A}^*) > R_{k+1}(\mathcal{A}^*)$ means that the item $i_1 = \mathcal{A}^*(k+1)$ has a price of $r_{i_1} < R_{k+1}(\mathcal{A}^*)$, and similarly, $R_{k+1}(\mathcal{A}^*) < R_{k+2}(\mathcal{A}^*)$ implies that the item $i_2 = \mathcal{A}^*(k+2)$ has a price of $r_{i_2} > R_{k+1}(\mathcal{A}^*)$. As a result, $r_{i_2} > r_{i_1}$. Now consider the assignment $\hat{\mathcal{A}}$, obtained from $\mathcal{A}^*$ by swapping between i_1 and i_2. Observe that $\hat{\mathcal{A}}[1,j] = \mathcal{A}^*[1,j]$ for every $j \in [n] \setminus \{k+1\}$, meaning that $R_j(\hat{\mathcal{A}}) = R_j(\mathcal{A}^*)$ for these positions. However, in light of the above observations, $R_{k+1}(\hat{\mathcal{A}}) \geq \min\{r_{i_2}, R_k(\mathcal{A}^*)\} > R_{k+1}(\mathcal{A}^*)$, where the former inequality holds since $R_{k+1}(\hat{\mathcal{A}})$ is a strict convex combination between $R_k(\mathcal{A}^*)$ and r_{i_2}, while the latter inequality holds since $r_{i_2} > R_{k+1}(\mathcal{A}^*)$ and also $R_k(\mathcal{A}^*) > R_{k+1}(\mathcal{A}^*)$. As a result, we obtain

$$\mathcal{R}\left(\hat{\mathcal{A}}\right) - \mathcal{R}\left(\mathcal{A}^*\right) = \lambda_{k+1} \cdot \left(R_{k+1}\left(\hat{\mathcal{A}}\right) - R_{k+1}\left(\mathcal{A}^*\right)\right) > 0 \ ,$$

where the strict inequality proceeds from Assumption 2.3, contradicting the optimality of $\mathcal{A}^*$.
■ ■

3.2 Approximation method

Let $\mathcal{P}_\wedge$ be the collection of assignments that satisfy the priority rules 1 and 2 (see Section 2.1) and induce a unimodal revenue function. In this section, we fix an arbitrary assignment $\mathcal{A} \in \mathcal{P}_\wedge$.

The basic idea. Given an accuracy level $\epsilon > 0$, our goal is to construct a function $f_{\mathcal{A}} : [n] \to \mathbb{R}^+$ that approximates the revenue function $k \mapsto R_k(\mathcal{A})$ from below up to a factor of $1 + O(\epsilon)$; specifically, for every position $k \in [n]$, we will have $f_{\mathcal{A}}(k) \leq R_k(\mathcal{A} \leq (1 + 4\epsilon) \cdot f_{\mathcal{A}}(k)$. Our approach, illustrated by Figure 2, is to define $f_{\mathcal{A}}(k)$ as a step function. To this end, we define a partition of the positions $[n]$ into intervals $I_1, \ldots, I_m$. Within each interval, the step function $f_{\mathcal{A}}$ is constant and equal to the minimum value of $R_k(\mathcal{A})$ over this interval. Namely, for every $k \in I_j$, we have

$$f_{\mathcal{A}}(k) = \min\{\mathcal{R}_q(\mathcal{A}) : q \in I_j\} \ . \tag{4}$$

Clearly, for any given partition of $[n]$, the resulting function $f_{\mathcal{A}}$ constitutes a lower bound on the revenue function $k \mapsto R_k(\mathcal{A})$. However, the main challenge is to devise a sufficiently refined partition $I_1, \ldots, I_m$, such that the resulting step function $f_{\mathcal{A}}$ provides a good approximation of the revenue function over all positions $k \in [n]$. Before constructing the partition $I_1, \ldots, I_m$, we explain how the right-hand side of equation (4) is approximately computed, by leveraging the notion of approximate assortments developed in Lemma 2.5 and the unimodality of the revenue function established in Lemma 3.1.

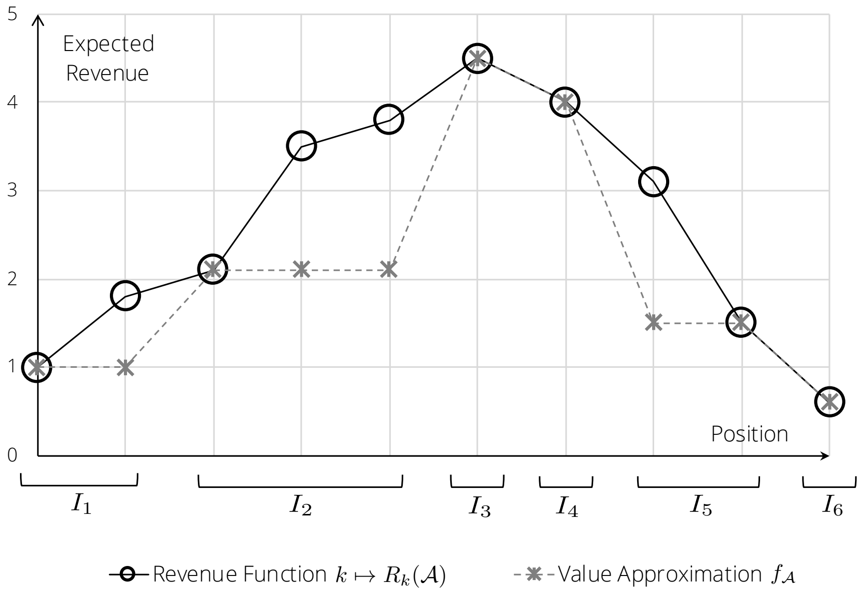


Figure 2: Illustration of the approximation method with $\boldsymbol{\epsilon = 1}$.

How do we leverage the unimodality of the revenue function? One important observation is that, since the revenue function $k \mapsto R_k(\mathcal{A})$ is unimodal, the minimum value of (4) is necessarily attained at one of the endpoints of I_j. Suppose that our partition decomposes into two sub-sequences, $I_1, \ldots, I_{j_{\text{mid}}}$ and $I_{j_{\text{mid}}+1}, \ldots, I_m$ such that $\bigcup_{j=1}^{j_{\text{mid}}} I_j = [1, k_{\text{mid}}]$ and $\bigcup_{j=j_{\text{mid}}+1}^{m} I_j = [k_{\text{mid}}+1, n]$. Furthermore, we define $e_j = \min I_j$ for every $j \in [j_{\text{mid}}]$, and $e_j = \max I_j$ for every $j \in [j_{\text{mid}}+1, n]$. With this definition at hand, given the unimodality of $k \mapsto R_k(\mathcal{A})$, we obtain $f_{\mathcal{A}}(k) = \mathcal{R}_{e_j}(\mathcal{A})$ for every position $k \in I_j$. This simplification has a notable computational implication: $f_{\mathcal{A}}(k)$ can be computed with the mere knowledge of the assortments $\mathcal{A}[e_1], \ldots, \mathcal{A}[e_m]$. In the sequel, the positions $e_1, \ldots, e_m$ will be referred to as the *events* of the array of positions.

How do we utilize approximate assortments? Rather than utilizing the assortments $\mathcal{A}[e_1], \ldots, \mathcal{A}[e_m]$ corresponding to these events, we utilize the families of sub-assortments $\mathcal{S}^-(\epsilon^2)$ and super-assortments $\mathcal{S}^+(\epsilon^2)$, constructed in Lemma 2.5. More specifically, for every $j \in [m]$, let $S_j^- \in \mathcal{S}^-(\epsilon^2)$ and $S_j^+ \in \mathcal{S}^+(\epsilon^2)$ be the approximate assortments achieving the Properties 1-3 of Lemma 2.5 with respect to $\mathcal{A}[e_j]$. Consequently, we define a modified approximation $\hat{f}_{\mathcal{A}}$ as follows:

$$\hat{f}_{\mathcal{A}}(k) = \frac{\rho(S_j^-)}{1 + w(S_j^+)} .$$

This definition is motivated by the design of a lower bound on $f_{\mathcal{A}}(k) = \frac{\rho(\mathcal{A}[e_j])}{1+w(\mathcal{A}[e_j])}$. Indeed, we utilize the sub-assortment S_j^- to compute an under-estimate $\rho(S_j^-) \le \rho(\mathcal{A}[e_j])$, and the super-assortment S_j^+ to derive an over-estimate $w(S_j^+) \ge w(\mathcal{A}[e_j])$. Consequently, by combining

Properties 2 and 3 of Lemma 2.5, it immediately follows that

$$\hat{f}_{\mathcal{A}}(k) = \frac{\rho(S_j^-)}{1 + w(S_j^+)} \geq \frac{(1-\epsilon^2) \cdot \rho(\mathcal{A}[e_j])}{1 + (1+\epsilon^2) \cdot w(\mathcal{A}[e_j])} \geq \left(1 - 2\epsilon^2\right) \cdot \frac{\rho(\mathcal{A}[e_j])}{1 + w(\mathcal{A}[e_j])} = \left(1 - 2\epsilon^2\right) \cdot f_{\mathcal{A}}(k) \ . \tag{5}$$

Hence, our final value approximation $\hat{f}_{\mathcal{A}}$ constitutes a lower bound on $k \mapsto f_{\mathcal{A}}(k)$, up to a multiplicative error of $1 - 2\epsilon^2$.

Partition of the positions. To complete our definition of $\hat{f}_{\mathcal{A}}$, it remains to specify the sequence of events $e_1, \ldots, e_m$, which in turn generates the partition $I_1, \ldots, I_m$. To provide intuition, these events are defined so that the revenue function does not vary by a factor greater than $1 + \epsilon$ between two successive events. Starting with the interval $[1, k_{\text{mid}}]$, the events are defined iteratively as follows:

- Initially, $e_1 = 1$.
- Next, $e_2 = \min\{k \in [e_1 + 1, k_{\text{mid}}] : R_k(\mathcal{A}) \geq (1+\epsilon) \cdot R_{e_1}(\mathcal{A})\}$.
- In general, $e_{j+1} = \min\{k \in [e_j + 1, k_{\text{mid}}] : R_k(\mathcal{A}) \geq (1+\epsilon) \cdot R_{e_j}(\mathcal{A})\}$.
- So on and so forth, until the set $\{k \in [e_j + 1, k_{\text{mid}}] : R_k(\mathcal{A}) \geq (1+\epsilon) \cdot R_{e_j}(\mathcal{A})\}$ is empty.

Next, we let j_{mid} be the index of the last event in $[1, k_{\text{mid}}]$, and turn our attention to the interval $[k_{\text{mid}} + 1, n]$:

- Initially, $e_{j_{\text{mid}}+1} = \max\{k \in [k_{\text{mid}} + 1, n] : R_k(\mathcal{A}) \geq \frac{1}{1+\epsilon} \cdot R_{e_{j_{\text{mid}}}}(\mathcal{A})\}$.
- Next, $e_{j_{\text{mid}}+2} = \max\{k \in [e_{j_{\text{mid}}+1} + 1, n] : R_k(\mathcal{A}) \geq \frac{1}{1+\epsilon} \cdot R_{e_{j_{\text{mid}}+1}}(\mathcal{A})\}$.
- In general, $e_{j+1} = \max\{k \in [e_j + 1, n] : R_k(\mathcal{A}) \geq \frac{1}{1+\epsilon} \cdot R_{e_j}(\mathcal{A})\}$.
- So on and so forth, until we reach the last event $e_m = n$.

As a result, we construct the corresponding partition into intervals $I_1, \ldots, I_m$. To this end, let $I_j = [e_j, e_{j+1} - 1] \cap [1, k_{\text{mid}}]$ for every $j \in [j_{\text{mid}}]$, and $I_j = [e_{j-1} + 1, e_j] \cap [k_{\text{mid}} + 1, n]$ for every $j \in [j_{\text{mid}} + 1, m]$. Clearly, $[1, k_{\text{mid}}] = \bigcup_{j=1}^{j_{\text{mid}}} I_j$ and $[k_{\text{mid}} + 1, n] = \bigcup_{j=j_{\text{mid}}+1}^{m} I_j$. It is easy to verify that our definitions are consistent with our preceding discussion, in the sense that the minimal value of the revenue function within each interval I_j is attained at its corresponding event e_j. Indeed, for every $j \in [j_{\text{mid}}]$, the revenue function is non-decreasing over I_j, and the event e_j is the left endpoint of I_j. Similarly, for every $j \in [j_{\text{mid}} + 1, m]$, the revenue function is non-increasing over I_j, and the event e_j is the right endpoint of I_j.

The next claim bounds the error of our approximation method, by showing that $\hat{f}_{\mathcal{A}}(k)$ is a tight under-estimate of $R_k(\mathcal{A})$ for every $k \in [n]$. The proof is deferred to Appendix C.3. .

Lemma 3.2. *For every assignment $\mathcal{A} \in \mathcal{P}_\wedge$, we have $(1 - 4\epsilon) \cdot R_k(\mathcal{A}) \leq \hat{f}_{\mathcal{A}}(k) \leq R_k(\mathcal{A})$.*

To help unclutter notation, we do not indicate the dependency of e_j, S_j^- and S_j^+ on the assignment $\mathcal{A}$ from this point on, unless specified otherwise.

4 Dynamic Programming Formulation

In this section, we devise an approximation scheme for the display optimization problem. We formulate a dynamic program, on the basis of the value approximation developed in Section 3. Despite the approximate nature of this formulation, the optimal sequence of dynamic programming actions can be efficiently converted into analogous assignment decisions. Below, we provide a more detailed outline of our algorithmic approach.

4.1 Outline

Value approximation. Based on the discussion in Section 3.2, Lemma 3.2 immediately implies that, in order to compute an assignment whose expected revenue is within factor $1 - O(\epsilon)$ of the optimal revenue $\mathcal{R}(\mathcal{A}^*)$, it suffices to maximize the value approximation $\hat{\mathcal{R}}(\cdot)$ defined over the collection of assignments $\mathcal{A} \in \mathcal{P}_\wedge$ as follows:

$$\hat{\mathcal{R}}(\mathcal{A}) = \sum_{k \in [n]} \lambda_k \cdot \hat{f}_{\mathcal{A}}(k) \quad = \quad \sum_{j=1}^{m} \left(\sum_{k \in I_j} \lambda_k \right) \cdot \frac{\rho(S_j^-)}{1 + w(S_j^+)} \ , \tag{6}$$

where we utilize the same notation as in Section 3.2. This value approximation has noticeable computational properties. As shown by the right-hand side of equation (6), $\hat{\mathcal{R}}(\mathcal{A})$ can be computed with the mere knowledge of the events e_j along with the approximate assortments $S_j^- \in \mathcal{S}^-(\epsilon^2)$ and $S_j^+ \in \mathcal{S}^+(\epsilon^2)$, of which there are polynomially-many combinations for any fixed $j \in [m]$ by Lemma 2.5. Consequently, this value approximation motivates a natural dynamic programming formulation of the display optimization problem, where the events e_j are processed sequentially over $j \in [m]$, and the corresponding approximation assortments $(S_j^-, S_j^+) \in \mathcal{S}^-(\epsilon^2) \times \mathcal{S}^+(\epsilon^2)$ are determined through exhaustive enumeration at each step $j \in [m]$. The remainder of this section is devoted to developing this dynamic programming approach.

Initial guess. We remind the reader that the revenue function $k \mapsto R_k(\mathcal{A}^*)$ induced by the optimal assignment $\mathcal{A}^*$ was shown to be unimodal by Lemma 3.1. Consequently, we initially guess $k_{\text{mid}} \in [n]$, the position separating the two monotone parts of the revenue function $k \mapsto R_k(\mathcal{A}^*)$. Computationally-speaking, this guess can be obtained through exhaustive enumeration. Our algorithm will explicitly make use of the guess k_{mid} in order to guarantee that the revenue function of the constructed assignment is non-decreasing over $[1, k_{\text{mid}}]$ and non-increasing over $[k_{\text{mid}}, n]$.

Roadmap. Given the initial guess of k_{mid}, our algorithmic approach is described through the following ingredients. In Section 4.2, we present a dynamic programming formulation for $\hat{\mathcal{R}}(\cdot)$. Next, in Section 4.3, we describe the corresponding constraints on the set of possible transitions at each state. Lastly, in Section 5.2, we explain how the optimal sequence of dynamic programming actions is efficiently converted into an analogous assignment $\tilde{\mathcal{A}}$, returned by our algorithm.

4.2 Dynamic program: Recursive equations

To formalize the dynamic programming approach outlined in Section 4.1, each state of our recursion is described by four parameters:

- The current event $e_j \in [n]$.
- The next event $e_{j+1} \in [n]$.
- The approximate assortments $(S_j^-, S_j^+) \in \mathcal{S}^-(\epsilon^2) \times \mathcal{S}^+(\epsilon^2)$ corresponding to the event e_j.
- The approximate assortments $(S_{j+1}^-, S_{j+1}^+) \in \mathcal{S}^-(\epsilon^2) \times \mathcal{S}^+(\epsilon^2)$ corresponding to the event e_{j+1}.

In vector form, letting $\vec{S}_j = (S_j^-, S_j^+)$, each state of the recursion can be represented by $(e_j, e_{j+1}, \vec{S}_j, \vec{S}_{j+1})$. In addition, it is convenient to denote by I_j the interval of positions between two successive events in which the revenue function has monotone variations; namely, $I_j = [e_j, e_{j+1} - 1] \cap [1, k_{\text{mid}}]$ if $e_j \leq k_{\text{mid}}$, and $I_j = [e_{j-1} + 1, e_j] \cap [k_{\text{mid}}, n]$ otherwise.

Consequently, we introduce the value function $F(\cdot)$, defined through the following recursive equation:

$$
\begin{aligned}
F\left(e_j, e_{j+1}, \vec{S}_j, \vec{S}_{j+1}\right) = \quad & \max_{e_{j-1}, \vec{S}_{j-1}} \quad F\left(e_{j-1}, e_j, \vec{S}_{j-1}, \vec{S}_j\right) + \left(\sum_{k \in I_j} \lambda_k\right) \cdot \frac{\rho(S_j^-)}{1 + w(S_j^+)} \\
& \text{s.t.} \qquad (e_{j-1}, \vec{S}_{j-1}) \in \mathcal{T}\left(e_j, e_{j+1}, \vec{S}_j, \vec{S}_{j+1}\right) \ ,
\end{aligned} \tag{7}
$$

where $\mathcal{T}(e_j, e_{j+1}, \vec{S}_j, \vec{S}_{j+1})$ imposes a number of restrictions on the feasible state transitions, which are fully described in Section 4.3. In the above equation, $F(e_j, e_{j+1}, \vec{S}_j, \vec{S}_{j+1})$ should be interpreted as a partial computation of our value approximation (6), corresponding to the cumulative expected revenue starting at the event $e_1 = 1$ and ending at the current event e_j. This recursive equation involves two terms: the "reward-to-go", corresponding to the earlier state $(e_{j-1}, e_j, \vec{S}_{j-1}, \vec{S}_j)$, and the "immediate reward" that approximates the combined expected revenue contributions over the interval I_j. In the latter reward, the quantity $\frac{\rho(S_j^-)}{1+w(S_j^+)}$ serves as a proxy for the revenue function evaluated at the event e_j, according to our approximation method of Section 3.2.

Boundary case. To fully specify the value function $F(\cdot)$, it remains to handle the boundary case of the recursion, corresponding to the first event $e_1 = 1$, for which the approximate assortments take the form of a singleton, i.e., $S_1^- = S_1^+ = \{i_1\}$ for some $i_1 \in [n]$. Here, our value approximation is computed explicitly by the equation

$$
F\left(e_1, e_2, \vec{S}_1, \vec{S}_2\right) = \left(\sum_{k \in I_1} \lambda_k\right) \cdot \frac{\rho_{i_1}}{1 + w_{i_1}} \ ,
$$

where I_1 designates the interval of positions $[e_1, e_2 - 1] \cap [1, k_{\text{mid}}]$.

Hereafter, we denote by $(e_T, e_{T+1}, \vec{S}_T, \vec{S}_{T+1})$ the root of the recursion, describing the last event $e_{T+1} = e_T = n$, for which $\vec{S}_{T+1} = \vec{S}_T = ([n], [n])$. It is easy to verify that $(e_T, e_{T+1}, \vec{S}_T, \vec{S}_{T+1})$ is the state that maximizes the value function $F(\cdot)$.

4.3 Dynamic program: Feasible transitions

In this section, we specify the collection of pairs $(e_{j-1}, \vec{S}_{j-1}) \in \mathcal{T}(e_j, e_{j+1}, \vec{S}_j, \vec{S}_{j+1})$ by imposing a number of constraints over which the recursive equation (7) is maximized, thereby defining the set of feasible state transitions. This ensemble of constraints plays a pivotal role for our analysis in Section 5. Indeed, we will verify in Section 5.1 that the constraints come without incurring any loss with respect to $\hat{\mathcal{R}}(\mathcal{A}^*)$, by showing that all state transitions associated with the optimal assignment $\mathcal{A}^*$ are feasible. Furthermore, these restrictions will be crucial to guarantee in Section 5.2 that the loss of information incurred by our approximation method is "reversible", in the sense that any sequence of dynamic programming actions can be approximately matched with analogous assignments of positions to items.

Our first constraint is related to the set inclusion properties between the approximate assortments:

$$S^-_{j-1} \subseteq S^+_{j-1} \subseteq S^+_j \ . \tag{8}$$

To provide intuition about this constraint, the relationship $S^-_{j-1} \subseteq S^+_{j-1}$ is motivated by Property 1 of Lemma 2.5. Moreover, we require that the super-assortments are nested $S^+_{j-1} \subseteq S^+_j$, in light of Lemma 2.6. Note that a similar set inclusion is not enforced for the sub-assortments S^-_{j-1} and S^-_j, in line with the observation made in Section 2.2.

Our next ensemble of constraints pertains to the accuracy of the approximate assortments:

$$\rho\left(S^+_{j-1} \setminus S^-_{j-1}\right) \leq 4\epsilon^2 \cdot \rho\left(S^-_{j-1}\right) \ , \tag{9}$$

and

$$w\left(S^+_{j-1} \setminus S^-_{j-1}\right) \leq 4\epsilon^2 \cdot w\left(S^-_{j-1}\right) \ . \tag{10}$$

Here, inequalities (2) and (3) are transcribed as constraints of the dynamic program.

Moreover, we relate the cardinality of the approximate assortments to the events through the following constraint:

$$e_{j-1} \leq \min\left\{\left|S^+_{j-1}\right|, e_j - \left|S^-_j \setminus S^+_{j-1}\right|\right\} \ . \tag{11}$$

To provide intuition about the latter inequality, by Property 1 of Lemma 2.5, the cardinality of the super-assortment should greater or equal to the corresponding event, meaning that $e_{j-1} \leq |S^+_{j-1}|$. In addition, we ensure that there are sufficiently-many positions between the successive events e_{j-1} and e_j to assign all items in $S^-_j \setminus S^+_{j-1}$, meaning that $|S^-_j \setminus S^+_{j-1}| \leq e_j - e_{j-1}$.

Lastly, we guarantee that the revenue function, in approximate form, has unimodal variations, through the following constraints:

$$\begin{cases} \dfrac{\rho(S_j^-)}{1+w(S_j^+)} \geq \left(1+\dfrac{\epsilon}{2}\right) \cdot \dfrac{\rho(S_{j-1}^-)}{1+w(S_{j-1}^+)} & \text{if } k_{\text{mid}} \geq e_j \ , \quad (12) \\ \dfrac{\rho(S_{j+1}^-)}{1+w(S_{j+1}^+)} \leq \min\left\{ \left(1-\dfrac{\epsilon}{2}\right) \cdot \dfrac{\rho(S_{j-1}^-)}{1+w(S_{j-1}^+)}, \left(1+\dfrac{\epsilon}{2}\right) \cdot \dfrac{\rho(S_j^-)}{1+w(S_j^+)} \right\} & \text{if } e_j \geq e_{j-1} \geq k_{\text{mid}} \ , \quad (13) \\ \dfrac{\rho(S_j^-)}{1+w(S_j^+)} \geq \left(1-\dfrac{3\epsilon}{2}\right) \cdot \dfrac{\rho(S_{j-1}^-)}{1+w(S_{j-1}^+)} & \text{if } e_j \geq k_{\text{mid}} \geq e_{j-1} \ . \quad (14) \end{cases}$$

Consequently, $\mathcal{T}(e_j, e_{j+1}, \vec{S}_j, \vec{S}_{j+1})$ is defined as the collection of pairs $(e_{j-1}, \vec{S}_{j-1})$ that satisfy the structural relationships (8)-(14).

4.4 Assignment decisions

Having fully specified our dynamic programming formulation (7), it remains to explain how the dynamic programming actions are converted into analogous assignment decisions. More specifically, in what follows, we efficiently construct the assignment $\tilde{\mathcal{A}}$, returned by our algorithm, based on the optimal sequence dynamic programming actions. The revenue performance of $\tilde{\mathcal{A}}$ will be analyzed in Section 5.2.

By the computation of the optimal dynamic programming value $F(e_T, e_{T+1}, \vec{S}_T, \vec{S}_{T+1})$, we are given the optimal sequence of dynamic programming actions, described by the events $e_1, \ldots, e_T$, and the corresponding sub-assortments $S_1^-, \ldots, S_T^-$ and super-assortments $S_1^+, \ldots, S_T^+$. Our goal is to convert these dynamic programming actions into a feasible assignment $\tilde{\mathcal{A}} : [n] \to [n]$.

Step 1: Partial ordering over items. First, we construct a sequence of subsets $(\Delta_1, \ldots, \Delta_T)$, forming a partition of the collection of items $[n]$. Intuitively, the role of the sequence $(\Delta_1, \ldots, \Delta_T)$ is to provide a partial ordering over items for the construction of the assignment $\tilde{\mathcal{A}}$, in the sense that any item $i_1 \in \Delta_1$ will be assigned by $\tilde{\mathcal{A}}$ to a higher-ranked position than any item $i_2 \in \Delta_2$, so on and so forth. Letting $j_{\text{mid}} = \max\{j \in [T] : e_j \leq k_{\text{mid}}\}$ be the index of the last event in $[1, k_{\text{mid}}]$, the subsets $\Delta_1, \ldots, \Delta_T$ are constructed as follows:

1. We let $\Delta_1 = S_1^-$, $\Delta_2 = S_2^- \setminus (S_1^+ \cup \Delta_1)$, $\Delta_3 = S_3^- \setminus (S_2^+ \cup \Delta_1 \cup \Delta_2)$, so on and so forth, until forming the subset $\Delta_{j_{\text{mid}}} = S_{j_{\text{mid}}}^- \setminus (S_{j_{\text{mid}}-1}^+ \cup (\bigcup_{j=1}^{j_{\text{mid}}-1} \Delta_j))$.
2. Next, we let $\Delta_{j_{\text{mid}}+1} = S_{j_{\text{mid}}+1}^+ \setminus (\bigcup_{j=1}^{j_{\text{mid}}} \Delta_j)$, $\Delta_{j_{\text{mid}}+2} = S_{j_{\text{mid}}+2}^+ \setminus (\bigcup_{j=1}^{j_{\text{mid}}+1} \Delta_j)$, so on and so forth, until obtaining the last subset $\Delta_T = S_T^+ \setminus (\bigcup_{j=1}^{T-1} \Delta_j)$.

Step 2: Construction of $\tilde{\mathcal{A}}$. The assignment $\tilde{\mathcal{A}}$ first assigns the items in Δ_1 to the positions $[1, |\Delta_1|]$ by order of decreasing price, then the items in Δ_2 to the positions $[|\Delta_1|+1, |\Delta_1|+|\Delta_2|]$ again by decreasing price, so on and so forth. Formally, let $\tilde{e}_j = |\Delta_1 \cup \cdots \cup \Delta_j|$ for every $j \in [1, T]$, and let $\tilde{e}_0 = 0$. With this notation at hand, for every $j \in [1, T]$ and $t \in [1, |\Delta_j|]$, we define $\tilde{A}(\tilde{e}_{j-1} + t)$ as the t-th item of Δ_j by order of decreasing price.

Note that, since $S_T = [n]$, we have $\bigcup_{j=1}^{T} \Delta_j = [n]$, and thus, the position-to-item assignment $\tilde{\mathcal{A}}$ is fully specified over all positions in $[n]$. To provide a concrete example, suppose that $T = 3$,

$\Delta_1 = \{1\}$, $\Delta_2 = \{3\}$, $\Delta_3 = \{2,4,5\}$, and $r_i = i$. Here, we obtain the assignments $\tilde{\mathcal{A}}(1) = 1$, $\tilde{\mathcal{A}}(2) = 3$, $\tilde{\mathcal{A}}(3) = 5$, $\tilde{\mathcal{A}}(4) = 4$, and $\tilde{\mathcal{A}}(5) = 2$.

5 Performance analysis

This section is devoted to proving Theorem 1.1 by analyzing the dynamic programming algorithm developed in Section 4. First, we show in Section 5.1 that the optimal dynamic programming value is at least $\hat{\mathcal{R}}(\mathcal{A}^*)$, where $\mathcal{A}^*$ is the optimal assignment. Next, in Section 5.2, we argue that the expected revenue generated by the assignment $\tilde{\mathcal{A}}$, returned by our algorithm, nearly matches the optimal dynamic programming value, and thereby, the optimality gap is of $O(\epsilon)$.

Additional notation. Going forward, we let $e_1^*, \ldots, e_m^*$ be the events generated by the approximation method of Section 3.2 with respect to the assignment $\mathcal{A}^*$. Further, we let $S_1^{*-}, \ldots, S_m^{*-}$ and $S_1^{*+}, \ldots, S_m^{*+}$ be the sub-assortments and super-assortments corresponding to these events, respectively. In addition, we designate by $I_1^*, \ldots, I_m^*$ the intervals forming the partition of $[n]$ induced by the sequence of events. Namely, $I_j^* = [e_j^*, e_{j^*+1} - 1] \cap [1, k_{\mathrm{mid}}]$ if $e_j^* \leq k_{\mathrm{mid}}$, and $I_j^* = [e_{j-1}^* + 1, e_j^*] \cap [k_{\mathrm{mid}}, n]$ otherwise. It is worth emphasizing that, unlike k_{mid}, these parameters are not utilized by our algorithm; in what follows, they only appear for purposes of analysis. Lastly, throughout this section, we assume that our algorithm is executed with an error parameter $\epsilon \in (0, \frac{1}{5})$.

5.1 Analyzing the optimal dynamic programming value

In the next claim, we relate the optimal value of our dynamic program to the value approximation $\hat{\mathcal{R}}(\cdot)$, evaluated at the optimal assignment $\mathcal{A}^*$.

Lemma 5.1. $F(e_T, e_{T+1}, \vec{S}_T, \vec{S}_{T+1}) \geq \hat{\mathcal{R}}(\mathcal{A}^*)$.

Proof. The proof proceeds by analyzing the dynamic programming states $(e_1^*, e_2^*, \vec{S}_1^*, \vec{S}_2^*), \ldots, (e_m^*, e_{m+1}^*, \vec{S}_m^*, \vec{S}_m^*$ induced by the optimal assignment $\mathcal{A}^*$, where, for completeness, we define $e_{m+1}^* = e_m^*$ and $\vec{S}_{m+1}^* = \vec{S}_m^*$. Suppose that these state transitions are feasible in the dynamic program (7), meaning that we indeed have $(e_{j-1}^*, \vec{S}_{j-1}^*) \in \mathcal{T}(e_j^*, e_{j+1}^*, \vec{S}_j^*, \vec{S}_{j+1}^*)$ for every $j \in [2, m]$. It follows that

$$F(e_T, e_{T+1}, \vec{S}_T, \vec{S}_{T+1}) \geq \sum_{j=1}^{m} \left(\sum_{k \in I_j^*} \lambda_k \right) \cdot \frac{\rho(S_j^{*-})}{1 + w(S_j^{*+})} = \hat{\mathcal{R}}(\mathcal{A}^*)$$

where the inequality is derived by iteratively plugging the recursive equation (7) over the sequence of dynamic programming states induced by $\mathcal{A}^*$, and the equality immediately follows from our definition of the value approximation $\hat{\mathcal{R}}(\cdot)$ in equation (6).

Consequently, it remains to verify that $(e_{j-1}^*, \vec{S}_{j-1}^*) \in \mathcal{T}(e_j^*, e_{j+1}^*, \vec{S}_j^*, \vec{S}_{j+1}^*)$ for every $j \in [2, m]$. Specifically, we show that the dynamic programming constraints (8)-(14) are satisfied at each state transition.

- *Inequality* (8)*:* By Lemma 2.6, we infer that $S^{*+}_{j-1} \subseteq S^{*+}_j$ for every $j \in [2, m]$. In addition, the set inclusion $S^{*-}_{j-1} \subseteq S^{*+}_{j-1}$ immediately proceeds from Property 1 of Lemma 2.5, instantiated with $\mathcal{U} = \mathcal{A}^*[e^*_{j-1}]$ and $\delta = \epsilon^2$.

- *Inequalities* (9) *and* (10)*:* These relationships immediately follow from (2) and (3), instantiated with $\mathcal{U} = \mathcal{A}^*[e^*_{j-1}]$ and $\delta = \epsilon^2$.

- *Inequality* (11)*:* On the one hand, we have $e^*_{j-1} = |\mathcal{A}^*[e^*_{j-1}]| \leq |S^{*+}_{j-1}|$ by Property 1 of Lemma 2.5, with respect to the assortment $\mathcal{U} = \mathcal{A}^*[e^*_{j-1}]$. On the other hand, we have

$$e^*_{j-1} = e^*_j - \left(e^*_j - e^*_{j-1}\right) = e^*_j - \left|\mathcal{A}^*[e^*_j] \setminus \mathcal{A}^*[e^*_{j-1}]\right| \leq e^*_j - \left|S^{*-}_j \setminus S^{*+}_{j-1}\right| ,$$

  where the latter inequality holds since $(S^{*-}_j \setminus S^{*+}_{j-1}) \subseteq (\mathcal{A}^*[e^*_j] \setminus \mathcal{A}^*[e^*_{j-1}])$ by Property 1 of Lemma 2.5, with respect to the assortments $\mathcal{A}^*[e^*_{j-1}]$ and $\mathcal{A}^*[e^*_j]$.

- *Inequality* (12)-(14)*:* In the case where $k_{\rm mid} \geq e^*_j$, observe that

$$\begin{aligned}
\frac{\rho(S^{*-}_j)}{1 + w(S^{*+}_j)} &\geq \frac{(1 - \epsilon^2) \cdot \rho(\mathcal{A}^*[e^*_j])}{1 + (1 + \epsilon^2) \cdot w(\mathcal{A}^*[e^*_j])} \\
&\geq \left(1 - 2\epsilon^2\right) \cdot \frac{\rho(\mathcal{A}^*[e^*_j])}{1 + w(\mathcal{A}^*[e^*_j])} \\
&= \left(1 - 2\epsilon^2\right) \cdot R_{e_j}\left(\mathcal{A}^*\right) \\
&\geq \left(1 + \epsilon - 2\epsilon^2\right) \cdot R_{e_{j-1}}\left(\mathcal{A}^*\right) \\
&\geq \left(1 + \frac{\epsilon}{2}\right) \cdot \frac{\rho(\mathcal{A}^*[e^*_{j-1}])}{1 + w(\mathcal{A}^*[e^*_{j-1}])} \\
&\geq \left(1 + \frac{\epsilon}{2}\right) \cdot \frac{\rho(S^{*-}_{j-1})}{1 + w(S^{*+}_{j-1})} .
\end{aligned}$$

  Here, the first inequality proceeds from Properties 2 and 3 of Lemma 2.5, instantiated with $\mathcal{U} = \mathcal{A}^*[e^*_j]$ and $\delta = \epsilon^2$. The third inequality follows from the definition of the event $e^*_j = \min\{k \in [e^*_{j-1} + 1, k_{\rm mid}] : R_k(\mathcal{A}^*) \geq (1 + \epsilon) \cdot R_{e^*_{j-1}}(\mathcal{A}^*)\}$ in Section 3.2 over the monotone non-decreasing part of the revenue function. The next inequality holds since $\epsilon \in (0, \frac{1}{5})$. The last inequality proceeds from Property 1 of Lemma 2.5, with respect to the assortment $\mathcal{U} = \mathcal{A}^*[e^*_{j-1}]$. The other case inequalities (13) and (14) proceed from nearly identical arguments; for concision, the proofs are deferred to Appendix C.4.

$\blacksquare$ $\blacksquare$

5.2 Analyzing the assignment $\tilde{\mathcal{A}}$

In what follows, we show that the expected revenue generated by the assignment $\tilde{\mathcal{A}}$, constructed in Section 4.4 nearly matches the value approximation guaranteed by these decisions, as stated by Lemma 5.1. In particular, we exploit the structural properties enforced by the dynamic programming constraints specified in Section 4.3.

In the next two claims, we begin by analyzing how the ρ-quantities and weights mutually evolve along the sequence of positions $\tilde{e}_1, \ldots, \tilde{e}_m$ in the assignment $\tilde{\mathcal{A}}$. Recall from Section 4.4 that, for every $j \in [T]$, $\tilde{e}_j = |\Delta_1 \cup \cdots \cup \Delta_j|$.

Lemma 5.2. *For every $j \in [T]$, $\rho(\tilde{\mathcal{A}}[\tilde{e}_j]) \geq (1 - 8\epsilon) \cdot \rho(S_j^-)$.*

Lemma 5.3. *For every $j \in [T]$, $w(\tilde{\mathcal{A}}[\tilde{e}_j]) \leq w(S_j^+)$.*

The proofs of these lemmas are presented in Appendices C.5 and C.6, respectively. Based on these claims, we proceed with our main result, providing a lower bound on $\mathcal{R}(\tilde{\mathcal{A}})$ relative to the optimal expected revenue $\mathcal{R}(\mathcal{A}^*)$. Consequently, our dynamic program (7) yields an approximation scheme for the display optimization problem, thereby completing the proof of Theorem 1.1.

Lemma 5.4. $\mathcal{R}(\tilde{\mathcal{A}}) \geq (1 - 15\epsilon) \cdot \mathcal{R}(\mathcal{A}^*)$.

Proof. In order to create an explicit connection with our value approximation (6), we define the intervals $\tilde{I}_j = [\tilde{e}_j, \tilde{e}_{j+1} - 1] \cap [1, k_{\mathrm{mid}}]$ if $j \leq j_{\mathrm{mid}}$, and $\tilde{I}_j = [\tilde{e}_{j-1} + 1, \tilde{e}_j] \cap [k_{\mathrm{mid}} + 1, n]$ if $j \geq j_{\mathrm{mid}} + 1$. We show, through the next claim, that the revenue function is unimodal within the interval $[\tilde{e}_j, \tilde{e}_{j+1}]$, for every $j \in [T-1]$. The proof is deferred to Appendix C.7.

Claim 5.5. *For every $j \in [T-1]$, the function $k \mapsto R_k(\tilde{\mathcal{A}})$ is unimodal over the interval $[\tilde{e}_j, \tilde{e}_{j+1}]$.*

Consequently, we obtain

$$\begin{aligned}
\mathcal{R}\left(\tilde{\mathcal{A}}\right) &= \sum_{k=1}^{n} \lambda_k \cdot R_k\left(\tilde{\mathcal{A}}\right) \\
&\geq \sum_{j=1}^{j_{\mathrm{mid}}} \left(\sum_{k \in \tilde{I}_j} \lambda_k \right) \cdot \underbrace{\min\left\{ R_{\tilde{e}_j}\left(\tilde{\mathcal{A}}\right), R_{\tilde{e}_{j+1}}\left(\tilde{\mathcal{A}}\right) \right\}}_{\mathrm{A_j}} \\
&\quad + \sum_{j=j_{\mathrm{mid}}+1}^{T} \left(\sum_{k \in \tilde{I}_j} \lambda_k \right) \cdot \underbrace{\min\left\{ R_{\tilde{e}_{j-1}}\left(\tilde{\mathcal{A}}\right), R_{\tilde{e}_j}\left(\tilde{\mathcal{A}}\right) \right\}}_{\mathrm{B_j}} . \qquad (15)
\end{aligned}$$

where inequality (15) follows from Claim 5.5.

Now, in order to bound the terms $\mathrm{A_j}$ and $\mathrm{B_j}$, note that

$$\begin{aligned}
\min\left\{ R_{\tilde{e}_j}\left(\tilde{\mathcal{A}}\right), R_{\tilde{e}_{j+1}}\left(\tilde{\mathcal{A}}\right) \right\} &= \min\left\{ \frac{\rho(\tilde{\mathcal{A}}[\tilde{e}_j])}{1 + w(\tilde{\mathcal{A}}[\tilde{e}_j])}, \frac{\rho(\tilde{\mathcal{A}}[\tilde{e}_{j+1}])}{1 + w(\tilde{\mathcal{A}}[\tilde{e}_{j+1}])} \right\} \\
&\geq (1 - 8\epsilon) \cdot \min\left\{ \frac{\rho(S_j^-)}{1 + w(S_j^+)}, \frac{\rho(S_{j+1}^-)}{1 + w(S_{j+1}^+)} \right\} , \qquad (16)
\end{aligned}$$

where the inequality follows from Lemmas 5.2 and 5.3. Therefore, by combining (16) with constraint (12), we obtain for every $j \leq j_{\mathrm{mid}} - 1$:

$$\mathrm{A_j} \geq (1 - 8\epsilon) \cdot \min\left\{ \frac{\rho(S_j^-)}{1 + w(S_j^+)}, \frac{\rho(S_{j+1}^-)}{1 + w(S_{j+1}^+)} \right\} = (1 - 8\epsilon) \cdot \frac{\rho(S_j^-)}{1 + w(S_j^+)} .$$

Similarly, by combining (16) with constraint (14), we obtain:

$$\mathrm{A}_{\mathrm{j_{mid}}} \geq (1-8\epsilon)\cdot \min\left\{\frac{\rho(S_j^-)}{1+w(S_j^+)}, \frac{\rho(S_{j+1}^-)}{1+w(S_{j+1}^+)}\right\} \geq (1-9\epsilon)\cdot \frac{\rho(S_j^-)}{1+w(S_j^+)} \ .$$

Lastly, by combining (16) with constraint (13), we obtain for every $j \geq j_{\mathrm{mid}}+1$:

$$\mathrm{B_j} \geq (1-8\epsilon)\cdot \min\left\{\frac{\rho(S_{j-1}^-)}{1+w(S_{j-1}^+)}, \frac{\rho(S_j^-)}{1+w(S_j^+)}\right\} \geq (1-10\epsilon)\cdot \frac{\rho(S_j^-)}{1+w(S_j^+)} \ .$$

By plugging these inequalities into (15), we obtain

$$\begin{aligned}
\mathcal{R}\left(\tilde{\mathcal{A}}\right) &\geq (1-10\epsilon)\cdot\sum_{j=1}^{T}\left(\sum_{k\in\tilde{I}_j}\lambda_k\right)\cdot\frac{\rho(S_j^-)}{1+w(S_j^+)} \\
&\geq (1-11\epsilon)\cdot\sum_{j=1}^{T}\left(\sum_{k\in I_j}\lambda_k\right)\cdot\frac{\rho(S_j^-)}{1+w(S_j^+)} &(17)\\
&= (1-11\epsilon)\cdot F\left(e_T, e_{T+1}, \vec{S}_T, \vec{S}_{T+1}\right) \\
&\geq (1-11\epsilon)\cdot\hat{\mathcal{R}}\left(\mathcal{A}^*\right) &(18)\\
&= (1-11\epsilon)\cdot\sum_{k=1}^{n}\lambda_k\cdot\hat{f}_{\mathcal{A}^*}(k) \\
&\geq (1-15\epsilon)\cdot\mathcal{R}\left(\mathcal{A}^*\right)\ , &(19)
\end{aligned}$$

where inequality (17) is established in Appendix C.8, based on a comparison of the positions e_j and $\tilde{e}_j$ for every $j \in [T]$. Inequality (18) holds due to Lemma 5.1. Inequality (19) follows from the definition of our value approximation in equation (6) and Lemma 3.2. ■ ■

6 Case Study

In this section, we demonstrate the predictive power of our modeling approach using a publicly-available data set, formed by search results and hotel bookings on Expedia's online platform. In Section 6.1, we describe our empirical set-up. In Section 6.2, we provide a detailed account of the explanatory features generated from the Expedia data set. Next, in Section 6.3, we specify the tested choice models. Subsequently, we explain in Section 6.4 how the choice models are fitted to data, using maximum-likelihood estimation methods (MLE). Lastly, the numerical results are discussed in Section 6.5.

6.1 Empirical set-up

We consider a publicly-available data set that describes search results and bookings for hotel rooms on the Expedia platform. This data was released by Expedia in 2013 on the platform Kaggle, as part of a data science competition to improve their ranking engine[1]. Given a search query, the relevant hotel listings are ranked and displayed to the users, across multiple web-

[1]See url: https://www.kaggle.com/c/expedia-hotel-recommendations.

pages; due to the analogy between rankings and positions, this setting is a canonical application of choice models with location effects. Consequently, our goal is to compare the consideration set-based approach, introduced in Section 1.1, against a utility-based choice model with location bias, which stands as the most natural alternative proposed in related literature (Davis et al. 2013, Abeliuk et al. 2016). In both cases, users' relative preferences over hotel listings are prescribed by similar feature-based MNL models, and the choice models have precisely the same number of parameters. However, these choice models differentiate themselves through the incorporation of location effects. In our model, location effects are captured by the nested structure of the consideration sets. In contrast, in the benchmark model, location effects are specified through an augmented utility function, with additional explanatory variables describing the position assigned to each item. As such, our experiments will enable us to evaluate the benefits of utilizing a consideration set-based modeling approach in order to capture location effects.

6.2 Data description

Features. Each observation in the Expedia data set corresponds to a unique hotel search; notation-wise, an observation is indexed by $t \in \mathcal{T}$, where $\mathcal{T}$ is the collection of all observations. For each observation $t \in \mathcal{T}$, the data set provides a rich set of contextual features. First, various parameters of the search query are recorded, such as length of stay, number of rooms, number of adults, etc. The features generated from search data are summarized in Table 1.

Trip_t:	Number of days before the beginning of the trip.
Stay_t:	Trip length (i.e., number of hotel nights).
$\#\ \text{Rooms}_t$:	Number of rooms.
$\#\ \text{Adults}_t$:	Number of adults.
$\#\ \text{Children}_t$:	Number of children.
Occupancy_t :	Ratio between the number of travelers and the number of rooms.
Is Weekend_t :	Indicates whether the trip includes a weekend.
$\text{Occupancy}_t \times \text{Trip}_t$:	Interaction term between the occupancy and number of days before the trip.

Table 1: Search features.

Next, we have access to the set of hotel listings S_t displayed to the user, formed by up to 38 alternatives. For each hotel listing $j \in S_t$, there are several descriptive features, including gross booking price, display ranking, average review score, etc. The complete list of features generated from hotel information is presented in Table 2. It is worth noting that the display ranking feature is encoded through the binary variables $\text{Display Ranking}_{jt}(k)$, indicating whether the ranking of hotel listing j is within $[5(k-1)+1, 5k]$, for every $k \in [8]$. In other words, rankings are bucketed into intervals formed by 5 consecutive positions.

Lastly, we have access to the hotel listing that was ultimately booked, if any. In the sequel, this choice outcome is described by the binary variables $z_{j,t} \in \{0,1\}$ for every $j \in S_t$. Note that $\sum_{j \in S_t} z_{j,t}$ is potentially equal to 0, in case none of the hotel options is booked.

Data pre-processing. Estimating the effects of location on user choice behavior from historical data is generally challenging since display rankings are endogenous: Expedia's ranking

Price_{jt}:	Nightly hotel rate for option $j \in S_t$.
Display $\text{Ranking}_{jt}(k)$:	Indicates whether the rank of hotel j is within $[5(k-1)+1, 5k]$, for each $k \in [8]$.
Avg Review Ratings_{jt} :	Average rating for the hotel in client reviews.
Location Score_{jt} :	Location score determined by Expedia.
Avg Hotel Price_j :	Average historical price of the hotel on a period preceding the experiment.
Is Promotion_{jt} :	Indicator of whether the hotel listing is under promotion.

Table 2: Hotel listing features.

engine uses the contextual information at hand to rank the hotel listings, potentially in a personalized way. Interestingly, a fraction of the data released by Expedia was generated though an experiment, where display rankings were fully randomized. As such, this data enables an almost ideal experimental setting to study the effects of location on users' choice behavior, as noted by Ursu (2018). Consequently, in our experimental study, we restrict attention to the data observations generated under these randomized rankings. The empirical analysis of Ursu (2018) further shows that price endogeneity is not a concern in this setting, since the control variables explain nearly 80% of price variability, and thus, conditional on the search parameters, price is unlikely to be correlated with the utility error term. As such, we perform minimal preprocessing of the data, only dropping observations where at least one of the features of interest is missing. In addition, we discovered that for a small number of observations, the booking prices seem abnormally high and might correspond to corrupted entries. Consequently, we filter out any observation where the nightly room price is greater than \$5000. Finally, we partition the Expedia searches by country websites, and restrict our attention to the 5 most popular ones by number of searches. These preprocessing steps yield 5 data sets of varying sizes.

6.3 Model specification

We proceed with a description of the two tested choice models. Hereafter, L-MNL designates the MNL model with location bias, which serves as a benchmark, while D-MNL is an instance of our choice model, introduced in Section 1.1 as part of the display optimization problem.

L-MNL model. The deterministic component of the utility generated by each choice alternative $j \in S_t$ is expressed as a linear combination of all features appearing in Tables 1 and 2:

$$\begin{aligned} u_{jt}^{\text{L-MNL}}(\vec{\alpha}) \quad &= \quad \alpha_0 + \alpha_1 \cdot \text{Trip}_t + \cdots + \alpha_8 \cdot \text{Occupancy}_t \times \text{Trip}_t + \alpha_9 \cdot \text{Price}_{jt} \\ &\quad + \alpha_{10} \cdot \text{Display Ranking}_{jt}(0) + \ldots + \alpha_{21} \cdot \text{Is Promotion}_{jt} , \end{aligned} \tag{20}$$

where $\vec{\alpha} = (\alpha_0, \ldots, \alpha_{21})$ is a real-valued vector, describing the L-MNL choice model parameters. Having specified the utility function $u_{jt}^{\text{L-MNL}}(\vec{\alpha})$, the choice probability $p_{\vec{\alpha}}(t,j)$ of hotel listing $j \in S_t$ is given by:

$$p_{\vec{\alpha}}(t,j) = \frac{e^{u_{jt}^{\text{L-MNL}}(\vec{\alpha})}}{1 + \sum_{i \in S_t} e^{u_{it}^{\text{L-MNL}}(\vec{\alpha})}} .$$

D-MNL model. As further elaborated in Section 1.1, our choice model is fully specified by a distribution over consideration sets $(\lambda_1, \ldots, \lambda_n)$ along with item weights $\{w_i : i \in [n]\}$. In the

D-MNL model, these weights are defined through a utility function similar to (20), using the search and hotel listing features generated in Section 6.2. More specifically, the utility function is expressed as a linear combination of all features appearing in Tables 1 and 2, at the exception of the binary variables $\{\text{Display Ranking}_{jt}(k)\}_{k\in[8]}$ encoding the location effects, namely

$$\begin{aligned} u_{jt}^{\text{D-MNL}}\left(\vec{\beta}\right) &= \beta_0 + \beta_1 \cdot \text{Trip}_t + \cdots + \beta_8 \cdot \text{Occupancy}_t \times \text{Trip}_t + \beta_9 \cdot \text{Price}_{jt} \\ &\quad + \beta_{10} \cdot \text{Avg Review Ratings}_{jt} + \ldots + \beta_{13} \cdot \text{Is Promotion}_{jt} \ , \end{aligned}$$

where $\vec{\beta} = (\beta_0, \ldots, \beta_{13})$ is a parameter vector of the D-MNL model corresponding to the intercept and the coefficients of the features in the utility function. On the other hand, for simplicity, the probabilities $(\lambda_1, \ldots, \lambda_n)$ are treated as exogenous quantities, not related to search and hotel features. That said, in order to mirror the structure of the L-MNL model specified by equation (20), we require that $\lambda_i = 0$ for every position $i \in [n] \setminus \{5, 10, \ldots, 40\}$. Consequently, by restricting attention to non-zero entries and re-indexing the consideration sets, our model can be described by a collapsed probability vector $\vec{\lambda} = (\lambda_1, \ldots, \lambda_8)$. Clearly, this structural restriction implies that L-MNL and D-MNL have precisely the same number of parameters, since D-MNL is uniquely specified by the pair $(\vec{\beta}, \vec{\lambda}) \in \mathbb{R}^{14} \times \Lambda$, where $\Lambda = \{\vec{\lambda} \in \mathbb{R}_+^8 : \sum_{k=1}^{8} \lambda_k \leq 1\}$. With these parameters at hand, the choice probability $p_{\vec{\beta},\vec{\lambda}}(t, j)$ of hotel listing $j \in S_t$ is given by:

$$p_{\vec{\beta},\vec{\lambda}}(t, j) = \sum_{k=1}^{8} \lambda_k \cdot \frac{e^{u_{jt}^{\text{D-MNL}}(\vec{\beta})}}{1 + \sum_{i \in S_t^k} e^{u_{it}^{\text{D-MNL}}(\vec{\beta})}} \ ,$$

where each consideration set S_t^k is formed by the hotel listings of S_t displayed in the $5k$ highest-ranked positions.

6.4 Parameter estimation

Here, we present the estimation methods utilized to fit the choice models introduced in Section 6.3, based on the maximum-likelihood estimation (MLE) principle.

L-MNL model. The L-MNL choice model is fitted to data through standard MLE techniques, which possess desirable statistical properties (McFadden 1973). That is, the MLE estimator is computed by solving the following problem:

$$\max_{\vec{\alpha} \in \mathbb{R}^{22}} \sum_{t \in \mathcal{T}} z_{j,t} \cdot \log\left(p_{\vec{\alpha}}(t, j)\right) \ . \tag{21}$$

It is well-known that the latter can be formulated as a convex optimization problem (McFadden 1973), and thereby, the MLE estimator can be computed efficiently. To this end, we implement a stochastic gradient-descent algorithm, using the Tensorflow package (Abadi et al. 2016). By executing our gradient-descent method with 30 random initializations, we pick the parameter estimate that maximizes the in-sample log-likelihood out of all those computed.

D-MNL model. The D-MNL choice model is fitted to data using MLE-based estimation as well. The MLE estimator is defined through the following optimization problem:

$$\max_{(\vec{\beta},\vec{\lambda})\in\mathbb{R}^{14}\times\Lambda} \sum_{t\in\mathcal{T}} z_{j,t}\cdot\log\left(p_{\vec{\beta},\vec{\lambda}}(t,j)\right) . \tag{22}$$

Unfortunately, in contrast to (21), this MLE problem is not necessarily convex. In fact, the D-MNL choice model can be viewed as a special case of a mixture of MNLs, for which the log-likelihood function generally exhibits multiple local minima. As such, we employ the standard expectation-maximization (EM) algorithm that computes approximate MLE estimates for mixtures of MNLs. We refer the reader to the book by Train (2009, Chap. 14.2), where this EM algorithm is presented in full generality. Starting with initial parameters $\vec{\lambda}^{(0)}$ and $\vec{\beta}^{(0)}$, as well as a prior on the realizations of the consideration sets $\vec{\pi}_t^{(0)}$ for each observation $t\in\mathcal{T}$, our EM algorithm iteratively computes parameters of the D-MNL model though posterior updates, using a convex surrogate of the log-likelihood function. The iterations of the EM algorithm are formally described in Appendix D. Consequently, for every $\ell\geq 1$, the parameters computed at the ℓ-th step of the EM algorithm are denoted by $(\vec{\beta}^{(\ell)},\ \vec{\lambda}^{(\ell)})$. After computing 30 iterations of the EM algorithm, we pick the parameter estimate $(\vec{\beta}^{(\ell)},\vec{\lambda}^{(\ell)})$ that maximizes the in-sample log-likelihood over all $\ell\in[30]$. This number of iterations was chosen so that the running times allotted to the estimation of the L-MNL and D-MNL models are nearly equivalent. Indeed, the main computational bottleneck of the EM-algorithm is a convex optimization problem having the same structure as (21).

6.5 Numerical results

The results of our experiments on each Expedia site are summarized in Table 3. Specifically, we report the normalized log-likelihood computed on hold-out data sets, measuring the out-of-sample predictive ability of the fitted L-MNL and D-MNL models. The first column identifies the Expedia site. Columns two and three specify the maximum cardinality of the assortment and the number of observations within each data set. Column four corresponds to the out-of-sample normalized log-likelihood $\mathcal{L}^L$ of the fitted L-MNL model. Column five corresponds to the out-of-sample normalized log-likelihood $\mathcal{L}^D$ of the fitted D-MNL model. The last column describes the percentage improvement of the D-MNL model over the L-MNL model, defined as $\frac{\mathcal{L}^D-\mathcal{L}^L}{\mathcal{L}^L}$. Each reported entry is the average normalized log-likelihood over 10 trials of our experiment. Namely, in each trial on a given Expedia site, we perform a 75/25 train/test split, where the hold-out data set is formed by the last 25% historical observations; next, we estimate the choice models using the MLE-based methods of Section 6.4, and compute the out-of-sample normalized log-likelihood of the fitted L-MNL and D-MNL models.

From Table 3, we infer that the D-MNL fits are more accurate than the L-MNL fits on all Expedia sites. The magnitude of improvement ranges from 0.50% to 1.95%. Moreover, all measured improvements are statistically significant ($p = 0.05$). It is worth highlighting that L-MNL and D-MNL only differentiate themselves through the incorporation of location effects, based on hotel rankings, which is one out of 15 features explaining the users' choice preferences. Further, as explained in Section 6.3, the two choice models have the same number of parameters.

Data set	Max. $\lvert S_t \rvert$	$\lvert \mathcal{T} \rvert$	L-MNL	D-MNL	% Improvement
Expedia Site 1	38	87818	0.808	0.804	0.50%
Expedia Site 2	36	14811	0.731	0.721	1.40%
Expedia Site 3	36	9751	0.730	0.725	0.69%
Expedia Site 4	32	7303	0.693	0.688	0.73%
Expedia Site 5	36	5956	0.573	0.562	1.95%

Table 3: Out-of-sample normalized log-likelihood of fitted L-MNLs and D-MNLs.

In light of these structural similarities, our results provide empirical evidence about the practical relevance of a consideration set-based choice model, and its fitting potential in online search applications.

7 Computational Experiments

In this section, we study the practical performance of our algorithm through computational experiments on randomly generated instances. These include a comparison to a suitable adaptation of the product framing algorithm of Gallego et al. (2016) and to several natural heuristic approaches proposed in previous literature. It is worth pointing out that the worst-case performance of these heuristics can be arbitrarily bad, as shown in Appendix A.

7.1 Generative model

Our simulations are run when the number of items n takes one of the values 30 and 300. Next, the additional instance parameters are randomly generated as follows:

- The MNL preference weights are set as $w_i = \theta \cdot u_i$, where the u_i values are uniformly sampled in the interval $[0, 1]$, and θ is a scaling parameter. Specifically, we wish to test different regimes of preference weights, where customers are likely to make purchase decisions (i.e., the outside option is not extremely strong). By picking $\theta = 2/n$, $\theta = 8/n$, or $\theta = 18/n$ we generate settings where the most patient customer type (with largest consideration set) makes a purchase with an expected probability of 50%, 80%, and 90%.
- The revenue parameters r_i are generated by independent samples of a standard log-normal distribution, with $\mu = 0$ and $\sigma \in \{0.3, 1.0\}$.
- We generate the position capacities $c_1, \dots, c_k$ by randomly partitioning the array of n positions into $k_\alpha = \alpha \cdot n$ parts, where $\alpha \in \{\frac{1}{3}, \frac{2}{3}\}$. Next, the sequence of position probabilities $\lambda_1, \dots, \lambda_k$ is generated by k independent samples of the $[0, 1]$-uniform distribution, which are then normalized to sum up to 1.

As one can empirically observe later on, the scaling parameter θ is the most important determinant for the practical performance of the different algorithms tested. To build intuition, when the preference weights have very small values, the MNL expected revenue function $S \mapsto R(S)$ is close to being linear in the elements of the assortment S. In particular, the monotone non-increasing part of the revenue function is expected to be of limited size and significance. In such settings, simple heuristics typically work well. On the contrary, when the preference weights get larger, the non-linearity of the expected revenue and the unimodal behavior of the revenue function prevail, and in turn, the performance of simple heuristics generally degrades.

7.2 Tested heuristics

The empirical performance of our algorithm is compared against four natural benchmarks: a local search procedure, a discrete-greedy algorithm, a heuristic based on simple priority rules, and an adaptation of the constant-factor approximation *NEST+* for the product framing problem, devised by Gallego et al. (2016). Given the experiments conducted in their paper, *NEST+* can be viewed as being state-of-the-art from a practical perspective. We proceed by explaining how each of these procedures is defined, with additional implementation details.

Local search (LS). This algorithm consists in sequentially improving the expected revenue, where at each step the current positions of two items are swapped, until reaching a local maximum. We start with an initial positions-to-items assignment, and iteratively implement the best pairwise swap between items, i.e., one that generates the largest incremental increase in the expected revenue.

Specifically, we define $\mathcal{S}$ as the collection of all pairs of distinct positions, i.e., $\mathcal{S} = \{(i,j) : 1 \leq i < j \leq n\}$. Letting $\mathcal{A}^{(k)}$ be the assignment reached at the end of step k, a feasible swap represents a pair of positions $(i,j) \in \mathcal{S}$ that are exchanged to produce the assignment $\mathcal{A}^{(k)}_{i \leftrightarrow j}$, where $\mathcal{A}^{(k)}_{i \leftrightarrow j}(i) = \mathcal{A}^{(k)}(j)$ and $\mathcal{A}^{(k)}_{i \leftrightarrow j}(j) = \mathcal{A}^{(k)}(i)$, while all other positions remain unchanged. With this definition, we proceed to step $k+1$ with the assignment that maximizes $\mathcal{R}(\mathcal{A}^{(k)}_{i \leftrightarrow j})$. In order to balance between performance and running time, the algorithm terminates when the incremental increase in the expected revenue falls below a factor of 0.1%, i.e., $\mathcal{R}(\mathcal{A}^{(k+1)})/\mathcal{R}(\mathcal{A}^{(k)}) \leq 1.001$. Finally, the initial positions-to-items assignment $\mathcal{A}^{(0)}$ is generated by picking a random permutation over the items.

Discrete-greedy (DG). The second approach we implement is a greedy heuristic where items are assigned iteratively over the positions $1, \ldots, n$, by selecting at each step k an unassigned item that maximizes the expected revenue due to the assortment $\mathcal{A}[k]$. That is, $\mathcal{A}[k]$ is formed by assigning position k to the item $i \in [n] \setminus \mathcal{A}[k-1]$ that maximizes $R(\mathcal{A}[k-1] \cup \{i\})$.

Priority-based heuristics. We examine two heuristics based on common-sense priority rules:

- *PH1:* The items are ranked by decreasing $\rho_i = r_i w_i$ quantities.
- *PH2:* The items are ranked by decreasing r_i quantities.

As argued by Gallego et al. (2016), these heuristics can be viewed as a reasonable proxy for the priority rules commonly used in the industry to sort search content.

The NEST+ algorithm. Finally, we adapt the approximation algorithm devised by Gallego et al. (2016) for the product framing problem, where positions can be left vacant. Specifically, we implement their improved algorithm, *NEST+*, where the basic $6/\pi^2$-approximation is coupled with a greedy procedure that fills vacant positions. Due to the latter feature, this algorithm naturally extends to our setting, where all positions have to be filled. Consequently, the resulting algorithm proceeds as follows: Initially, for every $k \in [n]$, we solve the capacitated assortment optimization problem with capacity k to compute the assortment $\mathcal{S}_k^*$ using the parametric search approach of Rusmevichientong et al. (2010). Next, we partially construct the assignment $\mathcal{A}_k : [k] \mapsto \mathcal{S}_k^*$, where items in $\mathcal{S}_k^*$ are ranked by their contribution to the expected revenue $R(\mathcal{S}_k^*)$. The remaining positions $[n] \setminus [k]$ are greedily filled with items in $[n] \setminus \mathcal{S}_k^*$, such that the next item to be picked maximizes the expected revenue of the resulting assortment. Finally, to arrive at a single assignment, we choose the one that maximizes $\mathcal{R}(\mathcal{A}_k)$ over all $k \in [0, n]$. It is worth noting that the performance of *NEST+* can only be better than that of the discrete-greedy heuristic, as the latter corresponds to the special case of choosing $k = 0$.

Relative optimality gap. Due to the large-scale instances we consider (up to $n = 300$), an optimal solution cannot be computed in practice through exhaustive enumeration, where all $n!$ possible assignments are examined. In fact, this approach results in exorbitant running times, which exceed one hour even for $n = 30$. Therefore, it is not possible to directly compute the optimal expected revenue for each of the instances tested. To compare our algorithm against the above-mentioned heuristics, we make use of the best solution available across all heuristics to compute their relative optimality gaps. For example, suppose that the local search heuristic returns an assignment whose expected revenue is 2, our algorithm returns an assignment with revenue 2.5, while all other heuristics generate an expected revenue of 1.8. Then, we report an optimality gap of 20% for the local search heuristic, 0% for our algorithm, and 28% for the remaining heuristics.

7.3 Implementation of our approximation scheme

Modified algorithm. In order to achieve better computational performance, our dynamic programming formulation (see Section 4) is implemented in slightly modified form. In light of Theorem 1.1, the performance of our algorithm is governed by the error parameter $\epsilon > 0$. Small values of ϵ reduce the error incurred by initially rounding the weights and ρ-quantities, following the instance alterations of Section 2.1. At the same time, the parameter ϵ governs the number of distinct approximate assortments that describe the states of the dynamic program. As explained in Appendix C.1, these assortments are constructed through efficient enumeration, by considering items in K consecutive weight classes and ρ-classes, where $K = \lceil \log_{(1+\epsilon^2)}(\frac{n}{\epsilon^2}) \rceil$. However, for $\epsilon = 0.5$, we have $K \approx 21$ when $n = 30$, and $K \approx 31$ when $n = 300$. As such, for small values of ϵ, an enumeration over all approximate assortments is not feasible in practice, as we target a running time in the order of a few minutes.

Consequently, we modify our algorithm by decoupling the parameters K and ϵ. That is, in the modified algorithm, the relationship $K = \lceil \log_{(1+\epsilon^2)}(\frac{n}{\epsilon^2}) \rceil$ is no longer satisfied. Specifically, we pick $\epsilon = 0.4$, while $K = 5$ for small-scale instances ($n = 30$), and $K = 2$ for large-scale instances ($n = 300$); the choice of these specific numerical values will be justified in the sequel. Intuitively, this decoupled design allows us to combine a compact enumeration over all approximate assortments, governed by the parameter K, with fine-grained initial rounding procedures, controlled by the parameter ϵ. The performance guarantee established by

Theorem 1.1 no longer holds for the modified algorithm, since our analysis critically relies on having $K = \lceil \log_{(1+\epsilon^2)}(\frac{n}{\epsilon^2}) \rceil$. That said, this modification will enable excellent trade-offs between performance and speed.

Parameter tuning. The specific values for K and ϵ were tuned as follows. On small-scale instances, we have evaluated the loss in optimality incurred by the transformations of Section 2.2, through exhaustive enumeration over all possible position-to-item assignments. Empirically, we found that $\epsilon = 0.4$ is the largest value of ϵ for which the optimality gap is consistently smaller than 1% across all generative settings used in our experiments. Next, having specified $\epsilon = 0.4$, the choice of K was solely governed by running time considerations. Specifically, the precise value of K was picked so that the running time is in the order of a few seconds. It is worth remarking that a similar tuning procedure can be adopted in other empirical settings.

7.4 Results

The experiments described above were conducted on a standard desktop with 2.8GHz Intel Core i5 processor and 32GB of RAM. The algorithms were implemented using the Python programming language. It is worth mentioning that improved computational performance can be achieved using low-level programming languages, such as C/C++. In Tables 4 and 5, ADP is our approximate dynamic program, LS designates the local search procedure, *NEST+* is the approximation algorithm of Gallego et al. (2016), DG refers to the discrete-greedy algorithm, PH1 is the ρ-based priority priority heuristic, and PH2 is the price-based priority heuristic. Table 4 provides statistics regarding relative optimality ratios, while Table 5 presents those related to the computational aspects and running times associated with our implementations.

Overall, the approximate dynamic program and the local search heuristic emerge as the strongest algorithms revenue-wise. Interestingly, the practical performance of the local search approach is somewhat surprising, given the $\Omega(n)$-bound on its approximation ratio, established in Appendix A. On average, our algorithm outperforms the local search heuristic by 1.5%, and by up to 3.3%, over the instances tested.

While *NEST+* exhibits near-optimal performance for small θ values, its incurred gap increases with θ, reaching 12% when $\theta = 18/n$. The latter regime corresponds to the case where the largest consideration sets entail a purchase decision with an expected probability of 90%. This observation relates to the differences between the display optimization problem and the product framing model. When θ is small, the outside option is influential, thus the monotone non-decreasing part occupies almost all array positions. In this case, only the tail items have to be filled greedily. On the contrary, when the outside option is weak, a large number of empty positions have to be filled. Similarly, the discrete-greedy approach, whose performance is always dominated by that of *NEST+*, turns out to be less effective in this regime. Moreover, we observe that a larger variability of prices ($\sigma = 1.0$) is detrimental to the algorithms *NEST+*, DG and PH1.

On average, the priority heuristics PH1 and PH2 have optimality gaps of about 8.5% and 11.5%, respectively. These results suggest that there is significant headroom when using common-sense practices, and that overlooking choice substitution and consideration set effects

Parameters						Average optimality gaps (%)					
n	K	σ	θ	k	#	ADP	LS	NEST+	DG	PH1	PH2
30	5	0.3	$2/n$	0.33	50	$< 10^{-2}$	0.16	0.18	3.10	0.97	26.76
30	5	0.3	$2/n$	0.66	50	$< 10^{-2}$	0.27	0.10	2.05	1.07	22.80
30	5	1.0	$2/n$	0.33	50	$< 10^{-2}$	0.15	1.03	26.81	2.14	7.95
30	5	1.0	$2/n$	0.66	50	$< 10^{-2}$	0.24	0.89	15.71	2.44	6.30
30	5	0.3	$8/n$	0.33	50	0.01	0.15	1.05	8.89	7.51	15.41
30	5	0.3	$8/n$	0.66	50	0.02	0.20	0.78	7.34	7.83	13.86
30	5	1.0	$8/n$	0.33	50	0.02	0.01	4.71	39.43	10.12	6.58
30	5	1.0	$8/n$	0.66	50	0.02	0.21	2.79	28.85	9.55	4.08
30	5	0.3	$18/n$	0.33	50	0.02	0.12	1.23	13.16	11.94	11.90
30	5	0.3	$18/n$	0.66	50	0.02	0.20	1.02	10.69	11.69	13.60
30	5	1.0	$18/n$	0.33	50	$< 10^{-2}$	0.09	5.91	44.99	16.06	4.85
30	5	1.0	$18/n$	0.66	50	$< 10^{-2}$	0.20	4.96	33.67	16.31	4.87
300	2	0.3	$2/n$	0.33	20	$< 10^{-2}$	2.80	0.35	2.45	0.99	25.77
300	2	0.3	$2/n$	0.66	20	$< 10^{-2}$	2.84	0.29	1.92	0.94	26.28
300	2	1.0	$2/n$	0.33	20	$< 10^{-2}$	2.59	2.34	29.36	2.77	6.57
300	2	1.0	$2/n$	0.66	20	$< 10^{-2}$	2.70	1.94	19.43	2.93	7.46
300	2	0.3	$8/n$	0.33	20	$< 10^{-2}$	3.19	1.21	9.93	7.81	15.72
300	2	0.3	$8/n$	0.66	20	$< 10^{-2}$	3.03	0.91	8.83	7.93	16.30
300	2	1.0	$8/n$	0.33	20	$< 10^{-2}$	2.38	8.84	44.98	11.95	4.46
300	2	1.0	$8/n$	0.66	20	$< 10^{-2}$	2.38	6.34	32.07	11.14	4.03
300	2	0.3	$18/n$	0.33	20	$< 10^{-2}$	3.16	2.05	15.00	12.32	11.88
300	2	0.3	$18/n$	0.66	20	$< 10^{-2}$	3.32	1.51	12.35	12.66	11.40
300	2	1.0	$18/n$	0.33	20	$< 10^{-2}$	2.41	12.51	52.87	16.52	3.54
300	2	1.0	18/n	0.66	20	$< 10^{-2}$	2.46	9.06	37.89	17.55	4.34

Table 4: Revenue performance of the algorithms tested

can be detrimental to revenue in realistic regimes of parameters. Interestingly, the two priority rules have a complementary performance that is highly correlated with the scaling factor θ. When preference weights are small, the function that maps any assortment to its expected revenue, $S \mapsto R(S)$, becomes approximately additive, in which case it is well-approximated by the sum of ρ-quantities. In this regime, the monotone decreasing part is expected to be of limited length, making the ρ-based priority rule close to optimal. On the contrary, when preference weights are large, the non-linearity effects and the monotone non-decreasing part are prevalent. In such settings, ranking by price is a more effective priority rule.

Parameters						Average running time (sec.)					
n	K	σ	θ	k	#	ADP	LS	NEST+	DG	PH1	PH2
30	5	0.3	$2/n$	0.33	50	6.83	0.06	0.13	0.00	0.00	0.00
30	5	0.3	$2/n$	0.66	50	6.87	0.06	0.12	0.00	0.00	0.00
30	5	0.3	$8/n$	0.33	50	6.09	0.06	0.20	0.00	0.00	0.00
30	5	0.3	$8/n$	0.66	50	6.65	0.06	0.19	0.00	0.00	0.00
30	5	0.3	$18/n$	0.33	50	6.49	0.06	0.22	0.00	0.00	0.00
30	5	0.3	$18/n$	0.66	50	7.23	0.07	0.23	0.00	0.00	0.00
300	2	0.3	$2/n$	0.33	20	5.05	27.71	158.04	0.50	0.00	0.00
300	2	0.3	$2/n$	0.66	20	5.02	28.22	153.85	0.51	0.00	0.00
300	2	0.3	$8/n$	0.33	20	4.99	22.52	372.26	0.50	0.00	0.00
300	2	0.3	$8/n$	0.66	20	5.02	24.85	362.24	0.50	0.00	0.00
300	2	0.3	$18/n$	0.33	20	4.99	28.22	449.43	0.50	0.00	0.00
300	2	0.3	$18/n$	0.66	20	5.00	26.92	459.77	0.50	0.00	0.00

Table 5: Running times

We conclude by turning our attention to running times, which are reported in Table 5 for $\sigma = 0.3$. To avoid redundancy, we omit the figures for $\sigma = 1.0$, which are nearly identical. These results support the practicality of the two configurations proposed for executing our modified algorithm, ADP. When the number of items is small ($n = 30$), we achieve near-optimal revenue performance with a computationally intensive state space description, requiring $K = 5$ weight classes. In this context, our approximation scheme is more computationally intensive than all other algorithms. In contrast, when the number of items is large ($n = 300$), our algorithm provides an excellent revenue vs. speed tradeoff, with only $K = 2$ item classes. The reported running times, in the order of a few seconds, suggest that our implemented algorithm can be readily utilized in offline applications, such as shelf-space allocation for brick-and-mortar retailing. Implementation by online platforms, where assortments are computed and displayed to users within milliseconds, requires further engineering and algorithmic enhancements, which are left for future research.

References

Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, et al. (2016) Tensorflow: A system for large-scale machine learning. *12th Symposium on Operating Systems Design and Implementation*, 265–283.

Abeliuk A, Berbeglia G, Cebrian M, Van Hentenryck P (2016) Assortment optimization under a multinomial logit model with position bias and social influence. *4OR* 14(1):57–75.

Agarwal A, Hosanagar K, Smith MD (2011) Location, location, location: An analysis of profitability of position in online advertising markets. *Journal of Marketing Research* 48(6):1057–1073.

Aouad A, Farias V, Levi R (2015) Assortment optimization under consider-then-rank choice models, working paper, available as SSRN report #2618823.

Ben-Akiva ME, Lerman SR (1985) *Discrete choice analysis: Theory and application to travel demand* (MIT press).

Bettman JR, Luce MF, Payne JW (1998) Constructive consumer choice processes. *Journal of Consumer Research* 25(3):187–217.

Blanchet JH, Gallego G, Goyal V (2016) A markov chain approximation to choice modeling. *Operations Research* 64(4):886–905.

Breugelmans E, Campo K, Gijsbrechts E (2007) Shelf sequence and proximity effects on online grocery choices. *Marketing Letters* 18(1-2):117–133.

Bront JJM, Méndez-Diaz I, Vulcano G (2009) A column generation algorithm for choice-based network revenue management. *Operations Research* 57(3):769–784.

Chandon P, Hutchinson JW, Bradlow ET, Young SH (2009) Does in-store marketing work? Effects of the number and position of shelf facings on brand attention and evaluation at the point of purchase. *Journal of Marketing* 73(6):1–17.

Chandukala SR, Kim J, Allenby GM, Otter T (2008) *Choice models in marketing: Economic Assumptions, Challenges and Trends* (Now Publishers Inc).

Davis JM, Gallego G, Topaloglu H (2013) Assortment planning under the multinomial logit model with totally unimodular constraint structures. Working paper.

Davis JM, Gallego G, Topaloglu H (2014) Assortment optimization under variants of the nested logit model. *Operations Research* 62(2):250–273.

Davis JM, Topaloglu H, Williamson DP (2015) Assortment optimization over time. *Operations Research Letters* 43:608–611.

Désir A, Goyal V, Zhang J (2014) Near-optimal algorithms for capacity constrained assortment optimization. Working paper. Available online as SSRN report #2543309.

Dreze X, Hoch SJ, Purk ME (1995) Shelf management and space elasticity. *Journal of Retailing* 70(4):301–326.

Farias V, Jagabathula S, Shah D (2013) A non-parametric approach to modeling choice with limited data. *Management Science* 59(2):305–322.

Feldman J, Topaloglu H (2017) Capacitated assortment optimization under the multinomial logit model with nested consideration sets. *Operations Research* 66(2):380–391.

Feldman J, Zhang D, Liu X, Zhang N (2018) Taking assortment optimization from theory to practice: Evidence from large field experiments on Alibaba. Working paper. Available online as SSRN report #3232059.

Feldman JB, Topaloglu H (2015) Bounding optimal expected revenues for assortment optimization under mixtures of multinomial logits. *Production and Operations Management* 24(10):1598–1620.

Ford LR (1957) Solution of a ranking problem from binary comparisons. *American Mathematical Monthly* 64(8):28–33.

Frank RE, Massy WF (1970) Shelf position and space effects on sales. *Journal of Marketing Research* 7(1):59–66.

Gallego G, Li A, Truong VA, Wang X (2016) Approximation algorithms for product framing and pricing. *Operations Research* Forthcoming.

Ghose A, Ipeirotis PG, Li B (2014) Examining the impact of ranking on consumer behavior and search engine revenue. *Management Science* 60(7):1632–1654.

Ghose A, Yang S (2009) An empirical analysis of search engine advertising: Sponsored search in electronic markets. *Management Science* 55(10):1605–1622.

Grover R, Vriens M (2006) *The Handbook of Marketing Research: Uses, Misuses, and Future Advances* (Sage Publications).

Guadagni PM, Little JD (1983) A logit model of brand choice calibrated on scanner data. *Marketing Science* 2(3):203–238.

Hauser JR (2014) Consideration-set heuristics. *Journal of Business Research* 67(8):1688–1699.

Hauser JR, Wernerfelt B (1990) An evaluation cost model of consideration sets. *Journal of Consumer Research* 16(4):393–408.

Hausman J, McFadden D (1984) Specification tests for the multinomial logit model. *Econometrica: Journal of the Econometric Society* 52(5):1219–1240.

Jerath K, Ma L, Park YH, Srinivasan K (2011) A "position paradox" in sponsored search auctions. *Marketing Science* 30(4):612–627.

Jeziorski P, Segal I (2015) What makes them click: Empirical analysis of consumer demand for search advertising. *American Economic Journal: Microeconomics* 7(3):24–53.

Kempe D, Mahdian M (2008) A cascade model for externalities in sponsored search. *Proceedings of the 4th International Conference on Web and Internet Economics*, 585–596.

Li G, Rusmevichientong P, Topaloglu H (2015) The d-level nested logit model: Assortment and price optimization problems. *Operations Research* 63(2):325–342.

Luce RD (1959) *Individual Choice Behavior a Theoretical Analysis* (John Wiley & Sons).

McFadden D (1973) Conditional logit analysis of qualitative choice behavior. *Frontiers in Econometrics* 105–142.

McFadden D (1980) Econometric models for probabilistic choice among products. *Journal of Business* 53(3):S13–S29.

McFadden D, Tye WB, Train K (1977) An application of diagnostic tests for the independence from irrelevant alternatives property of the multinomial logit model, Institute of Transportation Studies, University of California.

Mehta N, Rajiv S, Srinivasan K (2003) Price uncertainty and consumer search: A structural model of consideration set formation. *Marketing Science* 22(1):58–84.

Méndez-Diaz I, Bront JJM, Vulcano G, Zabala P (2014) A branch-and-cut algorithm for the latent-class logit assortment problem. *Discrete Applied Mathematics* 164(1):246–263.

Narayanan S, Kalyanam K (2015) Position effects in search advertising and their moderators: A regression discontinuity approach. *Marketing Science* 34(3):388–407.

Negahban S, Oh S, Shah D (2012) Iterative ranking from pair-wise comparisons. *Advances in Neural Information Processing Systems*, 2474–2482.

Payne JW (1976) Task complexity and contingent processing in decision making: An information search and protocol analysis. *Organizational Behavior and Human Performance* 16(2):366–387.

Plackett RL (1975) The analysis of permutations. *Applied Statistics* 24(2):193–202.

Rusmevichientong P, Shen ZJM, Shmoys DB (2010) Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Operations Research* 58(6):1666–1680.

Rusmevichientong P, Shmoys D, Tong C, Topaloglu H (2014) Assortment optimization under the multinomial logit model with random choice parameters. *Production and Operations Management* 23(11):2023–2039.

Rusmevichientong P, Topaloglu H (2012) Robust assortment optimization in revenue management under the multinomial logit choice model. *Operations Research* 60(4):865–882.

Shocker AD, Ben-Akiva M, Boccara B, Nedungadi P (1991) Consideration set influences on consumer decision-making and choice: Issues, models, and suggestions. *Marketing Letters* 2(3):181–197.

Silk AJ, Urban GL (1978) Pre-test-market evaluation of new packaged goods: A model and measurement methodology. *Journal of Marketing Research* 15(2):171–191.

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

Train KE (2009) *Discrete choice methods with simulation* (Cambridge University Press).

Tversky A, Kahneman D (1981) The framing of decisions and the psychology of choice. *Science* 211(4481):453–458.

Tversky A, Kahneman D (1986) Rational choice and the framing of decisions. *Journal of Business* 59(4.2):S251–S278.

Ursu RM (2018) The power of rankings: Quantifying the effect of rankings on online consumer search and purchase decisions. *Marketing Science* 37(4):530–552.

Wu J, Rangaswamy A (2003) A fuzzy set model of search and consideration with an application to an online market. *Marketing Science* 22(3):411–434.

Xiong C, Wang T, Ding W, Shen Y, Liu TY (2012) Relational click prediction for sponsored search. *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, 493–502.

Xu W, Manavoglu E, Cantu-Paz E (2010) Temporal click model for sponsored search. *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 106–113.

Yang S, Ghose A (2010) Analyzing the relationship between organic and sponsored search advertising: Positive, negative, or zero interdependence? *Marketing Science* 29(4):602–623.

A Counter-Examples

A.1 The greedy algorithm

A natural greedy heuristic to potentially obtain near-optimal solutions for the display optimization problem consists in assigning items iteratively over the positions $1, \dots, n$, by selecting at each step k an unassigned item that maximizes the expected revenue $R_k(\mathcal{A})$ due to the assortment $\mathcal{A}[k]$. As the next claim shows, the performance guarantee of this algorithm could be arbitrarily bad. For ease of presentation, we make use of the simplified notation introduced in Section 2.

Lemma A.1. *There are instances of the display optimization problem for which the greedy algorithm returns an assignment $\mathcal{A}_{\rm gr}$ such that $\mathcal{R}(\mathcal{A}_{\rm gr}) = O(\frac{1}{n}) \cdot \mathcal{R}(\mathcal{A}^*)$.*

Proof. Consider an instance where item 1 has revenue $r_1 = \frac{1}{n}$ and preference weight $w_1 = n^2$, while any other item $i \in [2, n]$ has $r_i = 1$ and $w_i = \frac{1}{n}$. In addition, the position probabilities are uniform, i.e., $\lambda_1 = \cdots = \lambda_n = \frac{1}{n}$. By observing that

$$\frac{r_1 w_1}{1 + w_1} = \frac{n}{1 + n^2} > \frac{1}{1 + n} = \frac{r_2 w_2}{1 + w_2} \ ,$$

it follows that $\mathcal{A}_{\rm gr}$ assigns item 1 to the first position, and then items $2, \ldots, n$ to the remaining positions in subsequent steps. Therefore, for every position $k \in [n]$,

$$R_k\left(\mathcal{A}_{\rm gr}\right) = \sum_{i \in \mathcal{A}_{\rm gr}[k]} \frac{r_i w_i}{1 + w(\mathcal{A}_{\rm gr}[k])} = \frac{n + (k-1)/n}{1 + n^2 + (k-1)/n} \leq \frac{2}{n} ,$$

implying that $\mathcal{R}(\mathcal{A}_{\rm gr}) \leq 2/n$. On the other hand, consider the assignment $\mathcal{A}$, where item 1 is assigned to position n, while all other items are placed in $1, \ldots, n-1$. In this case, for every position $k \in [\lceil n/2 \rceil, n-1]$,

$$R_k\left(\mathcal{A}\right) = \sum_{i \in \mathcal{A}[k]} \frac{r_i w_i}{1 + w(\mathcal{A}[k])} = \frac{k/n}{1 + k/n} \geq \frac{1}{4} ,$$

meaning that $\mathcal{R}(\mathcal{A}^*) \geq \mathcal{R}(\mathcal{A}) = \Omega(1)$. ■ ■

A.2 Local search

We now consider a single-swap heuristic, that consists in exchanging the positions of two items in a given assignment. Starting from an arbitrary initial assignment, this algorithm swaps between the positions of two items as long as this operation improves the expected revenue, until hitting a prespecified halting condition. In order to attain a polynomial running time, a typical halting condition terminates when any single-swap improves the expected revenue by a factor of at most $1 + \frac{1}{n^\alpha}$, for some fixed $\alpha > 0$, in which case the current assignment is called α-locally-optimal. The next lemma asserts that assignments of this nature can be arbitrarily bad in comparison to the optimal ones.

Lemma A.2. *For any fixed $\alpha > 0$, there is an instance of the display optimization problem and an α-locally-optimal assignment $\mathcal{A}_{{\rm ss}(\alpha)}$ such that $\mathcal{R}(\mathcal{A}_{{\rm ss}(\alpha)}) = O(\frac{1}{n}) \cdot \mathcal{R}(\mathcal{A}^*)$.*

Proof. Consider an instance formed by two heavy items and $n-2$ light items. Each heavy item $i \in \{1, 2\}$ has revenue $r_i = \frac{1}{n}$ and preference weight $w_i = n^{2\alpha+1}$, while the remaining light items $i \in [3, n]$ have $r_i = 1$ and $w_i = \frac{1}{n}$. The position probabilities are uniform, i.e., $\lambda_1 = \cdots = \lambda_n = \frac{1}{n}$.

Now, let us focus on the assignment $\mathcal{A}_{{\rm ss}(\alpha)}$ in which $\mathcal{A}_{{\rm ss}(\alpha)}(k) = k$ for every position $k \in [n]$. In order to conclude the proof, we establish the next two claims.

Claim A.3. $\mathcal{R}(\mathcal{A}_{{\rm ss}(\alpha)}) = O(\frac{1}{n}) \cdot \mathcal{R}(\mathcal{A}^*)$.

Claim A.4. *The assignment $\mathcal{A}_{{\rm ss}(\alpha)}$ is α-locally-optimal.*

■ ■

Proof of Claim A.3.. Due to placing the heavy items at positions 1 and 2, for every $k \in [2, n]$,

$$R_k\left(\mathcal{A}_{{\rm ss}(\alpha)}\right) = \sum_{i \in \mathcal{A}_{{\rm ss}(\alpha)}[k]} \frac{r_i w_i}{1 + w(\mathcal{A}_{{\rm ss}(\alpha)}[k])} = \frac{2n^{2\alpha} + (k-2)/n}{1 + 2n^{2\alpha+1} + (k-2)/n} \leq \frac{3}{2n} ,$$

while $R_1(\mathcal{A}_{\text{ss}(\alpha)}) \leq \frac{1}{n}$, and therefore, $\mathcal{R}(\mathcal{A}_{\text{ss}(\alpha)}) = O(\frac{1}{n})$. However, with respect to the reverse assignment $\overleftarrow{\mathcal{A}}_{\text{ss}(\alpha)}$, where $\overleftarrow{\mathcal{A}}_{\text{ss}(\alpha)}(k) = n - k$, for every position $k \in [\lceil n/2 \rceil, n-2]$,

$$R_k\left(\overleftarrow{\mathcal{A}}_{\text{ss}(\alpha)}\right) = \sum_{i \in \overleftarrow{\mathcal{A}}_{\text{ss}(\alpha)}[k]} \frac{r_i w_i}{1 + w(\overleftarrow{\mathcal{A}}_{\text{ss}(\alpha)}[k])} = \frac{k/n}{1 + k/n} \geq \frac{1}{4} ,$$

implying that $\mathcal{R}(\mathcal{A}^*) \geq \mathcal{R}(\overleftarrow{\mathcal{A}}_{\text{ss}(\alpha)}) = \Omega(1)$. ■ ■

Proof of Claim A.4.. We argue that any improving single-swap with respect to the assignment $\mathcal{A}_{\text{ss}(\alpha)}$ increases the expected revenue by a factor of $1 + O(\frac{1}{n^{2\alpha-1}})$. For this purpose, note that an improving single-swap necessarily involves a position $k_0 \in \{1, 2\}$, occupied by a heavy item, and a position $k_1 \in [3, n]$, which holds a light item; for any other single-swap, the expected revenue clearly remains unchanged. Let $\mathcal{A}$ denote the assignment resulting from swapping positions k_0 and k_1, as described earlier. For any position $k \geq k_1$, we have $R_k(\mathcal{A}) = R_k(\mathcal{A}_{\text{ss}(\alpha)})$, since $\mathcal{A}[k] = \mathcal{A}_{\text{ss}(\alpha)}[k]$. In addition,

$$\frac{r_1 w_1}{1 + w_1} = \frac{n^{2\alpha}}{1 + n^{2\alpha+1}} > \frac{1}{1+n} = \frac{r_{k_1} w_{k_1}}{1 + w_{k_1}} ,$$

which immediately implies that $R_1(\mathcal{A}_{\text{ss}(\alpha)}) \geq R_1(\mathcal{A})$. Finally, for every remaining position $k \in [2, k_1 - 1]$,

$$R_k(\mathcal{A}) - R_k(\mathcal{A}_{\text{ss}(\alpha)}) = \frac{n^{2\alpha} + (k-1)/n}{1 + n^{2\alpha+1} + (k-1)/n} - \frac{2n^{2\alpha} + (k-2)/n}{1 + 2n^{2\alpha+1} + (k-2)/n} \leq \frac{3}{n^{2\alpha+1}} .$$

By combining the above observations, we have

$$\mathcal{R}(\mathcal{A}) - \mathcal{R}(\mathcal{A}_{\text{ss}(\alpha)}) \leq \frac{k_1 - 2}{n} \cdot \frac{3}{n^{2\alpha+1}} \leq \frac{3}{n^{2\alpha+1}} ,$$

meaning that the ratio between the expected revenues of $\mathcal{A}$ and $\mathcal{A}_{\text{ss}(\alpha)}$ is bounded by

$$\frac{\mathcal{R}(\mathcal{A})}{\mathcal{R}(\mathcal{A}_{\text{ss}(\alpha)})} \leq 1 + \frac{3/n^{2\alpha+1}}{R_1(\mathcal{A}_{\text{ss}(\alpha)})/n} = 1 + O\left(\frac{1}{n^{2\alpha-1}}\right) . ■$$

■

B Instance Transformations

In what follows, we show that any instance of the display optimization problem can be modified to satisfy Assumptions 2.1, 2.2, and 2.3, with negligible effects on optimality. Our approach proceeds in three steps: We will first modify the preference weights, then the item prices, and finally the distribution over consideration sets.

B.1 Structural alterations

Step 1: Modified weights. As a preliminary step, we transform the item weights to obtain more structured instances. Specifically, for items with $w_i \geq \frac{\delta}{n}$, we round up w_i to the nearest power of $1+\delta$ multiplied by $\frac{\delta}{n}$, i.e., w_i is rounded to $\frac{\delta}{n} \cdot (1+\delta)^\tau$, where τ is the minimal integer satisfying $\frac{\delta}{n} \cdot (1+\delta)^\tau \geq w_i$. For items with $w_i < \frac{\delta}{n}$, we simply round up w_i to $\frac{\delta}{n}$. In both cases, the rounded weight of item i is denoted by $\tilde{w}_i$.

Step 2: Modified ρ-quantities and revenues. Furthermore, we would like the resulting ρ-quantities to approximate the initial quantities within a factor of $1+\delta$, i.e., to make sure that $\tilde{\rho}_i$ satisfies $\tilde{\rho}_i \leq \rho_i \leq \tilde{\rho}_i \cdot (1+\delta)$, where $\tilde{\rho}_i = \tilde{r}_i \tilde{w}_i$ for a modified revenue quantity $\tilde{r}_i$. To this end, we first determine $\tilde{\rho}_i$ by rounding down ρ_i to the nearest power of $1+\delta$ multiplied by $\rho_{\min}$, i.e., ρ_i is rounded to $\rho_{\min} \cdot (1+\delta)^\tau$, where τ is the maximum integer satisfying $\rho_i \geq \rho_{\min} \cdot (1+\delta)^\tau$. Next, in order to be compatible with the new weights and ρ-quantities, we redefine the item revenues accordingly, by picking $\tilde{r}_i = \tilde{\rho}_i / \tilde{w}_i$. Consequently, for every assortment $S \subseteq [n]$, the expected revenue $\tilde{R}(S)$ is defined with respect to the modified weights $\tilde{w}_1, \ldots, \tilde{w}_n$ and prices $\tilde{r}_1, \ldots, \tilde{r}_n$, that is, $\tilde{R}(S) = \sum_{i \in S} \frac{\tilde{r}_i \tilde{w}_i}{1+\tilde{w}(S)}$.

Step 3: Modified distribution. Finally, we modify the probability distribution over positions by defining $\tilde{\lambda}_k = \lambda_k + \frac{\delta}{n} \cdot \frac{r_{\min} \cdot w_{\min}}{(\sum_{i=1}^n r_i) \cdot (1+w([n]))}$ for every position $k \in [n]$, where $r_{\min} = \min_{i \in [n]} r_i$ and $w_{\min} = \min_{i \in [n]} w_i$. Observe that, following this transformation, $(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_n)$ is not longer a probability distribution, since $\sum_{k \in [n]} \tilde{\lambda}_k > 1$. These parameters come into play as positive weights in the modified objective $\tilde{\mathcal{R}}(\mathcal{A}) = \sum_{k=1}^n \tilde{\lambda}_k \cdot \tilde{R}(\mathcal{A}[1,k])$.

B.2 Analysis

All together, we have just defined a modified instance that satisfies Assumptions 2.1-2.3, where the new objective is that of maximizing $\tilde{\mathcal{R}}(\mathcal{A})$. The next claim shows that this transformation incurs a negligible loss in optimality.

Lemma B.1. *Let $\tilde{\mathcal{A}}$ be an α-approximate assignment for the modified instance. Then, $\mathcal{R}(\tilde{\mathcal{A}}) \geq (1-3\delta) \cdot \alpha \cdot \mathcal{R}(\mathcal{A}^*)$.*

Proof. To prove this claim, it suffices to show that, for any assignment $\mathcal{A}$,

$$(1-2\delta) \cdot \mathcal{R}(\mathcal{A}) \leq \tilde{\mathcal{R}}(\mathcal{A}) \leq (1+\delta) \cdot \mathcal{R}(\mathcal{A}) \ . \tag{23}$$

To establish the first inequality, observe that:

$$\begin{aligned} \tilde{\mathcal{R}}(\mathcal{A}) &= \sum_{k=1}^n \tilde{\lambda}_k \cdot \tilde{R}\left(\mathcal{A}[1,k]\right) \\ &\geq \sum_{k=1}^n \lambda_k \cdot \tilde{R}\left(\mathcal{A}[1,k]\right) \\ &= \sum_{k=1}^n \lambda_k \cdot \frac{\tilde{\rho}(\mathcal{A}[1,k])}{1+\tilde{w}(\mathcal{A}[1,k])} \\ &\geq \frac{1-\delta}{1+\delta} \cdot \sum_{k=1}^n \lambda_k \cdot \frac{\rho(\mathcal{A}[1,k])}{1+w(\mathcal{A}[1,k])} \\ &\geq (1-\delta)^2 \cdot \mathcal{R}(\mathcal{A}) \ , \end{aligned}$$

where the first inequality holds since $\tilde{\lambda}_k \geq \lambda_k$ and the second inequality is due to having

$$\tilde{w}(\mathcal{A}[1,k]) = \sum_{i \in \mathcal{A}[1,k] \cap I} \tilde{w}_i + \sum_{i \in \mathcal{A}[1,k] \cap \bar{I}} \tilde{w}_i \leq |\mathcal{A}[1,k] \cap I| \cdot \frac{\delta}{n} + (1+\delta) \cdot \sum_{i \in \mathcal{A}[1,k] \cap \bar{I}} w_i \leq \delta + (1+\delta) \cdot w\left(\mathcal{A}[1,k]\right) ,$$

where $I = \{i \in [n] : w_i \leq \frac{\delta}{n}\}$, as well as

$$\tilde{\rho}(\mathcal{A}[1,k]) = \sum_{i \in \mathcal{A}[1,k] \cap I} \tilde{\rho}_i + \sum_{i \in \mathcal{A}[1,k] \cap \bar{I}} \tilde{\rho}_i \geq (1-\delta) \cdot \sum_{i \in \mathcal{A}[1,k] \cap \bar{I}} \rho_i \geq (1-\delta) \cdot \rho\left(\mathcal{A}[1,k]\right) ,$$

where the second inequality proceeds from our definition of $\tilde{\rho}_i$ in Step 2 of Section B.1. To establish the second inequality in (23), observe that:

$$\begin{aligned}
\tilde{\mathcal{R}}(\mathcal{A}) &= \sum_{k=1}^{n} \tilde{\lambda}_k \cdot \tilde{R}\left(\mathcal{A}[1,k]\right) \\
&= \sum_{k=1}^{n} \lambda_k \cdot \tilde{R}\left(\mathcal{A}[1,k]\right) + \frac{r_{\min} w_{\min}}{1 + w([n])} \cdot \sum_{k=1}^{n} \frac{\delta}{n} \cdot \frac{\tilde{R}\left(\mathcal{A}[1,k]\right)}{\sum_{i=1}^{n} r_i} \\
&\leq \sum_{k=1}^{n} \lambda_k \cdot R\left(\mathcal{A}[1,k]\right) + \delta \cdot \frac{r_{\min} w_{\min}}{1 + w([n])} \\
&\leq (1+\delta) \cdot \mathcal{R}(\mathcal{A}) ,
\end{aligned}$$

where the first inequality holds since $\tilde{R}(\mathcal{A}[1,k]) \leq R(\mathcal{A}[1,k])$, as $\tilde{w}_i \geq w_i$ and $\rho_i \geq \tilde{\rho}_i$ for every item $i \in [n]$, and the second inequality holds since $\sum_{i=1}^{n} r_i$ is an upper bound on the expected revenue of any assortment, in particular, $\sum_{i=1}^{n} r_i \geq R(\mathcal{A}[1,k])$ for every $k \in [n]$. The third inequality proceeds is obtained by noting that

$$\mathcal{R}(\mathcal{A}) = \sum_{k=1}^{n} \lambda_k \cdot R(\mathcal{A}[1,k]) \geq \min_k R(\mathcal{A}[1,k]) \geq \frac{r_{\min} w_{\min}}{1 + w([n])} .$$

■ ■

C Additional Proofs

C.1 Proof of Lemma 2.5

In what follows, we explicitly construct the approximate assortments $\mathcal{U}^-$ and $\mathcal{U}^+$ through a combination of rounding procedures. Next, we argue that there exist $O(|\mathcal{I}|^{O(\frac{1}{\delta^2} \cdot \log \frac{1}{\delta})})$ distinct approximate assortments that are generated through such procedures.

Additional notation. For an integer $\ell \geq 0$, we designate by $\mathcal{W}^\ell$ the set of items with a preference weight of $\frac{\delta}{n} \cdot (1+\delta)^\ell$; the collection $\mathcal{W}^\ell$ will be referred to as *weight class* ℓ. By Assumption 2.1, the sequence of weight classes $\mathcal{W}^0, \mathcal{W}^1, \ldots$ forms a partition of the collection of items $[n]$. We denote by $\mathcal{W}^{\ell,m}$ the union of all items over the weight classes $\ell, \ell+1, \ldots, m$. Additionally, for every integer $\ell \geq 0$ and $k \in [0, |\mathcal{W}^\ell|]$, we let $\mathcal{W}^\ell[k]$ be the subset of $\mathcal{W}^\ell$ formed by the k items having the largest ρ-quantities. Similarly, for every integer $\ell \geq 0$, we define the *ρ-class* $\mathcal{Q}^\ell$ as the set of items with a ρ-quantity of $\rho_{\min} \cdot (1+\delta)^\ell$. We denote by $\mathcal{Q}^{\ell,m}$ the union of all items over ρ-classes $\ell, \ell+1, \ldots, m$. Finally, for every pair of integers $\ell \geq 0$ and $k \in [0, |\mathcal{Q}^\ell|]$, we let $\mathcal{Q}^\ell[k]$ be the subset of $\mathcal{Q}^\ell$ formed by the k items having the smallest weights.

With these new notation at hand, we revisit the priority rules defined in Section 2.1. Specifically, an assortment $\mathcal{U} \subseteq [n]$ satisfies the priority rules 1 and 2 if and only if, for every $\ell \geq 0$, we have $\mathcal{U} \cap \mathcal{W}^\ell = \mathcal{W}^\ell[n_\ell]$ and $\mathcal{U} \cap \mathcal{Q}^\ell = \mathcal{Q}^\ell[m_\ell]$, where $n_\ell = |\mathcal{U} \cap \mathcal{W}^\ell|$ and $m_\ell = |\mathcal{U} \cap \mathcal{Q}^\ell|$. In other words, within each weight class, the assortment $\mathcal{U}$ picks the top items by decreasing ρ-quantities. Within each ρ-class, the assortment $\mathcal{U}$ picks the top items by increasing weights.

Outline. At a high-level, our approach for approximating $\mathcal{U}$ is to construct assortments $\mathcal{U}^+$ and $\mathcal{U}^-$ that select nearly the same number of top items in each weight class and ρ-class, namely, $|\mathcal{U}^+ \cap \mathcal{W}^\ell|$ and $|\mathcal{U}^- \cap \mathcal{W}^\ell|$ are nearly equal to n_ℓ, and similarly, $|\mathcal{U}^+ \cap \mathcal{Q}^\ell|$ and $|\mathcal{U}^- \cap \mathcal{Q}^\ell|$ are nearly equal to m_ℓ. However, imposing these requirements for all $\ell \geq 0$ is impractical for generating a polynomially-sized collection of approximate assortments. Hence, a crucial ingredient of our procedure is to restrict the range of classes considered. Specifically, we will show below that it is sufficient to consider only $K = \lceil \log_{1+\delta}(\frac{n}{\delta}) \rceil$ distinct weight classes $\mathcal{W}^\ell$ and ρ-classes $\mathcal{Q}^\ell$, allowing us to generate only polynomially-many approximate assortments.

In what follows, from a terminology perspective, a "guess" refers to a certain value, related to the assortment $\mathcal{U}$, which is assumed to be available in order to execute our approximation procedure. Computationally-speaking, these values are obtained through an exhaustive enumeration, where our approximation procedure is executed with each possible guess. Ultimately, we pick the assortments $\mathcal{U}^+$ and $\mathcal{U}^-$ that satisfy all desired properties. By bounding the number of possible distinct guesses, we will argue that this guessing procedure indeed runs in polynomial time.

C.1.1 Construction of $\mathcal{U}^+$.

In order to construct $\mathcal{U}^+$, we separately define two supersets of $\mathcal{U}$, respectively denoted by $S_1(\delta, \mathcal{U})$ and $S_2(\delta, \mathcal{U})$. The first superset $S_1(\delta, \mathcal{U})$ is defined in order to approximate $\mathcal{U}$ in ρ-quantity terms; namely, it will satisfy $\rho(S_1(\delta, \mathcal{U})) \leq (1+\delta) \cdot \rho(\mathcal{U})$. The latter superset $S_2(\delta, \mathcal{U})$ is defined in order to approximate $\mathcal{U}$ in terms of weight, meaning that $w(S_2(\delta, \mathcal{U})) \leq (1+\delta) \cdot w(\mathcal{U})$. Ultimately, the super-assortment $\mathcal{U}^+$ is defined as $\mathcal{U}^+ = S_1(\delta, \mathcal{U}) \cap S_2(\delta, \mathcal{U})$, which clearly guarantees that Property 3 is met. In what follows, it will be convenient for the reader to refer to Figure 3, where the construction of $S_1(\delta, \mathcal{U})$ is illustrated though a simple example. It is worth noting that $S_1(\delta, \mathcal{U})$ and $S_2(\delta, \mathcal{U})$ are constructed using symmetrical procedures, where the notions of ρ-quantity and weight are fully interchangeable.

Construction of $S_1(\delta, \mathcal{U})$. Let $\bar{\eta}(\mathcal{U})$ be the largest index of a ρ-class that contains at least one item of $\mathcal{U}$, and let $\eta(\mathcal{U}) = \max\{0, \bar{\eta}(\mathcal{U}) - K + 1\}$. We first guess an under-estimate $\tilde{\rho}$ for $\rho(\mathcal{U})$ that satisfies $\frac{\rho(\mathcal{U})}{2} \leq \tilde{\rho} \leq \rho(\mathcal{U})$; this estimate is of the form $\tilde{\rho} = \rho_{\min} \cdot 2^\tau$ where $\tau \in \mathbb{N}$. Next, we let $\tilde{\theta}$ be the result of rounding the quantity $\delta \cdot \frac{\tilde{\rho}}{K}$ down to the nearest power of 2, namely $\tilde{\theta} = 2^{\lceil \log \delta \cdot \frac{\tilde{\rho}}{K} \rceil}$. In addition, we guess for every ρ-class $\ell \in [\eta(\mathcal{U}), \bar{\eta}(\mathcal{U})]$ an over-estimate ρ_ℓ^+ of $\rho(\mathcal{Q}^\ell \cap \mathcal{U})$, picked as the unique integer multiple of $\tilde{\theta}$ that satisfies $\rho(\mathcal{Q}^\ell \cap \mathcal{U}) \leq \rho_\ell^+ \leq \rho(\mathcal{Q}^\ell \cap \mathcal{U}) + \tilde{\theta}$. Next, for each ρ-class $\ell \in [\eta(\mathcal{U}), \bar{\eta}(\mathcal{U})]$, let $\hat{m}_\ell^+(\mathcal{U})$ be the maximal number of items in class $\mathcal{Q}^\ell$,

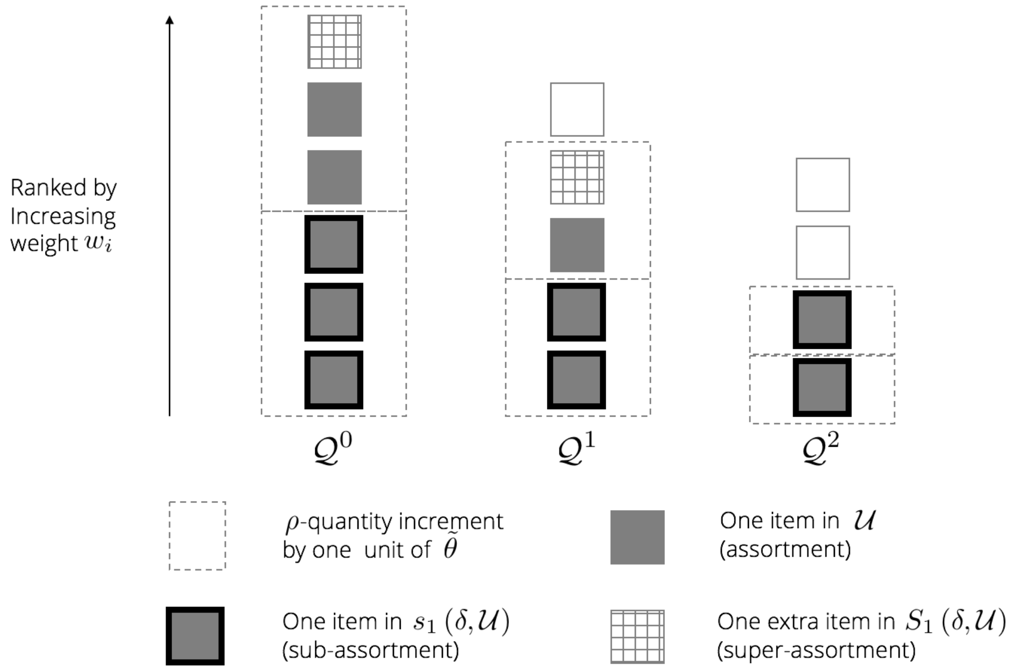


Figure 3: Illustration of the notion of approximate assortments in an example, with three ρ-classes. In the ρ-class $\mathcal{Q}^0$, $m_0 = 5$ items are picked by $\mathcal{U}$, while $\hat{m}_0^+(\mathcal{U}) = 6$ items are picked by the approximate assortment $S_1(\delta, \mathcal{U})$. In the ρ-class $\mathcal{Q}^2$, the same number of items $m_2(\mathcal{U}) = 2 = \hat{m}_2^+(\mathcal{U})$ are picked by $\mathcal{U}$ and $S_1(\delta, \mathcal{U})$.

chosen by increasing weights, whose total ρ-quantity satisfies $\rho(\mathcal{Q}^\ell[\hat{m}_\ell^+(\mathcal{U})]) \leq \rho_\ell^+$. By selecting all items of $\mathcal{Q}^{0,\eta(\mathcal{U})-1}$, as well as the top $\hat{m}_\ell^+(\mathcal{U})$ items in $\mathcal{Q}^\ell$ by increasing weight quantities, over all ρ-classes $\ell \in [\eta(\mathcal{U}), \bar{\eta}(\mathcal{U})]$, we obtain the assortment

$$S_1(\delta, \mathcal{U}) = \mathcal{Q}^{0,\eta(\mathcal{U})-1} \cup \left(\bigcup_{\ell=\eta(\mathcal{U})}^{\bar{\eta}(\mathcal{U})} \mathcal{Q}^\ell \left[\hat{m}_\ell^+(\mathcal{U}) \right] \right) , \tag{24}$$

where by convention $\mathcal{Q}^{0,\eta(\mathcal{U})-1} = \emptyset$ if $\eta(\mathcal{U}) \leq 0$. Since ρ_ℓ^+ is an over-estimate of $\rho(\mathcal{Q}^\ell \cap \mathcal{U})$, it follows that $(\mathcal{U} \cap \mathcal{Q}^\ell) \subseteq (S_1(\delta, \mathcal{U}) \cap \mathcal{Q}^\ell)$ by definition of $\hat{m}_\ell^+(\mathcal{U})$. In addition, we clearly have $(\mathcal{U} \cap \mathcal{Q}^{0,\eta(\mathcal{U})-1}) \subseteq (S_1(\delta, \mathcal{U}) \cap \mathcal{Q}^{0,\eta(\mathcal{U})-1})$. Consequently, we have just shown that $S_1(\delta, \mathcal{U})$ is a superset of $\mathcal{U}$. Furthermore, we can upper-bound the total ρ-quantity of $S_1(\delta, \mathcal{U})$ as follows:

$$\begin{aligned}
\rho\left(S_1\left(\delta, \mathcal{U}\right)\right) &= \rho\left(\mathcal{Q}^{0,\eta(\mathcal{U})-1}\right) + \sum_{\ell=\eta(\mathcal{U})}^{\bar{\eta}(\mathcal{U})} \rho\left(\mathcal{Q}^\ell\left[\hat{m}_\ell^+(\mathcal{U})\right]\right) \\
&\leq \delta \cdot \rho\left(\mathcal{U}\right) + \sum_{\ell=\eta(\mathcal{U})}^{\bar{\eta}(\mathcal{U})} \rho\left(\mathcal{Q}^\ell\left[\hat{m}_\ell^+(\mathcal{U})\right]\right) \qquad (25) \\
&\leq \delta \cdot \rho\left(\mathcal{U}\right) + \sum_{\ell=\eta(\mathcal{U})}^{\bar{\eta}(\mathcal{U})} \rho_\ell^+ \qquad (26) \\
&\leq \delta \cdot \rho\left(\mathcal{U}\right) + \sum_{\ell=\eta(\mathcal{U})}^{\bar{\eta}(\mathcal{U})} \left(\rho\left(\mathcal{Q}^\ell \cap \mathcal{U}\right) + \delta \cdot \frac{\tilde{\rho}}{K}\right) \qquad (27)
\end{aligned}$$

$$\leq \quad (1+\delta) \cdot \rho(\mathcal{U}) + \delta \cdot \tilde{\rho}$$
$$\leq \quad (1+2\delta) \cdot \rho(\mathcal{U}) \ , \tag{28}$$

where inequality (25) is explained below. Inequality (26) follows from our definition of $\hat{m}_\ell^+(\mathcal{U})$, whereby $\rho(\mathcal{Q}^\ell[\hat{m}_\ell^+(\mathcal{U})]) \leq \rho_\ell^+$. Inequality (27) holds since $\rho_\ell^+ \leq \rho(\mathcal{Q}^\ell \cap \mathcal{U}) + \tilde{\theta}$, by construction. The last inequality (28) holds since $\tilde{\rho}$ is an under-estimate for $\rho(\mathcal{U})$, meaning that $\tilde{\rho} \leq \rho(\mathcal{U})$.

In order to establish inequality (25), observe that, when $\eta(\mathcal{U}) \geq 1$:

$$\rho\left(\mathcal{Q}^{0,\eta(\mathcal{U})-1}\right) \leq \left|\mathcal{Q}^{0,\eta(\mathcal{U})-1}\right| \cdot \rho_{\min} \cdot (1+\delta)^{i(\mathcal{U})-1} \leq n \cdot \rho\left(\mathcal{Q}^{\bar{\eta}(\mathcal{U})} \cap \mathcal{U}\right) \cdot (1+\delta)^{-K} \leq \delta \cdot \rho(\mathcal{U}) \ , \tag{29}$$

where the second inequality holds since $\bar{\eta}(\mathcal{U}) = \eta(\mathcal{U}) + K - 1$ when $\eta(\mathcal{U}) \geq 1$, and the last inequality proceeds by noting that $(1+\delta)^{-K} \leq \frac{\delta}{n}$ since $K = \lceil \log_{1+\delta}(\frac{n}{\delta}) \rceil$.

Construction of $S_2(\delta,\mathcal{U})$. Now, let $\bar{\zeta}(\mathcal{U})$ be the largest index of a weight class that contains at least one item of $\mathcal{U}$, and let $\zeta(\mathcal{U}) = \max\{0, \bar{\zeta}(\mathcal{U}) - K + 1\}$. We begin by guessing an under-estimate $\tilde{w}$ for $w(\mathcal{U})$ that satisfies $\frac{w(\mathcal{U})}{2} \leq \tilde{w} \leq w(\mathcal{U})$; this estimate is of the form $\tilde{w} = \frac{\delta}{n} \cdot 2^\tau$ where $\tau \in \mathbb{N}$. Next, we let $\tilde{\omega}$ be the result of rounding the quantity $\delta \cdot \frac{\tilde{w}}{K}$ down to the nearest power of 2, namely $\tilde{\omega} = 2^{\lceil \log \delta \cdot \frac{\tilde{w}}{K} \rceil}$. In addition, we guess for every weight class $\ell \in [\zeta(\mathcal{U}), \bar{\zeta}(\mathcal{U})]$ an over-estimate w_ℓ^+ of $w(\mathcal{W}^\ell \cap \mathcal{U})$, picked as the unique integer multiple of $\tilde{\omega}$ that satisfies $w(\mathcal{W}^\ell \cap \mathcal{U}) \leq w_\ell^+ \leq w(\mathcal{W}^\ell \cap \mathcal{U}) + \tilde{\omega}$. Next, for each weight class $\ell \in [\zeta(\mathcal{U}), \bar{\zeta}(\mathcal{U})]$, let $\hat{n}_\ell^+(\mathcal{U})$ be the maximal number of items in the class $\mathcal{W}^\ell$, chosen by increasing weights, whose total weight quantity satisfies $w(\mathcal{W}^\ell[\hat{n}_\ell^+(\mathcal{U})]) \leq w_\ell^+$. By selecting the items of $\mathcal{W}^{0,\zeta(\mathcal{U})-1}$ as well as the top $\hat{n}_\ell^+(\mathcal{U})$ items in $\mathcal{W}^\ell$ by decreasing ρ-quantities, over all weight classes $\ell \in [\zeta(\mathcal{U}), \bar{\zeta}(\mathcal{U})]$, we obtain the assortment

$$S_2(\delta,\mathcal{U}) = \mathcal{W}^{0,\zeta(\mathcal{U})-1} \cup \left(\bigcup_{\ell=\zeta(\mathcal{U})}^{\bar{\zeta}(\mathcal{U})} \mathcal{W}^\ell \left[\hat{n}_\ell^+(\mathcal{U})\right] \right) . \tag{30}$$

By definition of w_ℓ^+ and $\hat{n}_\ell^+$, note that $(\mathcal{U} \cap \mathcal{W}^\ell) \subseteq (S_2(\delta,\mathcal{U}) \cap \mathcal{W}^\ell)$ for every $\ell \in [\zeta(\mathcal{U}), \bar{\zeta}(\mathcal{U})]$, and consequently, we have that $\mathcal{U} \subseteq S_2(\delta,\mathcal{U})$. Using a sequence of inequalities nearly identical to (28), it is easy to verify that $w(S_2(\delta,\mathcal{U})) \leq (1+2\delta) \cdot w(\mathcal{U})$. Hence, we may conclude that the assortment $\mathcal{U}^+ = S_1(\delta,\mathcal{U}) \cup S_2(\delta,\mathcal{U})$ satisfies all the desired inequalities of Properties 1 and 3, with an accuracy level of 2δ.

Bounding the number of guesses. Here, we argue that $\mathcal{U}^+ = S_1(\delta,\mathcal{U}) \cup S_2(\delta,\mathcal{U})$ can be constructed in polynomial time, by bounding the overall number of distinct guesses. As a by-product, we bound the number of distinct super-assortments that can be generated through our rounding procedures. For this purpose, the key observation is that the assortments $S_1(\delta,\mathcal{U})$ and $S_2(\delta,\mathcal{U})$ are uniquely defined by the following parameters:

- The integers $\bar{\eta}(\mathcal{U})$, $\bar{\zeta}(\mathcal{U})$, τ_1 and τ_2 where $\tilde{\rho} = \rho_{\min} \cdot 2^{\tau_1}$ and $\tilde{w} = \frac{\delta}{n} \cdot 2^{\tau_2}$.
- The integral vector $(k_t^1)_{t\in[K]}$, where $\rho_{\eta(\mathcal{U})+t}^+ = k_t^1 \cdot \tilde{\theta}$ for every $t \in [K]$.
- The integral vector $(k_t^2)_{t\in[K]}$, where $w_{\zeta(\mathcal{U})+t}^+ = k_t^2 \cdot \tilde{\omega}$ for every $t \in [K]$.

It immediately follows that $\tau_1 = O(|\mathcal{I}|)$ and $\tau_2 = O(\log \frac{1}{\delta}) + O(|\mathcal{I}|)$. In addition, it is easy to verify that $\bar{\eta}(\mathcal{U}) = O(\frac{1}{\delta} \cdot |\mathcal{I}|)$, since $\bar{\eta}(\mathcal{U})$ is the index of a non-empty ρ-class, and similarly, $\bar{\zeta}(\mathcal{U}) = O(\frac{1}{\delta} \cdot |\mathcal{I}|)$. Lastly, in order to bound the vector $(k_t^1)_{t\in[K]}$, observe that

$$\sum_{t=1}^{K} k_t^1 = \sum_{\ell=\eta(\mathcal{U})}^{\bar{\eta}(\mathcal{U})} \left\lceil \frac{\rho(\mathcal{Q}^\ell \cap \mathcal{U})}{\tilde{\theta}} \right\rceil \leq \sum_{\ell=\eta(\mathcal{U})}^{\bar{\eta}(\mathcal{U})} \left(2K \cdot \frac{\rho(\mathcal{Q}^\ell \cap \mathcal{U})}{\delta\tilde{\rho}} + 1 \right) \leq K \cdot \left(2 \cdot \frac{\rho(\mathcal{U})}{\delta\tilde{\rho}} + 1 \right) \leq \frac{5K}{\delta} ,$$

Using nearly identical inequalities, we may verify that $\sum_{t=1}^{K} k_t^2 \leq \frac{5K}{\delta}$. Therefore, given that $K = \lceil \log_{1+\delta}(\frac{n}{\delta}) \rceil$, standard counting arguments imply that the number of possible vectors $(k_t^1)_{t\in[K]}$ and $(k_t^2)_{t\in[K]}$ is $O(2^{O(\frac{K}{\delta})}) = O(n^{O(\frac{1}{\delta^2} \cdot \log \frac{1}{\delta})})$. It follows that $|\mathcal{S}^+(\delta)| = O(|\mathcal{I}|^{O(\frac{1}{\delta^2} \cdot \log \frac{1}{\delta})})$.

C.1.2 Construction of $\mathcal{U}^-$.

In order to construct $\mathcal{U}^-$, we separately define two subsets of $\mathcal{U}$, respectively denoted by $s_1(\delta,\mathcal{U})$ and $s_2(\delta,\mathcal{U})$. The first subset $s_1(\delta,\mathcal{U})$ is defined in order to approximate $\mathcal{U}$ in ρ-quantity terms; namely, it will satisfy $\rho(s_1(\delta,\mathcal{U})) \geq (1-\delta) \cdot \rho(\mathcal{U})$. The latter subset $s_2(\delta,\mathcal{U})$ is defined in order to approximate $\mathcal{U}$ in terms of weight, meaning that $w(s_2(\delta,\mathcal{U})) \geq (1-\delta) \cdot w(\mathcal{U})$. Ultimately, the sub-assortment $\mathcal{U}^-$ is defined as $\mathcal{U}^- = s_1(\delta,\mathcal{U}) \cup s_2(\delta,\mathcal{U})$, which clearly guarantees that Property 2 is met. We refer the reader to Figure 3, where the construction of $s_1(\delta,\mathcal{U})$ is illustrated through a simple example.

Construction of $s_1(\delta,\mathcal{U})$. Having defined the quantities $\eta(\mathcal{U})$, $\bar{\eta}(\mathcal{U})$ and $\tilde{\theta}$ in our construction of $S_1(\delta,\mathcal{U})$, we guess an under-estimate ρ_ℓ^- of $\rho(\mathcal{Q}^\ell \cap \mathcal{U})$ for every ρ-class $\ell \in [\eta(\mathcal{U}), \bar{\eta}(\mathcal{U})]$. Specifically, ρ_ℓ^- is defined as the unique integer multiple of $\tilde{\theta}$ that satisfies $\rho(\mathcal{Q}^\ell \cap \mathcal{U}) - \tilde{\theta} \leq \rho_\ell^- \leq \rho(\mathcal{Q}^\ell \cap \mathcal{U})$. Next, for each ρ-class $\ell \in [\eta(\mathcal{U}), \bar{\eta}(\mathcal{U})]$, let $\hat{m}_\ell^-(\mathcal{U})$ be the minimal number of items in class $\mathcal{Q}^\ell$, chosen by increasing weights, whose total ρ-quantity satisfies $\rho(\mathcal{Q}^\ell[\hat{m}_\ell^-(\mathcal{U})]) \geq \rho_\ell^-$. By selecting the top $\hat{m}_\ell^-(\mathcal{U})$ items in $\mathcal{Q}^\ell$ by increasing weight quantities, over all ρ-classes $\ell \in [\eta(\mathcal{U}), \bar{\eta}(\mathcal{U})]$, we obtain the assortment

$$s_1(\delta,\mathcal{U}) = \bigcup_{\ell=\eta(\mathcal{U})}^{\bar{\eta}(\mathcal{U})} \mathcal{Q}^\ell \left[\hat{m}_\ell^-(\mathcal{U}) \right] \ . \tag{31}$$

Since ρ_ℓ^- is an under-estimate of $\rho(\mathcal{Q}^\ell \cap \mathcal{U})$, it follows that $(s_1(\delta,\mathcal{U}) \cap \mathcal{Q}^\ell) \subseteq (\mathcal{U} \cap \mathcal{Q}^\ell)$ by definition of $\hat{m}_\ell^-(\mathcal{U})$. Consequently, we have just shown that $s_1(\delta,\mathcal{U})$ is a subset of $\mathcal{U}$. Furthermore, we can lower-bound the total ρ-quantity of $s_1(\delta,\mathcal{U})$ as follows:

$$\begin{aligned}
\rho\left(s_1\left(\delta,\mathcal{U}\right)\right) &= \sum_{\ell=\eta(\mathcal{U})}^{\bar{\eta}(\mathcal{U})} \rho\left(\mathcal{Q}^\ell \left[\hat{m}_\ell^-(\mathcal{U}) \right] \right) \\
&\geq \sum_{\ell=\eta(\mathcal{U})}^{\bar{\eta}(\mathcal{U})} \rho_\ell^- \qquad (32) \\
&\geq \sum_{\ell=\eta(\mathcal{U})}^{\bar{\eta}(\mathcal{U})} \left(\rho\left(\mathcal{Q}^\ell \cap \mathcal{U} \right) - \delta \cdot \frac{\tilde{\rho}}{K} \right) \qquad (33)
\end{aligned}$$

$$\begin{aligned} &\geq \rho(\mathcal{U}) - \rho\left(\mathcal{Q}^{0,\eta(\mathcal{U})-1}\right) - \delta \cdot \tilde{\rho} \\ &\geq (1-2\delta) \cdot \rho(\mathcal{U}) \ , \end{aligned} \tag{34}$$

where inequality (32) follows from our definition of $\hat{m}_\ell^-(\mathcal{U})$, whereby $\rho(\mathcal{Q}^\ell[\hat{m}_\ell^-(\mathcal{U})]) \geq \rho_\ell^-$. Inequality (33) holds since $\rho_\ell^- \geq \rho(\mathcal{Q}^\ell \cap \mathcal{U}) - \tilde{\theta}$, by construction. The last inequality (34) holds since $\rho(\mathcal{Q}^{0,\eta(\mathcal{U})-1}) \leq \delta \cdot \rho(\mathcal{U})$, as shown by inequality (29).

Construction of $s_2(\delta,\mathcal{U})$. Having defined the quantities $\zeta(\mathcal{U})$, $\bar{\zeta}(\mathcal{U})$ and $\tilde{\omega}$ in our construction of $S_2(\delta,\mathcal{U})$, we guess an under-estimate w_ℓ^- of $w(\mathcal{W}^\ell \cap \mathcal{U})$ for every weight class $\ell \in [\zeta(\mathcal{U}), \bar{\zeta}(\mathcal{U})]$. Specifically, w_ℓ^- is defined as the unique integer multiple of $\tilde{\omega}$ that satisfies $w(\mathcal{W}^\ell \cap \mathcal{U}) - \tilde{\omega} \leq w_\ell^- \leq w(\mathcal{W}^\ell \cap \mathcal{U})$. Next, for each weight class $\ell \in [\zeta(\mathcal{U}), \bar{\zeta}(\mathcal{U})]$, let $\hat{n}_\ell^-(\mathcal{U})$ be the minimal number of items in the class $\mathcal{W}^\ell$, chosen by increasing weights, whose total weight quantity satisfies $w(\mathcal{W}^\ell[\hat{n}_\ell^-(\mathcal{U})]) \geq w_\ell^-$. By selecting the top $\hat{n}_\ell^-(\mathcal{U})$ items in $\mathcal{W}^\ell$ by decreasing ρ-quantities, over all weight classes $\ell \in [\zeta(\mathcal{U}), \bar{\zeta}(\mathcal{U})]$, we obtain the assortment

$$s_2(\delta,\mathcal{U}) = \bigcup_{\ell=\zeta(\mathcal{U})}^{\bar{\zeta}(\mathcal{U})} \mathcal{W}^\ell \left[\hat{n}_\ell^-(\mathcal{U})\right] \ . \tag{35}$$

By definition of w_ℓ^- and $\hat{n}_\ell^-$, note that $(s_2(\delta,\mathcal{U}) \cap \mathcal{W}^\ell) \subseteq (\mathcal{U} \cap \mathcal{W}^\ell)$ for every $\ell \in [\zeta(\mathcal{U}), \bar{\zeta}(\mathcal{U})]$, and consequently, we have that $s_2(\delta,\mathcal{U}) \subseteq \mathcal{U}$. Using a sequence of inequalities nearly identical to (34), it is easy to verify that $w(s_2(\delta,\mathcal{U})) \geq (1-2\delta) \cdot w(\mathcal{U})$. Hence, we may conclude that the assortment $\mathcal{U}^- = s_1(\delta,\mathcal{U}) \cup s_2(\delta,\mathcal{U})$ satisfies all the desired inequalities of Properties 1 and 2, with an accuracy level of 2δ.

Using counting arguments similar to Section C.1.1, it is not difficult to show that the assortment $\mathcal{U}^- = s_1(\delta,\mathcal{U}) \cup s_2(\delta,\mathcal{U})$ can be constructed in polynomial time, by bounding the number of distinct guesses. The proof is omitted for concision.

C.2 Proof of Lemma 2.6

Recall that $\mathcal{U}_1^+ = S_1(\mathcal{U}_1, \delta) \cap S_2(\mathcal{U}_1, \delta)$ and $\mathcal{U}_2^+ = S_1(\mathcal{U}_2, \delta) \cap S_2(\mathcal{U}_2, \delta)$, as defined in the proof of Lemma 2.5. Consequently, in order to derive the desired claim, it is sufficient to show that $S_1(\mathcal{U}_1, \delta) \subseteq S_1(\mathcal{U}_2, \delta)$ and $S_2(\mathcal{U}_1, \delta) \subseteq S_2(\mathcal{U}_2, \delta)$. In what follows, we establish the former set inclusion; the latter one proceeds from identical arguments, where the ρ-quantities and weight quantities are interchanged.

To this end, by equation (24), it suffices to show that $\hat{m}_\ell^+(\mathcal{U}_1) \leq \hat{m}_\ell^+(\mathcal{U}_2)$ for every $\ell \in [\eta(\mathcal{U}_2), \bar{\eta}(\mathcal{U}_1)]$, given that $\eta(\mathcal{U}_2) \geq \eta(\mathcal{U}_1)$ and $\bar{\eta}(\mathcal{U}_2) \geq \bar{\eta}(\mathcal{U}_1)$. Now, let $\tilde{\rho}_1$ and $\tilde{\rho}_2$ be the ρ-quantity guesses for $\rho(\mathcal{U}_1)$ and $\rho(\mathcal{U}_2)$, respectively. Further, let $\tilde{\theta}_1 = 2^{\lceil \log \delta \cdot \frac{\tilde{\rho}_1}{K} \rceil}$ and $\tilde{\theta}_2 = 2^{\lceil \log \delta \cdot \frac{\tilde{\rho}_2}{K} \rceil}$. Since $\mathcal{U}_1 \subseteq \mathcal{U}_2$, we infer that $\tilde{\rho}_1 \leq \tilde{\rho}_2$, and thus, $\frac{\tilde{\theta}_2}{\tilde{\theta}_1} = 2^\kappa$ for some $\kappa \in \mathbb{N}$. Observe that $\hat{m}_\ell^+(\mathcal{U}_1)$ is the maximal number of items in class $\mathcal{Q}^\ell$, chosen in order of increasing weights, whose total ρ-quantity is at most ρ_1^ℓ, where ρ_1^ℓ is the smallest multiple of $\tilde{\theta}_1$ which is greater or equal to $\rho(\mathcal{U}_1 \cap \mathcal{Q}^\ell)$. Similarly, $\hat{m}_\ell^+(\mathcal{U}_2)$ is the maximal number of items in class $\mathcal{Q}^\ell$, chosen in order of increasing weights, whose total ρ-quantity is at most ρ_2^ℓ, where ρ_2^ℓ is the smallest multiple of $\tilde{\theta}_2$ which is greater or equal to $\rho(\mathcal{U}_2 \cap \mathcal{Q}^\ell)$. Consequently, it suffices to show that $\rho_2^\ell \geq \rho_1^\ell$. To this end, note that

$$\rho_2^\ell = \tilde{\theta}_2 \cdot \left\lceil \frac{\rho(\mathcal{U}_2 \cap \mathcal{Q}^\ell)}{\tilde{\theta}_2} \right\rceil \geq \tilde{\theta}_1 \cdot \left\lceil \frac{\rho(\mathcal{U}_2 \cap \mathcal{Q}^\ell)}{\tilde{\theta}_1} \right\rceil \geq \tilde{\theta}_1 \cdot \left\lceil \frac{\rho(\mathcal{U}_1 \cap \mathcal{Q}^\ell)}{\tilde{\theta}_1} \right\rceil = \rho_1^\ell \ .$$

C.3 Proof of Lemma 3.2

By inequality (5), it suffices to show that, for every $k \in [n]$,

$$(1-2\epsilon) \cdot R_k(\mathcal{A}) \leq f_{\mathcal{A}}(k) \leq R_k(\mathcal{A}) \ , \tag{36}$$

where we remind the reader that $f_{\mathcal{A}}(k) = \min\{\mathcal{R}_q(\mathcal{A}) : q \in I_j\}$ for every position $k \in [n]$. Indeed, by combining (36) with (5), we obtain:

$$(1-4\epsilon) \cdot R_k(\mathcal{A}) \leq \left(1-2\epsilon-2\epsilon^2\right) \cdot R_k(\mathcal{A}) \leq \hat{f}_{\mathcal{A}}(k) \leq R_k(\mathcal{A}) \ ,$$

Now, to establish inequality (36), we separately consider the cases $k \in [1, k_{\mathrm{mid}}]$ and $k \in [k_{\mathrm{mid}} + 1, n]$:

- *Non-decreasing part ($k \in [1, k_{\mathrm{mid}}]$):* Let $j \in [j_{\mathrm{mid}}]$ be the unique index for which $k \in I_j$. Since the revenue function $k \mapsto R_k(\mathcal{A})$ is non-decreasing over I_j and e_j is the left endpoint of this interval, we obtain $f_{\mathcal{A}}(k) = R_{e_j}(\mathcal{A}) \leq R_k(\mathcal{A})$. On the other hand, by definition of e_j and I_j, we have $R_k(\mathcal{A}) < (1+\epsilon) \cdot R_{e_j}(\mathcal{A})$, and therefore $f_{\mathcal{A}}(k) = R_{e_j}(\mathcal{A}) \geq (1-\epsilon) \cdot R_k(\mathcal{A})$.

- *Non-increasing part ($k \in [k_{\mathrm{mid}}+1, n]$):* Let $j \in [j_{\mathrm{mid}}+1, m]$ be the unique index for which $k \in I_j$. Since the revenue function is non-increasing over I_j and e_j is the right endpoint of this interval, we have $f_{\mathcal{A}}(k) = R_{e_j}(\mathcal{A}) \leq R_k(\mathcal{A})$. In addition, the definition of e_j and I_j implies that $f_{\mathcal{A}}(k) = R_{e_j}(\mathcal{A}) \geq (1-\epsilon) \cdot R_k(\mathcal{A})$ for every $k \in I_j$ and $j \in [j_{\mathrm{mid}}+2, m]$. In the remaining cases, where $k \in I_{j_{\mathrm{mid}}+1}$,

$$f_{\mathcal{A}}(k) = R_{e_{j_{\mathrm{mid}}+1}}(\mathcal{A}) \geq (1-\epsilon) \cdot R_{e_{j_{\mathrm{mid}}}}(\mathcal{A}) \geq (1-2\epsilon) \cdot R_{k_{\mathrm{mid}}}(\mathcal{A}) \geq (1-2\epsilon) \cdot R_k(\mathcal{A}) \ ,$$

  where the first and second inequalities proceed from our definition of the events $e_{j_{\mathrm{mid}}+1}$ and $e_{j_{\mathrm{mid}}}$, respectively. The last inequality holds since $R_k(\mathcal{A})$ is maximized at the position k_{mid}.

C.4 Proof of Lemma 5.1

- *Inequality* (13)*:* In the case where $e_j^* \geq e_{j-1}^* \geq k_{\mathrm{mid}}$, observe that

$$\begin{aligned} \frac{\rho(S_{j+1}^{*-})}{1+w(S_{j+1}^{*+})} &\leq \frac{\rho(\mathcal{A}^*[e_{j+1}^*])}{1+w(\mathcal{A}^*[e_{j+1}^*])} \\ &= R_{e_{j+1}^*}(\mathcal{A}^*) \\ &\leq \frac{1}{1+\epsilon} \cdot R_{e_{j-1}^*}(\mathcal{A}^*) \\ &= \frac{1}{1+\epsilon} \cdot \frac{\rho(\mathcal{A}^*[e_{j-1}^*])}{1+w(\mathcal{A}^*[e_{j-1}^*])} \end{aligned}$$

$$\leq \frac{1}{1+\epsilon} \cdot \frac{\rho(S^{*-}_{j-1})/(1-\epsilon^2)}{1+w(S^{*+}_{j-1})/(1+\epsilon^2)}$$

$$\leq \left(1-\frac{\epsilon}{2}\right) \cdot \frac{\rho(S^{*-}_{j-1})}{1+w(S^{*+}_{j-1})} \ .$$

Here, the first inequality proceeds proceeds from Property 1 of Lemma 2.5, with respect to the assortment $\mathcal{U} = \mathcal{A}^*[e^*_{j+1}]$. The second inequality follows from the definition of the events $e^*_j = \max\{k \in [e^*_{j-1}+1, n] : R_k(\mathcal{A}^*) \geq \frac{1}{1+\epsilon} \cdot R_{e^*_{j-1}}(\mathcal{A}^*)\}$ in Section 3.2 over the monotone non-increasing part of the revenue function, implying that $R_{e^*_{j+1}}(\mathcal{A}^*) \leq R_{e^*_j+1}(\mathcal{A}^*) \leq \frac{1}{1+\epsilon} \cdot R_{e_{j-1}}(\mathcal{A}^*)$. The next inequality proceeds from Properties 2 and 3 of Lemma 2.5, instantiated with $\mathcal{U} = \mathcal{A}^*[e^*_{j-1}]$ and $\delta = \epsilon^2$. The last inequality holds since $\epsilon \in (0, \frac{1}{5})$.

We derive the other upper-bound in (13) using a similar sequence of inequalities:

$$\begin{aligned}
\frac{\rho(S^{*-}_{j+1})}{1+w(S^{*+}_{j+1})} &\leq \frac{\rho(\mathcal{A}^*[e^*_{j+1}])}{1+w(\mathcal{A}^*[e^*_{j+1}])} \\
&= R_{e^*_{j+1}}(\mathcal{A}^*) \\
&\leq R_{e^*_j}(\mathcal{A}^*) \\
&= \frac{\rho(\mathcal{A}^*[e^*_j])}{1+w(\mathcal{A}^*[e^*_j])} \\
&\leq \frac{\rho(S^{*-}_j)/(1-\epsilon^2)}{1+w(S^{*+}_j)/(1+\epsilon^2)} \\
&\leq \left(1+\frac{\epsilon}{2}\right) \cdot \frac{\rho(S^{*-}_j)}{1+w(S^{*+}_j)} \ ,
\end{aligned}$$

where the second inequality holds since the revenue function $k \mapsto R_k(\mathcal{A}^*)$ is non-increasing over the interval of positions $[k_{\mathrm{mid}}+1, n]$.

- *Inequality* (14)*:* In the case where $e^*_j \geq k_{\mathrm{mid}} \geq e^*_{j-1}$, observe that

$$\begin{aligned}
\frac{\rho(S^{*-}_j)}{1+w(S^{*+}_j)} &\leq \frac{\rho(\mathcal{A}^*[e^*_j])}{1+w(\mathcal{A}^*[e^*_j])} \\
&= R_{e^*_j}(\mathcal{A}^*) \\
&\leq R_{k_{\mathrm{mid}}}(\mathcal{A}^*) \\
&\leq (1+\epsilon) \cdot R_{e_{j-1}}(\mathcal{A}^*) \\
&= (1+\epsilon) \cdot \frac{\rho(\mathcal{A}^*[e^*_{j-1}])}{1+w(\mathcal{A}^*[e^*_{j-1}])} \\
&\leq (1+\epsilon) \cdot \frac{\rho(S^{*-}_{j-1})/(1-\epsilon^2)}{1+w(S^{*+}_{j-1})/(1+\epsilon^2)} \\
&\leq \left(1+\frac{3\epsilon}{2}\right) \cdot \frac{\rho(S^{*-}_{j-1})}{1+w(S^{*+}_{j-1})} \ ,
\end{aligned}$$

Here, the first inequality proceeds proceeds from Property 1 of Lemma 2.5, with respect to the assortment $\mathcal{U} = \mathcal{A}^*[e^*_{j+1}]$. The second inequality holds since the revenue function $k \mapsto R_k(\mathcal{A}^*)$ is maximized at the position k_{mid}. The third inequality holds since e^*_{j-1} is the last event in the non-decreasing part of the revenue function, meaning that the set $\{k \in [e^*_{j-1}+1, n] : R_k(\mathcal{A}^*) \geq (1+\epsilon) \cdot R_{e^*_{j-1}}(\mathcal{A}^*)\}$ is empty, and in particular, $R_{k_{\text{mid}}}(\mathcal{A}^*) \leq (1+\epsilon) \cdot R_{e^*_{j-1}}(\mathcal{A}^*)$. The next inequality proceeds from Properties 2 and 3 of Lemma 2.5, instantiated with $\mathcal{U} = \mathcal{A}^*[e^*_{j-1}]$ and $\delta = \epsilon^2$. The last inequality holds since $\epsilon \in (0, \frac{1}{5})$.

C.5 Proof of Lemma 5.2

Fix $j \in [T]$, and let $(e_{j-1}, e_j, \vec{S}_{j-1}, \vec{S}_j)$ be the corresponding dynamic programming state. We first consider the case where $j \geq j_{\text{mid}+1}$. Clearly, by our construction of the subsets $\Delta_1, \ldots, \Delta_m$ in Section 5.2, we have $S^+_j \subseteq \bigcup_{t=1}^{j} \Delta_t = \tilde{\mathcal{A}}[\tilde{e}_j]$. It immediately follows that $\rho(\tilde{\mathcal{A}}[\tilde{e}_j]) \geq \rho(S^+_j) \geq \rho(S^-_j)$, where the last inequality holds by constraint (8).

In the opposite case, $j \leq j_{\text{mid}}$, and by construction, we have $\bigcup_{t=1}^{j}(S^-_t \setminus S^+_{t-1}) = \bigcup_{t=1}^{j} \Delta_t = \tilde{\mathcal{A}}[\tilde{e}_j]$, where we define $S^+_0 = \emptyset$ for completeness. On the other hand, for every $t_1 \leq j$, the set inclusion constraint (8) implies that $S^-_{t_2} \subseteq S^+_{t_2} \subseteq S^+_{t_1}$ for every $t_2 \leq t_1$. Consequently, the subsets of items $S^-_1 \setminus S^+_0, \ldots, S^-_j \setminus S^+_{j-1}$ are pairwise disjoint. Based on the preceding observations, we have:

$$
\begin{aligned}
\rho\left(\tilde{\mathcal{A}}\left[\tilde{e}_j\right]\right) &= \rho\left(\bigcup_{t=1}^{j}(S^-_t \setminus S^+_{t-1})\right) && (37)\\
&= \sum_{t=1}^{j} \rho\left(S^-_t \setminus S^+_{t-1}\right) \\
&\geq \sum_{t=1}^{j} \left(\rho\left(S^-_t \setminus S^-_{t-1}\right) - 4\epsilon^2 \cdot \rho\left(S^-_{t-1}\right)\right) && (38)\\
&\geq \sum_{t=1}^{j} \left(\rho\left(S^-_t \setminus S^-_{t-1}\right) - \frac{4\epsilon^2}{\epsilon/2} \cdot \left(\rho\left(S^-_t\right) - \rho\left(S^-_{t-1}\right)\right)\right) && (39)\\
&\geq (1-8\epsilon) \cdot \sum_{t=1}^{j} \left(\rho\left(S^-_t\right) - \rho\left(S^-_{t-1}\right)\right) \\
&= (1-8\epsilon) \cdot \rho\left(S^-_j\right) .
\end{aligned}
$$

Here, inequality (38) immediately follows from constraint (9). Inequality (39) proceeds by remarking that constraint (12) implies in particular that $\rho(S^-_t) \geq \rho(S^-_{t-1}) + \frac{\epsilon}{2} \cdot \rho(S^-_{t-1})$ for every $t \leq j$.

C.6 Proof of Lemma 5.3

We prove by induction that $\tilde{\mathcal{A}}[\tilde{e}_j] \subseteq S^+_j$ for every $j \in [T]$. The latter set inclusion immediately implies the desired claim $w(\tilde{\mathcal{A}}[\tilde{e}_j]) \leq w(S^+_j)$, for every $j \in [T]$.

The base case of $j = 1$ follows immediately by observing that $\tilde{\mathcal{A}}[\tilde{e}_1] = \tilde{\mathcal{A}}[|\Delta_1|] = S^-_1 \subseteq S^+_1$ by the dynamic programming constraint (8). For the induction step, we separately consider two cases, depending on the value of $j \in [2, T]$.

- *Case 1:* $j \in [2, j_{\text{mid}}]$. In this case, we have:

$$\tilde{\mathcal{A}}[\tilde{e}_j] = \left(\tilde{\mathcal{A}}[\tilde{e}_{j-1}] \cup \Delta_j\right) \subseteq \left(S^+_{j-1} \cup \left(S^-_j \setminus S^+_{j-1}\right)\right) \subseteq S^+_j \ ,$$

  where the first set inclusion proceeds from the induction hypothesis and the definition of Δ_j. The last set inclusion holds since $S^+_{j-1} \subseteq S^+_j$ and $S^-_j \subseteq S^+_j$ due to constraint (8).

- *Case 2:* $j \in [j_{\text{mid}} + 1, m]$. Here, we observe that

$$\tilde{\mathcal{A}}[\tilde{e}_j] = \left(\tilde{\mathcal{A}}[\tilde{e}_{j-1}] \cup \Delta_j\right) \subseteq \left(S^+_{j-1} \cup S^+_j\right) = S^+_j \ ,$$

  where the first set inclusion proceeds from the induction hypothesis and the definition of Δ_j. The last set inclusion holds since $S^+_{j-1} \subseteq S^+_j$ due to constraint (8).

C.7 Proof of Claim 5.5

In what follows, we fix $j \in [T-1]$. Clearly, when $k \mapsto R_k(\mathcal{A})$ is non-decreasing over the entire interval $[\tilde{e}_j, \tilde{e}_{j+1}]$, this function is in particular unimodal. Otherwise, let $k \in [\tilde{e}_j + 1, \tilde{e}_{j+1}]$ be the smallest index for which $R_k(\mathcal{A}) < R_{k-1}(\mathcal{A})$. Then, observe that the expected revenue $R_k(\mathcal{A})$ can be decomposed as follows:

$$R_k\left(\mathcal{A}\right) = \frac{w_{\mathcal{A}(k)}}{1 + w_{\mathcal{A}(k)} + w(\mathcal{A}[k-1])} \cdot r_{\mathcal{A}(k)} + \frac{1 + w(\mathcal{A}[k-1])}{1 + w_{\mathcal{A}(k)} + w(\mathcal{A}[k-1])} \cdot R_{k-1}\left(\mathcal{A}\right) \ .$$

Consequently, $R_k(\mathcal{A})$ is a strict convex combination between $r_{\mathcal{A}(k)}$ and $R_{k-1}(\mathcal{A})$ since Assumption 2.1 implies in particular that $w_{\mathcal{A}(k)} > 0$. It immediately follows that $r_{\mathcal{A}(k)} \leq R_k(\mathcal{A})$. Based on this observation, we show by induction that $r_{\mathcal{A}(t)} \leq R_t(\mathcal{A}) \leq R_{t-1}(\mathcal{A})$ for every $t \geq k$. Indeed, note that $r_{\mathcal{A}(t+1)} \leq r_{\mathcal{A}(t)} \leq R_t(\mathcal{A})$, where the former inequality holds since $\mathcal{A}$ introduces items by non-increasing prices over the interval of positions $[\tilde{e}_j + 1, \tilde{e}_{j+1}]$ and the latter follows from the induction hypothesis. Using a similar reasoning as above, it is easy to verify that $R_{t+1}(\mathcal{A})$ is a strict convex combination between $r_{\mathcal{A}(t+1)}$ and $R_t(\mathcal{A})$. Hence, we conclude that $r_{\mathcal{A}(t+1)} \leq R_{t+1}(\mathcal{A}) \leq R_t(\mathcal{A})$.

C.8 Proof of Inequality (17)

We begin by highlighting a basic property of the assignment $\tilde{\mathcal{A}}$, allowing us to compare the positions e_j and $\tilde{e_j}$. The proof of Claim C.1 is deferred to the end of this section.

Claim C.1. *$\tilde{e_j} \leq e_j$ for every $j \in [j_{\text{mid}}]$, and $e_j \leq \tilde{e_j}$ for every $j \in [j_{\text{mid}} + 1, T]$.*

Now, in order to establish (17), we separately bound the expressions $\sum_{j=1}^{j_{\text{mid}}} (\sum_{k \in \tilde{I}_j} \lambda_k) \cdot \alpha_j$ and $\sum_{j=j_{\text{mid}}+1}^{T} (\sum_{k \in \tilde{I}_j} \lambda_k) \cdot \alpha_j$, where we use the shorthand notation $\alpha_j = \frac{\rho(S^-_j)}{1+w(S^+_j)}$.

Letting $\alpha_0 = 0$ for simplicity of notation, the former expression is lower-bounded by

$$\sum_{j=1}^{j_{\text{mid}}} \left(\sum_{k \in \tilde{I}_j} \lambda_k\right) \cdot \alpha_j \quad = \quad \sum_{j=1}^{j_{\text{mid}}} \left(\sum_{k=\tilde{e}_j}^{k_{\text{mid}}} \lambda_k\right) \cdot (\alpha_j - \alpha_{j-1})$$

$$
\begin{aligned}
&\geq \sum_{j=1}^{j_{\text{mid}}} \left(\sum_{k=e_j}^{k_{\text{mid}}} \lambda_k \right) \cdot (\alpha_j - \alpha_{j-1}) \\
&= \sum_{j=1}^{j_{\text{mid}}} \left(\sum_{k \in I_j} \lambda_k \right) \cdot \alpha_j \ , \qquad (40)
\end{aligned}
$$

where the inequality follows from Claim C.1, implying in particular that $e_j \geq \tilde{e}_j$ for every $j \in [1, j_{\text{mid}}]$, along with constraint (12), by which we derive that $\alpha_j \geq \alpha_{j-1}$.

Now, in order to lower-bound the latter expression, for every $j \in [j_{\text{mid}} + 1, T]$, we define $\bar{\alpha}_j = \min\{\alpha_t : t \in [j_{\text{mid}} + 1, j]\}$. By constraint (13), we have

$$
\left(1 - \frac{\epsilon}{2}\right) \cdot \alpha_j \leq \bar{\alpha}_j \leq \alpha_j \ . \qquad (41)
$$

This inequality is established by observing that, for every integer $t \in [j_{\text{mid}} + 1, j - 1]$ such that $j - t$ is even:

$$
\alpha_t \geq \left(1 - \frac{\epsilon}{2}\right)^{\frac{t-j}{2}} \cdot \alpha_j \geq \alpha_j \ ,
$$

where the first inequality proceeds by iteratively utilizing constraint (13). Similarly, for every integer $t \in [j_{\text{mid}} + 1, j - 1]$ such that $j - t$ is odd, we have

$$
\alpha_t \geq \left(1 - \frac{\epsilon}{2}\right)^{\frac{t-j+1}{2}} \cdot \alpha_{j-1} \geq \alpha_{j-1} \geq \frac{1}{1 + \epsilon/2} \cdot \alpha_j \geq \left(1 - \frac{\epsilon}{2}\right) \cdot \alpha_j \ .
$$

By combining the latter two inequalities, we obtain that $\bar{\alpha}_j \geq (1 - \frac{\epsilon}{2}) \cdot \alpha_j$. Consequently, we have

$$
\begin{aligned}
\sum_{j=j_{\text{mid}}+1}^{T} \left(\sum_{k \in \tilde{I}_j} \lambda_k \right) \cdot \alpha_j &\geq \sum_{j=j_{\text{mid}}+1}^{T} \left(\sum_{k \in \tilde{I}_j} \lambda_k \right) \cdot \bar{\alpha}_j \\
&= \left(\sum_{k=k_{\text{mid}}+1}^{n} \lambda_k \right) \cdot \bar{\alpha}_T + \sum_{j=j_{\text{mid}}+1}^{T-1} \left(\sum_{k=k_{\text{mid}}}^{\tilde{e}_j} \lambda_k \right) \cdot (\bar{\alpha}_j - \bar{\alpha}_{j+1}) \\
&\geq \left(\sum_{k=k_{\text{mid}}+1}^{n} \lambda_k \right) \cdot \bar{\alpha}_T + \sum_{j=j_{\text{mid}}+1}^{T-1} \left(\sum_{k=k_{\text{mid}}}^{e_j} \lambda_k \right) \cdot (\bar{\alpha}_j - \bar{\alpha}_{j+1}) \\
&= \sum_{j=j_{\text{mid}}+1}^{T} \left(\sum_{k \in I_j} \lambda_k \right) \cdot \bar{\alpha}_j \\
&\geq (1 - \frac{\epsilon}{2}) \cdot \sum_{j=j_{\text{mid}}+1}^{T} \left(\sum_{k \in I_j} \lambda_k \right) \cdot \alpha_j \ , \qquad (42)
\end{aligned}
$$

where the first and third inequalities follow from (41). The second inequality proceeds from Claim C.1, implying in particular that $\tilde{e}_j \geq e_j$ for every $j \in [j_{\text{mid}} + 1, T]$, and the fact that $\bar{\alpha}_j \geq \bar{\alpha}_{j+1}$ by definition of these terms.

To conclude, we observe that inequality (17) immediately follows from (40) and (42).

Proof of Claim C.1. For every $j \in [j_{\text{mid}}+1, m]$, we have $S_j^+ \subseteq \bigcup_{t=1}^{j} \Delta_t$ by definition of Δ_j. Therefore, $\tilde{e}_j = |\bigcup_{t=1}^{j} \Delta_t| \geq |S_j^+| \geq e_j$, where the last inequality proceeds from constraint (11). On the other hand, for every $j \in [1, j_{\text{mid}}]$, we show that $\tilde{e}_j \leq e_j$ by induction over j.

- *Base case ($j = 1$):* We have

$$\tilde{e}_j = |\Delta_1| = \left|S_1^-\right| \leq \left|S_1^+\right| \leq e_1 \ ,$$

where the last inequality here immediately follows from constraint (11).

- *Induction step ($j \in [2, j_{\text{mid}}]$):* Here, we argue that

$$\tilde{e}_j = \tilde{e}_{j-1} + |\Delta_j| = \tilde{e}_{j-1} + \left|S_j^- \setminus \left(S_{j-1}^+ \cup \left(\bigcup_{t=1}^{j-1} \Delta_t\right)\right)\right| \leq e_{j-1} + \left|S_j^- \setminus S_{j-1}^+\right| \leq e_j \ ,$$

where the first equality is due to the fact that $\tilde{e}_j = \sum_{t=1}^{j} |\Delta_t|$ since the subsets $\Delta_1, \ldots, \Delta_j$ are mutually disjoint. The first inequality proceeds from the induction hypothesis. The last inequality immediately follows from constraint (11).

■

D EM Algorithm

Starting with initial parameters $\vec{\lambda}^{(0)} = (\frac{1}{8})_{k\in[8]}$ and $\vec{\beta}^{(0)} = (0)_{k\in[8]}$, as well as a uniform prior on the realizations of the consideration sets $\vec{\pi}_t^{(0)} = (\frac{1}{8})_{k\in[8]}$ for each observation $t \in \mathcal{T}$, our EM algorithm iteratively computes parameters of the D-MNL model though posterior updates, using a convex surrogate of the log-likelihood function. Specifically, for every $\ell \geq 1$, we compute $\vec{\beta}^{(\ell)}$, $\vec{\lambda}^{(\ell)}$, and $(\vec{\pi}_t^{(\ell)})_{t\in\mathcal{T}}$ as follows:

1. *E-step:* We begin by computing the expected value of the log-likelihood function with respect to the realizations of the consideration sets:

$$\mathcal{L}\left(\vec{\beta}\middle|\,\vec{\lambda}^{(\ell-1)}\right) = \sum_{t\in\mathcal{T}} z_{j,t} \cdot \sum_{k=1}^{8} \pi_{t,k}^{(\ell-1)} \log\left(p_{\vec{\beta},\vec{e}_k}(t,j)\right) \ ,$$

where, for every $k \in [8]$, we denote by $\vec{e}_k$ the distribution over consideration sets such that the consideration set indexed by k occurs with probability 1. It is worth noting that, in contrast to the original log-likelihood function in (22), the expected log-likelihood $\mathcal{L}(\vec{\beta}|\vec{\lambda}^{(\ell-1)})$ is a concave function of the parameter $\vec{\beta}$.

2. *M-step:* We update the parameters of the D-MNL model. To this end, we define $\vec{\beta}^{(\ell)}$ as the vector $\vec{\beta} \in \mathbb{R}^{14}$ that maximizes the expected log-likelihood function $\mathcal{L}(\vec{\beta}|\vec{\lambda}^{(\ell-1)})$ defined in the E-step. Next, $\vec{\lambda}^{(\ell)}$ is computed by solving the convex optimization problem derived from our original MLE problem (22) by fixing $\beta = \vec{\beta}^{(\ell)}$, namely

$$\vec{\lambda}^{(\ell)} = \text{argmax}_{\vec{\lambda}\in\Lambda} \sum_{t\in\mathcal{T}} z_{j,t} \cdot \log\left(p_{\vec{\beta}^{(\ell)},\vec{\lambda}}\right) \ .$$

Lastly, for each observation $t \in \mathcal{T}$, the probability $\vec{\pi}_{t,k}^{(\ell)}$ that the consideration set $k \in [8]$ occurs is updated according to the posterior rule:

$$\vec{\pi}_{t,k}^{(\ell)} = \frac{\sum_{j \in S_t} \lambda_k^{(\ell)} \cdot z_{t,j} \cdot p_{\vec{\beta}^{(\ell)}, \vec{e}_k}(t,j)}{(1 - \sum_{q=1}^{8} \lambda_q^{(\ell)}) \cdot (1 - \sum_{j \in S_t} z_{t,j}) + \sum_{q=1}^{8} \sum_{j \in S_t} \lambda_q^{(\ell)} \cdot z_{t,j} \cdot p_{\vec{\beta}^{(\ell)}, \vec{e}_q}(t,j)} \ .$$

It is worth noting that the maximization of the expected log-likelihood function $\beta \mapsto \mathcal{L}(\vec{\beta}|\vec{\lambda}^{(\ell-1)})$ is the main computational bottleneck of the EM-algorithm.