Product Ranking on Online Platforms

Product Ranking on Online Platforms

Mahsa Derakhshan
University of Maryland, Department of Computer Science, College Park, MD, mahsa@cs.umd.edu

Negin Golrezaei
MIT Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA, golrezae@mit.edu

Vahideh Manshadi
Yale School of Management, Yale University, New Haven, CT, vahideh.manshadi@yale.edu

Vahab Mirrokni
Google Research, New York, NY, mirrokni@google.com

On online platforms, consumers face an abundance of options that are displayed in the form of a position ranking. Only products placed in the first few positions are readily accessible to the consumer, and she needs to exert effort to access more options. For such platforms, we develop a two-stage sequential search model where in the first stage, the consumer sequentially screens positions to observe the preference weight of the products placed in them and forms a consideration set. In the second stage, she observes the additional idiosyncratic utility that she can derive from each product and chooses the highest-utility product within her consideration set. For this model, we first characterize the optimal sequential search policy of a welfare-maximizing consumer. We then study how platforms with different objectives should rank products. We focus on two objectives: (i) maximizing the platform's market share and (ii) maximizing the consumer's welfare. Somewhat surprisingly, we show that ranking products in decreasing order of their preference weights does not necessarily maximize market share or consumer welfare. Such a ranking may shorten the consumer's consideration set due to the externality effect of high-positioned products on low-positioned ones, leading to insufficient screening. We then show that both problems—maximizing market share and maximizing consumer welfare—are NP-complete. We develop novel near-optimal polynomial-time ranking algorithms for each objective. Further, we show that even though ranking products in decreasing order of their preference weights is suboptimal, such a ranking enjoys strong performance guarantees for both objectives. We complement our theoretical developments with numerical studies using synthetic data in which we show (1) that heuristic versions of our algorithms that do not rely on model primitives perform well and (2) that our model can be effectively estimated using a maximum likelihood estimator.

Key words: two-stage consumer search, product ranking, search costs, online platforms

---

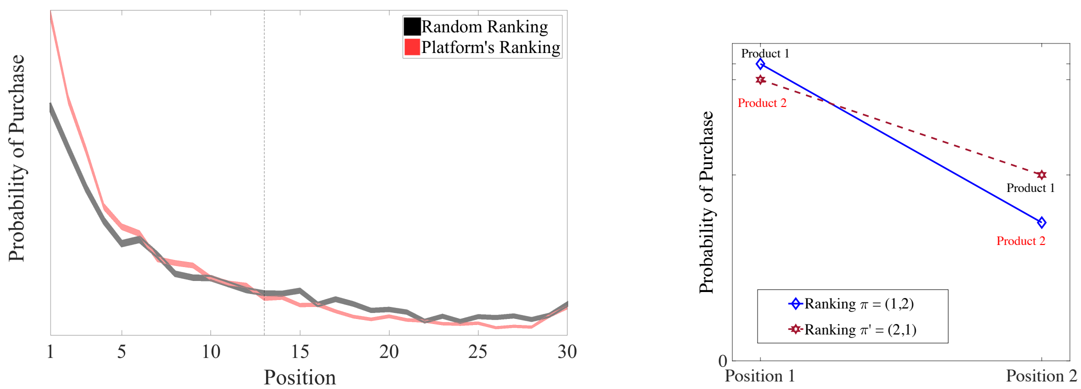
1. Introduction

With the record-breaking growth in online shopping (WSJ 2017) and the profusion of products that can be offered online, the role of online platforms in shaping consumer choice has become increasingly important. In a typical online shopping experience, upon entering a keyword, the consumer is presented with numerous search results displayed in a sequence of panels or webpages, with most of the options/products being initially invisible. For the most part, the number of options that a platform offers far exceeds the consumer's attention and cognitive resources. Millward Brown reports that 70% of Amazon users do not go beyond the first page of search results (clavisinsight.com 2015). As a result of such consumer behavior, it is well documented that the *ranking* of the products displayed to the consumer has a substantial impact on her choice (See Section 2 for references). This makes product ranking an instrumental lever for a platform's achievement of various goals in line with its business model, such as increasing market share or improving consumer welfare. Considering this, in the present work, we study the product-ranking problem that platforms face each time a consumer enters a keyword.

In the following, we first provide insights into consumer behavior by analyzing query-level data from a large online platform. Then, we provide a summary of our contributions.

Empirical Evidence for Impact of Ranking on Consumer Choice: We analyze a dataset from a platform that, for each query, displays a number of options, ranging from a few to a few dozen. In Figure 1a, we present aggregate-level data for queries in response to which 30 options were displayed. We plot the probability of purchase for each position for two ranking mechanisms: random ranking and the platform's ranking policy, where under the random ranking, all possible permutations are equally likely. Under both ranking mechanisms, products in lower positions are less likely to be purchased. Such a decreasing trend suggests that consumer search is sequential and costly. However, this decline is sharper when the platform's ranking is used. This trend provides strong evidence of the externality that products in higher positions impose on lower-positioned ones. In random ranking, there is no correlation between a product's utility and its position; therefore, top positions may be filled with low-utility products. In that case, the consumer is more likely to continue her search beyond the few top positions. The platform's ranking, in contrast, tends to place higher-utility products in higher positions; the benefit of doing so is evident in the increased purchase probability of the first five positions in this setting compared to the random ranking setting. At the same time, due to the externality that those high-utility products impose on lower positions, the purchase probability for positions lower than 13 is smaller in this ranking compared to the random ranking. The overall probability of purchase under the platform's ranking is 5% higher than that under the random ranking.

To further investigate the externality effect, we analyze purchase probabilities on an individual product level for the subset of queries in response to which two options are displayed. Under the random ranking, for each pair of products, Product 1 and Product 2, we trace all the queries in response to which these two products are displayed. We denote the ranking that displays Product



(a) (b)

Figure 1 Figure 1a shows the probability of purchase vs. the position under random ranking and the platform's ranking. The width of each line represents the mean standard error. Figure 1b shows the probability of purchase under ranking $\pi = (1,2)$ vs. $\pi' = (2,1)$. Note that Product 1 is more popular than Product 2. The absolute values are hidden for confidentiality, but the ratios are preserved.

1 in the top position (and thus Product 2 in the bottom position) by $\pi = (1,2)$ and the ranking with reverse order by $\pi' = (2,1)$. We focus on a pair of products that is among the most frequently displayed pairs. For this pair, Product 1 is significantly more popular than Product 2. (For each product, we have a "popularity" measure that can be viewed as a proxy for its expected utility.) Figure 1b compares the purchase probability for each product under rankings π and π'. We observe that placing Product 1 in position 2, i.e., the bottom position, results in a 37% decrease in its purchase probability, whereas for Product 2, lowering its position reduces its purchase probability by 51%. This further highlights the higher externality that the popular product in the top position imposes on the product in the bottom position, compared to the case where the positions of these products are reversed. Interestingly, the overall purchase probability under ranking π is 7% lower than under ranking π'. We note that this suboptimal behavior is not an exception in our empirical study. In fact, for 75% of pairs that appear more than 1000 times in our dataset, placing the more popular product in the top position results in a smaller overall probability of purchase. This suggests that *placing the most popular product in the higher position is suboptimal for maximizing the overall probability of purchase.*

These empirical findings raise a key question that motivates our work: Facing such consumer search behavior, how should an online platform rank its products?

Summary of Contributions: Motivated by our empirical observations, we (1) develop a novel two-stage consumer search model that captures key features observed in our empirical study as well as in other empirical work, (2) characterize the consumer-optimal search policy under our model, (3) design polynomial-time approximation schemes (PTASs) for the platform's rank optimization problem for the two objectives of maximizing market share and maximizing consumer welfare, and (4) analyze the performance of a greedy algorithm that orders the products in decreasing order of their "popularity" for both aforementioned objectives.

A New Two-Stage Consumer Search Model: We develop a new model of consumer search and choice that captures the key features of consumer behavior. Consistent with the choice-modeling literature, we assume that the utility that a consumer derives from a product consists of two parts: an intrinsic utility, which is the logarithm of the product's *preference weight*, and an idiosyncratic part. We assume that both parts are independently drawn from two different distributions and the distribution for the idiosyncratic part is the standard Gumbel. Initially, the consumer does not know either part. However, she knows the distributions. In particular, the consumer goes through a two-stage process where, in the first stage, she *screens* positions to observe the preference weights of the products placed in those positions and forms a *consideration set*, and, in the second stage, she further evaluates the products in her consideration set to observe the idiosyncratic part of their utility. Then, the consumer purchases the product in her consideration set—which includes an outside option—that has the greatest utility. This amounts to having a Multinomial Logit choice model among the products *within* the consideration set.

In the first stage, the consumer undertakes a sequential screening process to observe the preference weights of the products and forms her consideration set. At each step of this process, she decides whether to stop her screening and finalize her consideration set or to screen another position by paying its position-dependent search cost and add the product placed in that position to her consideration set. The search cost that the consumer incurs accounts for both the screening cost she pays to observe its preference weight and the evaluation cost she incurs when she evaluates the product to observe its idiosyncratic part in the second stage, after the consideration set is formed. Such a modeling assumption is aligned with the consumer behavior underlying the consider-then-choose literature. We highlight that screening a product only reveals its preference weight and not the idiosyncratic part of its utility. The latter is revealed in the second stage, after the consideration set is formed. In this sequential process, the consumers' goal is to maximize their expected welfare, which is the utility of the purchased product minus the search costs that they incurred during the first stage. We assume that the consumer population comprises different types, where the search costs vary across types: more patient types have lower search costs compared to less patient ones. Further, we assume that for any type, position-dependent search costs are higher for lower positions.

Consumers' Optimal Search Policy: We show that the consumer's optimal search strategy is an index-based policy that can be described with a set of indices that we call *reservation prices* (see Theorem 1). Under the index-based policy, if the consumer finds it optimal to screen another position, she chooses the one with the highest reservation price, which is the topmost unscreened position. Further, the consumer stops her search if the *sum of the weights* of the products in her consideration set exceeds the reservation prices of all the unscreened positions. This stems from the fact that all the products—not only the one with the highest weight—in the consideration set have a chance of being purchased.

We now highlight the important features of the index-based optimal policy and its resultant consideration set. Under the index-based policy, (i) because search costs increase as the visibility of positions decreases, the consumer screens products starting from the highest position and moves downward; (ii) the consideration set is *endogenously* determined by both the ranking and the consumer type; (iii) placing products with high weights in higher positions may reduce the size of the consideration set, which captures the *externality* that such products impose on lower-positioned ones; and (iv) consumer types with smaller search costs will form larger consideration sets.

Platform's Ranking Problems: Under our novel consumer search model, we study the platform's optimal-ranking problems. We assume that the platform knows the preference weight of all products. However, it does not observe the consumer's type and only knows the type distribution. Thus, the platform faces the challenge of characterizing a ranking that performs well in expectation where the expectation is taken with respect to the randomness in the consumer's type. We focus on two objectives: maximizing the platform's market share and maximizing consumers' welfare. The platform's market share is the probability that the consumer makes a purchase. For many platforms, maximizing the consumer's measure of "satisfaction", such as her welfare or her probability of purchase, is the foremost objective, as these platforms seek to build a long-run reputation of helping consumers find their favorite products without too much effort.

For the objectives of market share and consumer welfare, we show that the problem of optimizing ranking is NP-complete (see Theorem 2 and Proposition 1). In light of this hardness result, we then develop and analyze approximation algorithms. For the objective of market share, we measure the performance of an algorithm using both multiplicative and additive approximation factors. In contrast, for the objective of consumer welfare, we focus on additive factors because optimal consumer welfare can be negative. Further, we note that even though the two objectives bear some similarity, there is a crucial difference between them. In particular, the objective of consumer welfare takes into account the search costs that the consumer incurs; as a result, consumer welfare may decrease if she screens more positions. This is in contrast to the behavior of market share, which can only increase with more screened positions. Due to this difference, each objective requires unique theories and analyses.

We first focus on market share maximization. We analyze the performance of a greedy algorithm that we call *w-ordered*. This algorithm sorts the products in decreasing order of their weights; that is, positions with higher visibility are filled with products with higher weights than those with lower visibility. This algorithm is practically appealing, as it does not require any knowledge of search costs or the type distribution. We show that the w-ordered ranking achieves a multiplicative approximation factor of 1/2 and an additive factor of 0.1716 (Theorem 3). Further, we show that our analysis of the w-ordered ranking for both factors is tight (Example EC.1).

We further present a near-optimal approximation algorithm that, for a fixed $\epsilon > 0$, achieves a multiplicative approximation factor of $(1-\epsilon)$ and an additive factor of ϵ. Let ρ be the ratio of the maximum preference weight to the minimum preference weight. Then, for $\rho = O(1)$, the running time of our algorithm is polynomial in the number of products and scales quadratically with the number of consumer types (see Algorithm 1 and Theorem 4). Our algorithm relies on a rounding procedure in which we bucket the products based on their weights using a geometric grid. We round up their weights given the bucket into which each product falls. We then utilize the rounded weights to reformulate the market share maximization problem as a dynamic program with a state space whose size remains polynomial in the number of products and linear in the number of consumer types. By replacing the weights with their rounded versions, we can reduce the complexity of the problem while hoping that the objective value does not change significantly. Unfortunately, this is not the case because rounding errors impact the formation of the consideration set. In fact, any error in the formation of the consideration set of a consumer type can lead to a substantial loss. We overcome this challenge by restricting the search space of our algorithm to the class of *Low-Weight Priority* (LWP) rankings (see Definition 1) that contains an ϵ-optimal ranking. In the LWP class, within each bucket, low-weight products have higher priority compared to high-weight ones in filling the top positions. This restriction allows us to use the weights themselves for formation of consideration sets and leads to generation of an algorithm with near-optimal performance.

We next study the problem of maximizing consumer welfare. We present our tight analysis of w-ordered ranking and show that it achieves an additive approximation factor of 0.6931 (Proposition 2). We develop an algorithm that achieves an additive approximation factor of ϵ (see Theorem 5) under the assumption that $\rho = O(1)$. Our algorithm combines a low-complexity brute-force procedure with a greedy algorithm and its running time is polynomial in the number of products and linear in the number of types.

Finally, in Section 7, we carry out numerical studies using synthetic data. First, we develop heuristic versions of our algorithms that do not rely on model primitives. Then, we evaluate the performance of our heuristics as well as our PTAS (Algorithm 1) in two settings that differ in the platform's access to the model parameters. In the first setting, model parameters are known whereas in the second one they are estimated using the procedure we describe in Appendix EC.14.

Our numerical results suggest that model parameters can be estimated effectively and our PTAS algorithm as well as the aforementioned heuristics are robust to small estimation errors.

The rest of the paper is organized as follows. In Section 2, we review the related literature. In Section 3, we introduce our model of consumer search as well as the platform's objectives. Section 4 is devoted to characterizing her optimal policy. In Sections 5 and 6, we turn our attention to studying the platform's rank optimization problem for the objectives of maximizing market share and maximizing consumer welfare, respectively. Section 7 presents our complementary numerical studies, and Section 8 concludes the paper. The detailed proofs of all the statements, our preliminary empirical evidence, and the description of the estimation process are provided in appendices.

2. Related Work

Our work relates to and contributes to several streams of literature.

Product Ranking and Empirical Studies: A set of papers in the marketing literature empirically study the causal impact of ranking and search costs on consumer choice (Kim et al. 2010, De los Santos and Koulayev 2013, Ghose et al. 2014, Jeziorski and Segal 2015, Chen and Yao 2016, Ursu 2016).[1] Using various techniques, all the aforementioned papers find that placing a product in positions with high visibility increases its probability of purchase. Another finding consistent across this line of work is that the w-ordered ranking improves consumer welfare compared to the currently employed ranking methods. We note, however, that these works do not claim that the w-ordered ranking maximizes consumer welfare. The focus of our work is different from that of the above papers; besides proposing a new consumer search model, we also focus on designing efficient algorithms for a platform's optimal-ranking problems.

From a modeling perspective, closest to our work is that of Ursu (2016), who analyzes a unique dataset from Expedia that includes the purchasing outcomes of the random ranking as well as the platform's ranking. The main difference between the modeling approach of Ursu (2016) and ours is related to the a priori knowledge of consumers about products. In particular, Ursu (2016) assumes that consumers know the preference weight of *all* products (or, equivalently, learn them for free) and only need to learn the idiosyncratic part of the utility. However, as discussed in the introduction, given that most products are not initially visible to the consumer and that she needs to exert effort to even take a first glance, such an assumption may not bear out in practice. In our work, we relax such an assumption by proposing a two-stage search model where, in the first stage, we explicitly account for the search cost of learning the preference weights.

[1] Recently, via running a field experiment, Ngwe et al. (2018) studied the impact of increasing search costs on consumer behavior on online platforms.

Rank Optimization: A few recent papers study platforms' rank optimization problems utilizing models that reflect the impact of position on consumer choice (Davis et al. 2013, Abeliuk et al. 2015, Aouad and Segev 2015, Abeliuk et al. 2016, Gallego et al. 2016, Lei et al. 2018). [2]

Here, we mainly discuss the differences in the modeling approaches of our work and the above papers. Davis et al. (2013), Abeliuk et al. (2015, 2016), and Lei et al. (2018) study an MNL choice model where the impact of position on a product's intrinsic utility is incorporated exogenously (for example, by multiplying the weight of the product by an exogenous position-dependent factor).[3] Aouad and Segev (2015) take a different approach, proposing a consider-then-choose model in which the population is composed of different types of consumers; for each type, the size of the consideration set is exogenously determined and is independent of the ranking. Gallego et al. (2016) take a similar approach and study a closely related problem. In contrast to the above papers, we construct a choice model in which consideration sets are *endogenously* determined by the ranking.

Position Auctions: Motivated by online advertising platforms that sell multiple ad slots through auctions and the externality that ads in higher positions impose on those in lower positions, a stream of papers in computer science and economics have proposed various models that incorporate the impact of position on the click-through rate (CTR). This line of work mainly focuses on properties of the generalized second-price (GSP) auction and its relation to the Vickrey-Clarke-Groves (VCG) mechanism. Varian (2007) and Edelman et al. (2007) model the impact of position by multiplying the CTR by an exogenous position-dependent factor. However, as shown in Joachims et al. (2007), Craswell et al. (2008), and Goldman and Rao (2016), there is limited empirical evidence that supports such a modeling approach. Meanwhile, Aggarwal et al. (2008), Kempe and Mahdian (2008), and Athey and Ellison (2011) have proposed models that capture the externality that higher-positioned products impose on lower-positioned ones. All models assume that the user views ads sequentially starting from the highest position; at any position, the user may decide to stop her search independently with an ad-specific probability. However, Jeziorski and Segal (2015) provide empirical evidence that some users click on a lower position before clicking on a higher one.

In a recent work, Chu et al. (2017) propose an elegant sequential search model based on the Weitzman (1979) framework. They primarily focus on auction design for selling the top positions, proposing a new mechanism called surplus-ordered ranking (SOR). They show that their mechanism achieves a constant factor in maximizing a fairly general objective that encompasses consumer and seller surplus as well as the platform's revenue. In contrast to the setting of Chu et al. (2017),

[2] See also Ba et al. (2020) that studies a rank optimization problem in a sequential product presentation framework motivated by voice-based virtual assistants.

[3] See also Abeliuk et al. (2015) and Vaccari et al. (2018) for studies of the rank optimization problem in the presence of social influence and learning and L'Ecuyer et al. (2017) for a setting with reputation effects. In all these papers, the impact of position is incorporated by exogenous multiplicative factors.

we do not consider an auction design problem, as we do not study platforms that sell the top positions. We discuss the model of Chu et al. (2017) as well as other prior related models in Section 3.1.1.

3. Model and Preliminaries

Consider an online platform that displays n items indexed by $i \in [n] = \{1, 2, \ldots, n\}$. The items are ordered in n positions, indexed by $j \in [n]$, where the positions with lower indices have a higher visibility. Depending on the market that the platform hosts, an item can be a product, an app, a service, or an ad featuring one of these. The platform displays the items to the consumer in the form of an ordering, seeking the desired action from her. The consumer's action can take the form of a click, a purchase, or a download. To be concrete, hereafter, we refer to the items and the desired action as *products* and *purchase*, respectively. The platform's goal is to find the *optimal* ranking of the products. To be more precise, the platform's decision is a permutation $\pi : [n] \to [n]$, where $\pi(j) = i$ means product i is placed in position j. We will elaborate on the platform's objective in Section 3.2.

3.1. Consumer's Search Behavior and Choice Model

We first discuss the consumer's utility derived from each product $i \in [n]$. The utility consists of two parts: a shared intrinsic utility denoted by U_i and an idiosyncratic part denoted by Z_i. We adopt the convention of representing random variables U_i and Z_i by uppercase letters and their realizations by the lowercase letters u_i and z_i. A priori, the consumer does not know the intrinsic utility of product $i \in [n]$ and needs to exert effort to observe it. We model this lack of information on the consumer side by assuming that U_i is a random variable. However, we stress that after exerting the necessary effort, the consumer will know the realized intrinsic utility u_i.

We assume that the two parts of the utility are independent, i.e., U_i and Z_i are independent of each other. For tractability, we assume that Z_i's are drawn independently from the standard Gumbel distribution. Such an assumption is very common in the choice-modeling literature; see, for example, Ben-Akiva et al. (1985), Mahajan and Van Ryzin (2001), and Talluri and Van Ryzin (2004). As will become clear later, we will work with the exponent of the intrinsic utilities, which are called the *preference weights* (for the sake of brevity, hereafter, we refer to them simply as weights). Therefore, we define $W_i = e^{U_i}$ for any $i \in [n]$. We assume that W_i's are i.i.d. samples from a probability distribution $f_W : [w_{\min}, w_{\max}] \to \mathbb{R}^+$ where $w_{\min}$ and $w_{\max} \in \mathbb{R}^+$ are finite positive numbers. Further, we define $\rho = \frac{w_{\max}}{w_{\min}} > 1$ as the ratio of the maximum value that W takes to its minimum value, and we assume ρ is constant. In fact, as we will discuss later, our algorithms are polynomial only when $\rho = O(1)$. In our model, consumers are oblivious to the platform's ranking algorithm; that is, f_W is not a function of the ranking algorithm. This assumption, which is prevalent in the literature (Chu et al. (2017), Chen and Yao (2016)), is practically reasonable. This is due to the complexity of ranking algorithms employed by platforms and consumers' lack of knowledge with regard to platforms' objective.

The consumer also has an outside option, represented by product 0. We normalize the weight of the outside option by setting $w_0 = 1$. However, the idiosyncratic part Z_0 is an independent draw from the standard Gumbel distribution. We assume that the consumer knows the distributions f_W and f_Z and that $w_0 = 1$. However, initially, she does not know the product weights, and she needs to exert an effort to observe them. This assumption is partly motivated by the large number of products on online platforms, most of which are initially invisible to the consumer.

We now describe the consumer's search process to observe her utility for a product. The consumer goes through a two-stage process. In the first stage, she conducts a costly sequential search to *screen* positions. In particular, when the consumer screens a position, she observes the weight of the product placed in that position, and she adds that product to her *consideration set*. In the second stage, she observes the idiosyncratic part of the utility of the products *within* her consideration set, and she selects a product (from her consideration set) with the highest total utility (i.e., the summation of the logarithm of the weight and the idiosyncratic part). In the following, we provide a micro-foundation model for how the consumer optimally conducts the search process and forms her consideration set.

Our search model is motivated by how consumers conduct their search on online platforms. First, consumers eyeball the products in the high positions to observe basic features, which correspond to the weight in our model. However, eyeballing does not reveal all the details about the products. Therefore, after eyeballing a subset of products, the consumer *evaluates* them individually (for example, by visiting webpages of those products) in order to learn more details about those products (which correspond to the idiosyncratic part of their utility). Our two-stage search process captures such consumer behavior. Both stages of this process are costly for the consumer. In stage one, she incurs a *screening* cost for eyeballing the product, whereas in stage two, she pays an *evaluation* cost for each product in her consideration set. Consistent with some papers in the literature on consider-then-choose modeling (Hauser and Wernerfelt 1990, Wang and Sahin 2017), we assume that the consumer incorporates the evaluation cost that she will pay in the second stage *upfront* when deciding on her consideration set. Later, in Section 3.1.1, we provide more detailed discussion on how our model differs from the existing search models.

As stated earlier, to form the consideration set, consumers undergo a costly search process that highly depends on the position-dependent search costs of the consumer. We assume that our consumer population comprises K types and that the position-dependent search cost varies across types. More precisely, each type $k \in [K]$ is characterized by a set of costs $\{s_1^k, s_2^k, \ldots, s_n^k\}$ where s_j^k is the search cost of the consumer of type k to screen the product in position j. We assume s_j^k is finite and non-negative. Further, we assume that $s_{j+1}^k \geq s_j^k$ for $k \in [K]$ and $j \in [n-1]$; that is, for any type $k \in [K]$, higher positions have lower search costs. The assumption that s_j^k increases in j is in line with the empirical evidence in Kim et al. (2010), Chen and Yao (2016), and Ursu (2016). We further assume that as k increases, consumers get more patient, i.e., their search cost decreases. That is, we assume that $s_j^k \geq s_j^{k+1}$ for any $j \in [n]$ and $k \in [K-1]$. This is a reasonable assumption when one can sort the customers in terms of their patience levels. We also assume that the consumer type distribution is given by $(\theta^1, \theta^2, \ldots, \theta^K)$; that is, a customer belongs to a type $k \in [K]$ with probability θ^k.

Consumers' Search Procedure: Facing a ranking π, a consumer of type $k \in [K]$ sequentially screens positions to observe the weight of products placed in them and includes those products in her consideration set. Note that this is a finite-horizon sequential decision-making process. A non-anticipatory policy for this problem entails the following decisions: At each step, the consumer needs to decide whether to screen another position or stop her search procedure. If she decides to screen another position, then she also needs to decide which position to screen next. When she screens position j, then she observes the weight of product $i = \pi(j)$, i.e., the product placed in position j, and incurs the search cost of s_j^k, which is the sum of the screening and evaluation costs. Remark that for observing the weight of product i, the consumer only pays its screening cost. However, she also accounts for the evaluation cost that she will pay later to observe the idiosyncratic part.

Note that by adding product i, the consumer only observes its intrinsic part of the utility (equivalently, its weight), and not its idiosyncratic part. If the consumer stops, she observes the idiosyncratic part of the utility of all the products in her consideration set and purchases the one that has the highest realized utility. The consumer needs to make the above stopping decision only based on her *current* information, i.e., based on the weights of the products that she has screened so far. We denote the class of all non-anticipatory policies (as described above) by Ω. We assume that the consumer always screens the no-purchase option; that is, the consideration set of the consumer includes product 0. Thus, when the consumer stops her search, there is a chance that she leaves the platform without making any purchase.

Let $\mathcal{C}$ be the consideration set of the consumer when she stops her search. Define $\mathcal{W}$ as $\{w_i\}_{i \in \mathcal{C}}$. Note that set $\mathcal{W}$ is fully known to the consumer. Then, the expected welfare of a consumer of type k under ranking π is given by

$$\mathsf{Wel}_\pi^k(\mathcal{C}, \mathcal{W}) = \mathbb{E}\Big[\max_{i \in \mathcal{C} \cup \{0\}} \{\log(w_i) + Z_i\}\Big] - \sum_{i \in \mathcal{C}} s_{\pi^{-1}(i)}^k, \tag{1}$$

where the expectation is with respect to (w.r.t.) Z_i, $i \in \mathcal{C} \cup \{0\}$. Note that the outside option, product 0, is always included in the consideration set; however, to simplify the notation, we do not explicitly include it in set $\mathcal{C}$. Here, the first term is the expected maximum utility among the products within the consideration set $\mathcal{C}$, and the second term is the incurred search cost. Observe that the search cost depends on the position of the products in $\mathcal{C}$.

The consumer's problem in this sequential search process can be formulated as a dynamic programming (DP) problem. Let $V_\pi^k(\mathcal{C}, \mathcal{W})$ be the expected welfare of the consumer of type k if she follows an optimal policy from this step onward when the products in $\mathcal{C}$ with weights $\mathcal{W} = \{w_i\}_{i\in\mathcal{C}}$ are already added to her consideration set. In the rest of this section, to simplify the notation, we denote $V_\pi^k(\mathcal{C}, \mathcal{W})$ by $V(\mathcal{C}, \mathcal{W})$. Then, function V should satisfy the following recursive equation:

$$V(\mathcal{C}, \mathcal{W}) = \max\left\{\mathsf{Wel}_\pi^k(\mathcal{C}, \mathcal{W}), \quad \max_{j\in[n]\setminus\pi^{-1}(\mathcal{C})} \mathbb{E}\Big[V(\mathcal{C} \cup \{\pi(j)\}, \mathcal{W} \cup \{W_{\pi(j)}\})\Big]\right\}, \tag{2}$$

where with a slight abuse of notation, $\pi^{-1}(\mathcal{C})$ is the set of positions of the products within set $\mathcal{C}$, i.e., $j \in \pi^{-1}(\mathcal{C})$ if and only if $\pi(j) \in \mathcal{C}$. Here, the first term, i.e., $\mathsf{Wel}_\pi^k(\mathcal{C}, \mathcal{W})$, which is defined in (1), is the expected welfare of the consumer if she stops her search. The second term, i.e., $\max_{j\in[n]\setminus\pi^{-1}(\mathcal{C})} \mathbb{E}\Big[V(\mathcal{C} \cup \{\pi(j)\}, \mathcal{W} \cup \{W_{\pi(j)}\})\Big]$, is the expected consumer welfare if she continues her search, where the expectation in this term is w.r.t. f_W. When the consumer continues her search, she needs to decide about the next position that she wants to screen. The maximization in the second term captures this decision. In Section 4, we present the consumer's optimal policy, i.e., the solution to the aforementioned DP. Having described our search model, next we provide a discussion on difference between our model and other search models.

3.1.1. Comparison with Other Models As mentioned above, our search model is motivated by the two-stage process that the consumer undergoes when searching online. A single-stage search model, such as the seminal work of Weitzman (1979), assumes that the consumer learns *both* the weight and the idiosyncratic utility of a product in a single stage of the search. Such a model is used by Ursu (2016) and Chu et al. (2017). In the single-stage search model of Ursu (2016) and Chu et al. (2017), products placed in lower positions do not impact the choice probability of higher-positioned products unless the consumer searches all of the products. Said another way, the substitution effects between lower-positioned products and higher-positioned ones are not captured in these models. In practice, such models would translate to the following process: on an online platform, a consumer fully examines the first product and observes her utility from that product; then, she either stops her search or moves on to the second product and fully examines it. This process differs from the consumer search behavior (first eyeballing and then carefully evaluating a subset) that we propose in our work.

In modeling the consumer search as a two-stage process, we follow the *consider-then-choose* modeling paradigms that have been widely proposed and studied in Marketing and Operations literature. In consider-then-choose modeling, while in some papers (e.g., Aouad et al. (2015), Jagabathula and Rusmevichientong (2016), Aouad et al. (2019)) the consideration set is exogenously sampled, in some others (e.g., Roberts and Lattin (1991), Hauser (2014), Wang and Sahin (2017)), to form a consideration set, the consumer accounts for the evaluation cost of products. In the latter set of papers, the consumer evaluates a product in the second stage after she has already formed her consideration set. However, she accounts for this evaluation cost upfront when she includes the product in her consideration set. We use the same process in the second stage of our search model. The main factor differentiating our model from the existing consider-then-choose models pertains to the first stage of the search. In our work, the consideration set is formed endogenously through a sequential screening process in which the consumer needs to pay a position-dependent screening cost to take a first glance at a product (to observe its preference weight). We remark that on online platforms most products are not initially visible to the consumer. As such, she needs to exert effort to take the first glance. We explicitly account for this screening cost.

3.2. Platform's Information and Objectives

We start by describing the platform's knowledge regarding products and consumer types. We then proceed to present the platform's objectives. The platform knows the distribution of the weights, f_W, as well as the (realized) weights of the n products to be ranked. Access to a large amount of historical data and computational power makes it reasonable to assume that well-established online platforms have such perfect knowledge. Note that there is an information asymmetry between the platform and consumers: unlike the platform, the consumers have no information about the product weights (except for knowing the distribution f_W). Rather, they need to incur search costs in order to observe the weights for a subset of products. In regard to consumers' type and search costs, the platform knows the type distribution $(\theta^1, \theta^2, \ldots, \theta^K)$. However, the platform does not observe the type of consumers. Further, we assume that the platform knows the search costs s_j^k, $j \in [n]$, $k \in [K]$. (Later, in Sections 5 and 6, we analyze the w-ordered ranking that does not rely on the platform's knowledge of the type distribution and search costs.)

We focus on a platform-centric objective as well as a consumer-centric one. In particular, our first objective is maximizing the platform's market share, whereas our second one is maximizing consumer welfare. We study the platform's ranking problems for each objective separately.

Market Share: Let $\mathcal{C}_\pi^k$ be the consideration set of consumers of type k under ranking π when the consumer follows her optimal search policy. (In Section 4, we describe how $\mathcal{C}_\pi^k$ is formed under the optimal search policy.) The platform's market share of consumers of type k is equivalent to the fraction of such consumers who make a purchas. Under our assumption that the idiosyncratic part of utility is drawn from the standard Gumbel distribution, the probability that a consumer of type k makes a purchase from the platform is given by

$$\frac{\sum_{i\in\mathcal{C}_\pi^k} w_i}{1+\sum_{i\in\mathcal{C}_\pi^k} w_i}\,. \tag{3}$$

With a slight abuse of notation, for any set $\mathcal{C}$, we denote $\sum_{i\in\mathcal{C}} w_i$ with $w(\mathcal{C})$. Then, the probability of purchase of a consumer of type k under ranking π is $\frac{w(\mathcal{C}_\pi^k)}{1+w(\mathcal{C}_\pi^k)}$. Note that under our model, Z_i's, $i\in\mathcal{C}_\pi^k\cup\{0\}$, are i.i.d. samples with the standard Gumbel distribution. It is well known that $\mathbb{P}\left(\arg\max_{i\in\mathcal{C}_\pi^k\cup\{0\}}\{\log(w_i)+Z_i\}\neq 0\right)=\frac{w(\mathcal{C}_\pi^k)}{1+w(\mathcal{C}_\pi^k)}$; see Mahajan and Van Ryzin (2001) and Talluri and Van Ryzin (2004). Observe that the purchase probability is a function of ranking π as the ranking influences the consideration set. Further, since the platform has the full knowledge of w_i's, it can compute the consideration set $\mathcal{C}_\pi^k$ and its resulting probability of purchase. Hence, the platform that wishes to maximize its market share needs to solve the following optimization problem:

$$\mathsf{Ms} \;=\; \max_{\pi\in\Pi}\sum_{k\in[K]}\theta^k\frac{w(\mathcal{C}_\pi^k)}{1+w(\mathcal{C}_\pi^k)}\,, \tag{Ms}$$

where Π is the set of all possible $n!$ permutations. With a slight abuse of notation, we also use (Ms) as our shorthand for the market share maximization problem. To solve this optimization problem, one needs to understand how the offered ranking π impacts different types of consumers. Section 5 is dedicated to studying the above optimization problem.

Consumer Welfare: Here, the platform that aims at maximizing the consumer's welfare solves the following optimization problem:

$$\mathsf{Wel}=\max_{\pi\in\Pi}\sum_{k\in[K]}\theta^k\;\mathsf{Wel}_\pi^k(\mathcal{C}_\pi^k,\;\{w_i\}_{i\in\mathcal{C}_\pi^k})\,, \tag{Wel}$$

where $\mathsf{Wel}_\pi^k(\cdot,\cdot)$ is defined in Equation (1). Note that the welfare maximization problem faced by the platform is different from that faced by the consumer. This difference arises from their information asymmetry w.r.t. weights of the n products. Given a ranking π and not knowing the weights of products, the consumer optimizes her search policy to maximize her expected welfare, in which she balances the search cost for each position with the expected increase in her utility if she adds another product to her consideration set. Meanwhile, the platform optimizes ranking knowing the weights of all the products and the consumers' consideration set given a ranking. We further highlight that in the above problem, the platform takes expectation w.r.t. consumer type distribution, $(\theta^1,\theta^2,\ldots,\theta^K)$. To avoid any confusion about the welfare problem from the platform's vs. consumer's perspectives, we refer to the above objective as consumer welfare. However, on the consumer side, we explicitly mention that she maximizes her expected welfare.

4. Consumers' Optimal Search Policy

In this section, we present the optimal search policy of a consumer whose objective is to maximize her expected welfare. In particular, we provide the optimal solution of the DP in Equation (2). The state of the DP—which would entail the subset of screened positions—would grow exponentially in the number of products n. Thus, at a first glance, solving for the optimal policy seems to be intractable. However, in the following, we first show that the state of the above DP collapses to two dimensions. In particular, in the optimal search policy, the consumer only needs to consider $w(\mathcal{C}) = \sum_{i\in\mathcal{C}} w_i$ and $\sum_{i\in\mathcal{C}} s^k_{\pi^{-1}(i)}$; see Lemma 1. Further, in Theorem 1 (which is the main result of this section), we will show that the optimal sequential search policy has an intuitive index-based structure.

We start our analysis of the DP, presented in (2), by utilizing the properties of Gumbel distribution. In particular, in the following lemma, we show that the expected utility of the consumer from screening set $\mathcal{C}$—with its corresponding set of observed weights $\mathcal{W}$—admits a simple form.

LEMMA 1 **(Consumer's Expected Utility – Section 3.5 of Train (2009))**. *The expected utility of a consumer who only screens products in set $\mathcal{C}$ is given by*

$$\mathbb{E}\Big[\max_{i\in\mathcal{C}\cup\{0\}} \{\log(w_i) + Z_i\}\Big] = \log\big(1 + w(\mathcal{C})\big) + \gamma, \tag{4}$$

where $w(\mathcal{C}) = \sum_{i\in\mathcal{C}} w_i$ and γ is the Euler-Mascheroni constant.

Lemma 1 is shown in Section 3.5 of Train (2009). Nevertheless, for convenience, we present its proof in Appendix EC.3. This lemma implies that in order to compute $\mathsf{Wel}^k_\pi(\mathcal{C},\mathcal{W})$, it suffices to only know $w(C)$ and $\sum_{i\in\mathcal{C}} s^k_{\pi^{-1}(i)}$. Utilizing this result, in the following, we first present a simple index-based policy. Then, in Theorem 1, we prove that this index-based policy is indeed optimal.

Our policy can be described by a set of indices for positions that we call *reservation prices*, r^k_j's, $k \in [K]$ and $j \in [n]$, where r^k_j solves the following equation:

$$\mathbb{E}\left[\log\left(1 + r^k_j + W\right)\right] - \log\left(1 + r^k_j\right) = s^k_j\,. \tag{5}$$

In Lemma EC.1 in Appendix EC.2, we show that the above equation admits a unique solution. Here, the expectation is w.r.t. the random variable $W \sim f_W$. In order to gain insight, for a moment, suppose that $n = 2$ and that the consumer has already screened the first position and observed $w_{\pi(1)}$. Now she is deciding to whether to screen the second position or not. If she screens, then her expected welfare would be $-s^k_2 + \mathbb{E}\left[\log\left(1 + w_{\pi(1)} + W\right)\right] + \gamma$. In contrast, if she does not screen, her expected welfare is simply $\log\left(1 + w_{\pi(1)}\right) + \gamma$. Now, by definition of reservation prices given in (5), if $w_{\pi(1)} = r^k_2$, then the consumer is indifferent between these two options. If $w_{\pi(1)} > r^k_2$, then the consumer will not screen the second position, whereas if $w_{\pi(1)} < r_2^k$, then she will. Generalizing this insight, roughly speaking, the reservation price r_j^k is the deterministic reward that makes the consumer indifferent between (i) screening position j and (ii) leaving it unscreened. We note that the reservation price r_j^k only depends on the search cost of position j and the distribution of preference weights f_W.

We remark that $r_j^k \geq r_{j+1}^k$ for any $j \in [n-1]$ and $k \in [K]$. This follows from our assumption that search costs are increasing with position (i.e., $s_j^k \leq s_{j+1}^k$, $j \in [n-1]$ and $k \in [K]$) and the observation that function $\log(1+x)$ is increasing. Further, we point out that $r_j^k < 0$ if $\mathbb{E}[\log(1+W)] < s_j^k$. Under the index-based policy, the consumer of type $k \in [K]$ starts screening positions downward—which corresponds to the decreasing order of their reservation prices—and she stops her search at the first position j that satisfies the following condition:

$$\sum_{j' \in [j]} w_{\pi(j')} \ \geq \ r_{j+1}^k \,.$$

In that case, under our index-based search policy, the consideration set of the consumer of type k, $\mathcal{C}_\pi^k$, consists of the first j products; that is, the consideration set is $\mathcal{C}_\pi^k = \{\pi(1), \pi(2) \ldots, \pi(j)\}$. We stress that if $r_1^k \leq 0$, then the consideration set is $\emptyset$. The formation of the consideration set $\mathcal{C}_\pi^k$ is summarized below.

Forming Consideration Set $\mathcal{C}_\pi^k$ under Index-Based Search Policy

IF $r_1^k \leq 0$, $\mathcal{C}_\pi^k = \emptyset$. ELSE,

Initialize $\mathcal{C}_\pi^k = \emptyset$. Then, examine the positions in increasing order of their indices, and stop in position j if

$$\sum_{j' \in [j]} w_{\pi(j')} \ \geq \ r_{j+1}^k, \tag{6}$$

otherwise continue and add product $\pi(j+1)$ to the consideration set.

The following theorem shows that the described index-based policy is optimal, i.e., it solves the consumer's sequential search problem.

Theorem 1 **(Optimality of Index-Based Policy)**. *For any consumer type $k \in [K]$, among the set of non-anticipatory policies, Ω, the index-based policy defined above is optimal, i.e., it solves $V(\emptyset, \emptyset)$, where $V(\cdot,\cdot)$ is defined in* (2).

The proof of Theorem 1 is presented in Appendix EC.4. We now highlight some features and structural properties of the optimal search policy and its resulting consideration set:

(i) Under the optimal search policy, consumers screen downward, starting from the top positions, and thus, their consideration set consists of products in top positions. Such consumer behavior has been verified empirically (Ngwe 2017, Ngwe and Teixeira 2017) and is considered in some recent theoretical work (Aouad and Segev 2015, Chu et al. 2017). However, we emphasize that we do not enforce downward search as an assumption; rather, we prove that such an order is optimal when search costs increase with position index.

(ii) The consideration set is endogenously determined by the ranking; that is, the order of the products changes the consideration set. Note that in Aouad and Segev (2015) and Gallego et al. (2016), the size of the consideration set of consumers is exogenously drawn from a distribution.

(iii) Products with a high weight can impose significant externality on products that are placed in lower positions. This holds because the consumer stops her search whenever $\sum_{j'\in[j]} w_{\pi(j')}$ exceeds the reservation price r^k_{j+1}. One can think of $\sum_{j'\in[j]} w_{\pi(j')}$ as the externality that products in positions $1, 2, \ldots, j$ impose on the products in positions $j+1, j+2, \ldots, n$.

Further, we remark that even though the sequential search problem that the consumer faces bears similarity to the setting introduced in the seminal work of Weitzman (1979) (also known as the Pandora's box problem), there is a crucial difference that prevents us from directly applying the result of Weitzman (1979). In our setting, after screening a position (which can be thought of as a box in the Pandora's box problem), the consumer does not observe her entire utility; rather, she only learns the weight of the product in that position. Consequently, as evident in (1), the consumer welfare after screening a subset of product $\mathcal{C}$ depends on the weight of *all* of the screened products, and not just the product with the maximum weight. Conversely, in the Pandora's box problem, the consumer observes her entire utility for a product after screening, and thus, she only needs to keep the product with the maximum weight. Due to this departure, the proof of Theorem 1 does not follow from Weitzman (1979).

5. Optimal Ranking for Maximizing Market Share

In this section, we study the ranking problem of a platform concerned with maximizing its market share as defined in (Ms). We first show that Problem (Ms) is NP-complete. Then, in Section 5.1, we present an algorithm that ranks the products in decreasing order of their preference weights. We refer to this algorithm as w-ordered. As stated earlier, such an algorithm is practically appealing because it does not require any knowledge of the type distribution or the search costs. We show that this simple algorithm obtains a multiplicative approximation factor of $\frac{1}{2}$ and an additive approximation factor of 0.1716, where both approximation factors are tight. Finally, in Section 5.2, under the assumption of $\rho = O(1)$, we present a novel PTAS for the market share maximization problem whose running time scales with K^2. Recall that K is the number of consumer types.

We start our study of the market share maximization problem by establishing that the ranking problem that the platform faces, i.e., Problem (Ms), is NP-complete.

THEOREM 2 **(NP-Completeness)**. *Problem (*Ms*) is NP-complete.*

The proof of Theorem 2 is provided in Appendix EC.6.

5.1. Performance of W-Ordered Algorithm

Having established the NP-completeness of Problem (Ms), in this section, we analyze the performance of a natural algorithm, which we call w-ordered. This algorithm ranks the products in decreasing order of their preference weights. We remark that such an algorithm is attractive from a practical implementation perspective because it is very simple and because it only relies on the platform's knowledge of the product weights. In other words, in order to implement this algorithm, the platform does not need to estimate the type distribution, or the search costs for different positions/types. In the following theorem, we show that this algorithm also enjoys a strong performance guarantee.

THEOREM 3 **(Performance of W-Ordered Algorithm)**. *Consider the w-ordered algorithm that ranks the products in decreasing order of their weights. Then, the market share of the w-ordered algorithm, denoted by* Ms_w*, satisfies the following equations:*

$$\mathsf{Ms}_w \ \geq \ \max\left\{\frac{\mathsf{Ms}}{2}, \mathsf{Ms} - \frac{\sqrt{2}}{(\sqrt{2}+2)(\sqrt{2}+1)}\right\} \ \approx \ \max\left\{\frac{\mathsf{Ms}}{2}, \mathsf{Ms} - 0.1716\right\},$$

where Ms *is the maximum market share.*

In Appendix EC.12, we provide an example that shows that our analysis in Theorem 3 is tight for both the multiplicative and the additive factors.

5.2. PTAS for Market Share Maximization Problem

As mentioned above, even though the w-ordered ranking has desirable practical properties, it suffers from the inefficiency caused by the externality of high-weight products when placed in top positions. Consequently, as discussed in the proof of Theorem 3, each consumer type $k \in [K]$ forms the smallest consideration set when faced with a w-ordered ranking compared to any other ranking. To improve upon the performance guarantee of the w-ordered ranking, an algorithm needs to strike a balance between (1) inducing screening (by filling top positions with "moderate"-weight products) and (2) including products with high weights in the top positions, which may shorten the screening process due to the externality effect. The algorithm also needs to resolve another trade-off that arises when faced with multiple types of consumers: to keep the types with lower search costs engaged in further screening, the platform may forfeit the opportunity to display the highest-weight products to all types. As Theorem 2 established, Problem (Ms) is NP-complete. However, in the following theorem, we develop a novel DP-based algorithm whose market share is at least $(1-\epsilon)\mathsf{Ms}$ for any $\epsilon \in (\log(\rho)/n, 1)$. We show that the running time of our algorithm is $O(K^2 n^{\log(\rho)/\epsilon+1})$. [4] We point out that because $\mathsf{Ms} < 1$, a multiplicative approximation factor of $(1-\epsilon)$ also implies an additive approximation factor of ϵ.

THEOREM 4 **(PTAS for Market Share Maximization Problem)**. *Assume that* $\rho = O(1)$. *Then, Problem (*Ms*) admits a PTAS.*

We note that a PTAS is an algorithm that takes an instance of an optimization problem, i.e., the weight vector $(w_1, w_2, \ldots, w_n)$ in our case, and a parameter $\epsilon > 0$. The algorithm then produces a solution within a factor $1-\epsilon$ of the optimum solution in polynomial time. For a constant ϵ, the running time of a PTAS is polynomial in input size, n. We also highlight that the running time of our PTAS is quadratic in the number of consumer types, K, which makes it practically appealing, especially when K is large.

We now proceed to present our PTAS, which has two main steps. The first step involves bucketing products based on a rounding procedure. Specifically, in the first step, products' weights will be discretized (bucketed) using a geometric grid and rounding up. The second step requires reformulating the market share maximization problem as a dynamic program with a state space whose size remains polynomial in n and linear in K and with an optimal value that is within a factor $(1-\epsilon)$ of the optimum value of Problem (Ms).

Rounding Weights: Let $\epsilon \in (0,1)$, $\underline{w} = \min_{i\in[n]} w_i$, and $\bar{w} = \max_{i\in[n]} w_i$ where $\underline{w} \geq w_{\min}$ and $\bar{w} \leq w_{\max}$. We bucket the products via a rounding procedure parameterized by ϵ. In particular, we add product i with weight w_i to bucket $b \in [B]$ if

$$\underline{w}(1+\epsilon)^{b-1} \leq w_i < \underline{w}(1+\epsilon)^b \,.$$

Here, B is the total number of buckets, which is less than or equal to $\frac{\log(\rho)}{\log(1+\epsilon)} + 1 = O(\frac{\log(\rho)}{\epsilon})$, where $\rho = \frac{w_{\max}}{w_{\min}} \geq \frac{\bar{w}}{\underline{w}}$. Then, for any product i that falls into bucket b, we define $\tilde{w}_i = \underline{w}(1+\epsilon)^b > w_i$; that is, in our rounding procedure, all the weights are rounded up. Later in this section, we will explain why rounding up is crucial in the design of our PTAS. Further, we define $\mathsf{Rw}_b = \underline{w}(1+\epsilon)^b$ as the rounded weights of all the products that fall in bucket b.

We now explain how these buckets (rounded weights) are used in formulating our DP along with a subtlety that requires us to define an ordering of products within each bucket. We start by

[4] Note that when $\epsilon < \log(\rho)/n$, the number of products n is constant, and as a result, the optimal ranking can be computed by evaluating all $O(2^n)$ ranking candidates. This is so because $\rho = O(1)$.

revisiting our objective function, i.e., the platform's market share. For a ranking π and type k, the market share of that type is given by

$$\frac{w(\mathcal{C}_\pi^k)}{1+w(\mathcal{C}_\pi^k)} .$$

Note that for $\delta \geq 1$ and $x \geq 0$, $\frac{\delta x}{1+\delta x} \leq \delta \frac{x}{1+x}$. Thus, if instead of w_i, we use $\tilde{w}_i$ (i.e., its rounded version), we can control for the approximation error in the above ratio. Specifically, with rounding up, we have $\tilde{w}_i \leq (1+\epsilon)w_i$, which in turn implies that

$$\frac{\tilde{w}(\mathcal{C}_\pi^k)}{1+\tilde{w}(\mathcal{C}_\pi^k)} \leq (1+\epsilon)\frac{w(\mathcal{C}_\pi^k)}{1+w(\mathcal{C}_\pi^k)} ,$$

where $\tilde{w}(\mathcal{C}_\pi^k) = \sum_{i\in\mathcal{C}_\pi^k} \tilde{w}_i$. However, the weight of product i also impacts the formation of the consideration set $\mathcal{C}_\pi^k$, as dictated by the stopping rule of (6); that is, rounding weights may result in the formation of a different consideration set. For example, suppose that $n=2$, $K=1$, and $\pi=(1,2)$, i.e., in ranking π, products 1 and 2 are placed in positions 1 and 2, respectively. Further, suppose that $w_1 < r_2 \leq \tilde{w}_1$. With the actual weights, the consumer screens w_1 and w_2, and thus, her consideration set becomes $\{1,2\}$. However, with the rounded weights, $\tilde{w}_1$ and $\tilde{w}_2$, she stops after screening product 1.

In order to overcome the above challenge—presented by the complex impact that rounding weights can have on the consideration set of each type—we use the following idea: we limit the search space of our algorithm to the "*low-weight priority*" ranking class defined as follows:

Definition 1 (Low-weight Priority (LWP) Rankings). A ranking π belongs to the LWP class if there does not exist a pair of products (i_1,i_2) with $w_{i_1} < w_{i_2}$ that belong to the same bucket $b\in B$ and $\pi^{-1}(i_1) > \pi^{-1}(i_2)$, that is, under π, product i_1 should be placed in a better position, i.e., a position with a lower search cost, than product i_2.

As we will explain later, in our DP, we sequentially decide how to fill the positions from the top position moving downward. Within the class of LWP rankings, if we just know the number of products from each bucket that are placed in the first $j-1$ positions and we decide to place a product from a bucket b in position j, we know the *index of* the product (and not only its bucket index). Let us clarify this with an example. Suppose that we have 6 products with weights $w_1 < w_2 < \ldots < w_6$. Assume that we have two buckets, where products $1,2,3$ belong to bucket 1 and products $4,5,6$ belong to bucket 2. Now suppose that we have filled the first three positions with two products from bucket 2 and one product from bucket 1. If we decide to place a product from bucket 1 in the fourth position, then we know that product is product 2—with the original weight of w_2—which has the second smallest weight among the products in bucket 1. Knowing the index of the product to be placed in each position j (fixing the decisions in the first $j-1$ positions) enables us to reconstruct the consideration set of each type using the actual weights.

Restricting the search space to the class of LWP rankings, however, raises the following question: what if the optimal ranking, i.e., the one that solves Problem (Ms), does not belong to this class? In the proof of Theorem 4 below, we overcome this issue by constructing an auxiliary ranking that belongs to this class. We then show that the consideration set formed under the auxiliary ranking is larger than that under optimal ranking. Thus, the market share of the auxiliary ranking serves as an upper bound on that of the optimal ranking when the rounded weights are used in computing market share. Here, we crucially use the fact that the market share increases as the consumers screen more products.

DP-Based Algorithm: In the following, we formulate a finite-horizon dynamic program. Our program has $n+1$ steps. In each step j, we make the following decision: from which bucket b we should pick a product to fill position j. We highlight that once we decide to take a product from bucket b, we will choose the product with the smallest actual weight among the remaining products in bucket b. This ensures that the resultant ranking will belong to the LWP class. To this end, we sort the products in each bucket $b \in [B]$ in increasing order of their weights. Then, with a slight abuse of notation, we define $w_b^{(i)}$ as the i^{th} smallest weight in bucket b.

Suppose that we are at step j, i.e., we have already decided about the first $j-1$ positions. We denote the state by $\mathcal{S} = (\mathcal{B}, \kappa)$, where $\mathcal{B} = (m_1, m_2, \ldots, m_B)$ represent the status of buckets. Recall that $B = O(\frac{\log(\rho)}{\epsilon})$ is the number of buckets. Precisely, for bucket $b \in [B]$, $m_b \geq 0$ is the number of products from this bucket that are already placed in the first $j-1$ positions. (Note that for $j=1$, $m_b = 0$ for $b \in [B]$.) Thus, $\sum_{b\in[B]} m_b = j-1$. Given $\mathcal{B}$, we define the following two summations that prove useful in the description of our DP:

$$\text{weight}(\mathcal{B}) = \sum_{b\in[B]} \sum_{i=1}^{m_b} w_b^{(i)} \quad \text{and} \quad \text{Rweight}(\mathcal{B}) = \sum_{b\in[B]} m_b \cdot \mathsf{Rw}_b \,. \tag{7}$$

We further define κ as the smallest consumer type that has not stopped yet after screening the first $j-1$ positions. Recall that if type k does not stop at a given position, any type $k' > k$ will not stop either. This is so because we assume $s_j^k \geq s_j^{k+1}$ for $j \in [n]$ and $k \in [K-1]$.

As mentioned before, at each step j, our goal is to choose a bucket b for position j. First note that the chosen bucket b should not be empty; that is, it should have at least one product left. Thus, at step j, we search over all non-empty buckets b. Let us now describe how the state of our DP transitions if we pick bucket b. We denote the new state by $\mathcal{S}' = (\mathcal{B}', \kappa')$. Given the initial state $\mathcal{S}$, if we select bucket b, then we have

$$\mathcal{B}' = (\mathcal{B}_{-b}, m_b + 1)\,. \tag{8}$$

Here, $\mathcal{B}_{-b}$ represents the status of all the buckets except bucket b. In words, the status of all buckets except bucket b remains the same. For bucket b, the number of products that have been taken from it increases by one.

Next, we explain how κ' is calculated. Recall that κ' represents the smallest consumer type that has not stopped yet after screening the first j positions, where position j is filled with a product from bucket $b \in [B]$. By construction, the product in position j will be the product with the (m_b+1)-st smallest weight in bucket b. Further, the summation of the weights in the first $j-1$ positions is given by $\mathrm{weight}(\mathcal{B})$ (as defined in (7)). Thus, we have

$$\kappa' = \min\left\{k \in [\kappa, K] : w_b^{(m_b+1)} + \mathrm{weight}(\mathcal{B}) < r_{j+1}^k\right\}. \tag{9}$$

Note that $\kappa' \geq \kappa$, and any consumer with types $\kappa, \kappa+1, \ldots, \kappa'-1$ stop screening at position j. Note that the actual weight of the products (not the rounded version) is used in determining κ'.

Having described the state, action, and state transition at step j, we finally define the per-step reward at step j. The reward consists of the market share of consumer types that stop screening exactly at position j, which is given by

$$\frac{\mathrm{Rweight}(\mathcal{B}) + \mathrm{Rw}_b}{1+\mathrm{Rweight}(\mathcal{B}) + \mathrm{Rw}_b} \cdot \sum_{k=\kappa}^{\kappa'-1} \theta^k\,,$$

where $\mathrm{Rweight}(\mathcal{B})$ is defined in (7). Having specified all the components of our DP, we are now ready to present it. For any $j \in \{1, \ldots, n+1\}$ and any state $\mathcal{S} = (\mathcal{B}, \kappa)$, we define $W_j(\mathcal{S})$ as the platform's maximum market share from the consumer types $\kappa, \ldots K$ by following an optimal policy for positions $j, \ldots, n$. Thus, $W_j(\mathcal{S})$ solves the following recursive equation:

$$W_j(\mathcal{S}) = \max_{\text{non-empty } b\in[B]} \left\{W_{j+1}(\mathcal{S}') + \frac{\mathrm{Rweight}(\mathcal{B}) + \mathrm{Rw}_b}{1+\mathrm{Rweight}(\mathcal{B}) + \mathrm{Rw}_b} \cdot \sum_{k=\kappa}^{\kappa'-1} \theta^k\right\}. \tag{10}$$

We emphasize that, as mentioned at the beginning of the description of the DP, in order to obtain a ranking that belongs to the LWP class, at any step j, if we decide to pick a product from bucket b to fill position j, we will select a remaining product with the smallest weight.

We are now ready to present our PTAS.

ALGORITHM 1: A PTAS Algorithm for the Market Share Maximization Problem in (Ms)

Initialization: Set $\mathcal{S}=(\mathcal{B},1)$, where $\mathcal{B}=(0,0,\ldots,0)$; that is, $m_b=0$, $b\in[B]$.

FOR $j=1,2,\ldots,n$,

—**Choose the right bucket:** Let bucket b^* be given as

$$\mathsf{b}^* = \underset{\text{non-empty } b\in[B]}{\arg\max} \left\{ W_{j+1}(\mathcal{S}') + \frac{\text{Rweight}(\mathcal{B})+\text{Rw}_b}{1+\text{Rweight}(\mathcal{B})+\text{Rw}_b}\cdot\sum_{k=\kappa}^{\kappa'-1}\theta^k \right\},$$

where $\mathcal{S}'=(\mathcal{B}',\kappa')$; $\mathcal{B}'$ and κ' are defined in (8) and (9), respectively.

—**Choose a product from bucket b^*:** Select the product from bucket b^*, defined above, with the smallest weight, i.e., $w_{\mathsf{b}^*}^{(m_{\mathsf{b}^*}+1)}$, and place it in position j.

—**Update the state:** Update $\mathcal{B}$ and κ as follows:

$$\mathcal{B}\leftarrow(\mathcal{B}_{-b^*},m_{b^*}+1)$$
$$\kappa\leftarrow\min\left\{k\in[\kappa,K]: w_{\mathsf{b}^*}^{(m_{\mathsf{b}^*}+1)}+\text{weight}(\mathcal{B})<r_{j+1}^k\right\}.$$

Note that Algorithm 1 would have returned the optimal solution to Problem (Ms) if we had not bucketed the weights. Thus, to bound the performance of Algorithm 1, we need to characterize the impact of rounding on the solution returned by the algorithm as well as the impact of restricting the search space to the class of LWP rankings. The following claim shows that Algorithm 1 returns a solution within a $(1-\epsilon)$ fraction of the optimum solution. Further, it shows that for a fixed ϵ, the running time of Algorithm 1 is polynomial in n. Thus, Claim 1 proves Theorem 4.

Claim 1 **(Proof of Theorem 4)**. *For $\epsilon\in(\log(\rho)/n,1)$, Algorithm 1 achieves a market share of at least $(1-\epsilon)\mathsf{Ms}$. Further, the running time of Algorithm 1 is $O(K^2 n^{\log(\rho)/\epsilon+1})$.*

The proof of Claim 1 is provided in Appendix EC.8.

6. Optimal Ranking for Maximizing Consumer Welfare

In this section, we turn our attention to the ranking problem of a platform concerned with maximizing consumer welfare, as defined in (Wel). As discussed in Section 3.2, to characterize the welfare-maximizing ranking, i.e., to solve Problem (Wel), the platform takes into account the optimal search policy of the consumers given a ranking. The objective of Problem (Wel) bears some similarity to that of (Ms). In particular, for each consumer type k, both objectives are increasing and concave in the summation of the weights of the products included in the consideration set, i.e., $w(\mathcal{C}_\pi^k)$. However, there is a crucial difference between these objectives. Unlike market share, consumer welfare may decrease as consumers screen more products. This is so because screening products is costly for the consumer. Thus, to maximize consumer welfare, the platform needs to be mindful of the search costs while trying to increase the summation of the weights of the products that a type screens. Further, due to this difference, most of the analysis in the previous section is not directly applicable here.

In the following, we first show that Problem (Wel) is NP-complete. Then, in Section 6.1, we show that the w-ordered algorithm enjoys a strong theoretical guarantee. Finally, in Section 6.2, we present an additive PTAS for the welfare maximization problem.

PROPOSITION 1 **(NP-Completeness)**. *Problem (*Wel*) is NP-complete.*

The proof of Proposition 1, in spirit, is similar to that of Theorem 2, as it is based on reducing an instance of the uniform knapsack problem to our consumer welfare maximization problem. The proof is deferred to Appendix EC.9.

6.1. Performance of W-Ordered Algorithm

Having proved the NP-completeness of Problem (Wel), in this section, we analyze the performance of the w-ordered algorithm, which ranks products in decreasing order of their weights. As discussed in Section 5, such a greedy algorithm is practically appealing, as it only relies on knowledge of products' weights. In the following proposition, we show that this algorithm also has a strong performance guarantee when the performance is measured in terms of consumer welfare.

PROPOSITION 2 **(Performance of W-Ordered Algorithm)**. *Consider the w-ordered algorithm that ranks the products in decreasing order of their weights. Then, the consumer welfare of the w-ordered algorithm, denoted by* Wel_w*, satisfies the following equations:*

$$\mathsf{Wel}_w \;\geq\; \mathsf{Wel} - \log(2) \;\approx\; \mathsf{Wel} - 0.6931\,,$$

where Wel *is the maximum consumer welfare.*[5]

The proof of Proposition 2 uses similar ideas to that of Theorem 3, and it is deferred to Appendix EC.10. In Appendix EC.13, we present an example that shows our analysis is tight.

6.2. PTAS for Consumer Welfare Maximization Problem

As discussed in Section 5, across all rankings, each consumer type $k \in [K]$ forms the smallest consideration set when faced with the w-ordered ranking. This is due to the externality of high-weight products when placed in top positions. However, because of the idiosyncratic part of the utility, the consumer may benefit from screening more products even though such a screening is costly. In order to achieve a better performance guarantee than that of the w-ordered ranking, an algorithm needs to rank the products in a way that the resulting consideration set is not too small

[5] Throughout the paper, we present the natural logarithm by $\log(\cdot)$.

(which happens under w-ordered ranking); at the same time, the summation of the weights of the screened products is large enough. Our next algorithm achieves these goals.

In the following, we present an additive PTAS that aims to address the challenges mentioned above; that is, given $\epsilon \in (\frac{\rho}{n}, 1)$, our algorithm returns a solution in polynomial time whose welfare is at least $\mathsf{Wel} - \epsilon$. (Recall that $\rho = \frac{w_{\max}}{w_{\min}}$.) Observe that an additive PTAS is stronger than a multiplicative one. Our algorithm constructs a class of rankings as follows. Let $M := \frac{\rho}{\epsilon}$, and consider any ordered set $S \subseteq [n]$ with size M. The first M positions are filled with an ordered set S with size M. Then, the rest of the positions are filled with the remaining products in decreasing order of their weights. For each ranking in this class, our algorithm computes the consumer welfare and then returns the ranking with the maximum welfare. By considering all possibilities for the first M positions, we ensure that less patient types achieve optimal consumer welfare (more precisely, those with a consideration set of a size of at most M in the optimal ranking). For the more patient types that screen beyond position M (in the optimal ranking), we ensure that their welfare loss is insignificant. As becomes more clear in the proof, this follows from filling out the low positions greedily in decreasing order of product weights.

Theorem 5 **(PTAS for Consumer Welfare Maximization Problem)**. *Assume that $\rho = O(1)$. Then, Problem (*Wel*) admits an additive PTAS.*

We now present our PTAS for the welfare maximization problem.

ALGORITHM 2: A PTAS Algorithm for the Consumer Welfare Maximization Problem in (Wel)

Let $M = \frac{\rho}{\epsilon}$, where $\rho = \frac{w_{\max}}{w_{\min}}$.

- Consider any ordered set $S \subseteq [n]$ with size M. For any such S, construct a ranking as follows.

  —**First M positions:** For any $i \in [M]$, place the i^{th} element of the ordered set S in position i.

  —**Next $n - M$ positions:** For any $i \in [n-M]$, place the product that has the i^{th} highest weight among the remaining products, i.e., $[n] \backslash S$, in position $M + i$.

Return the best constructed ranking, i.e., return the one with the highest consumer welfare.

The following claim, which proves Theorem 5, shows that Algorithm 2 returns a solution within an additive factor of ϵ of the optimum solution, while having a running time that is polynomial in n for a fixed ϵ. We highlight that the running time of Algorithm 2 grows linearly with the number of types K. (The claim is proven in Appendix EC.11.)

Claim 2 **(Proof of Theorem 5)**. *For $\epsilon \in (\frac{\rho}{n}, 1)$, Algorithm 2 achieves consumer welfare of at least $\mathsf{Wel} - \epsilon$. Further, the running time of Algorithm 2 is $O(Kn^{1+\rho/\epsilon})$.*

We finish this section by briefly discussing why a DP-based PTAS similar to Algorithm 1 may not work for the objective of maximizing consumer welfare. At a high level, Algorithm 1 may not work here because it uses the fact that the market share increases as the size of the consideration sets grows, which is no longer the case for consumer welfare. Recall that a crucial step in the analysis of Algorithm 1 is to compare the objective of π_{opt} with that of the DP's optimal ranking. In order to do so, in the proof of Claim 1, we constructed an auxiliary ranking denoted by π_{aux} that has the same bucket representation as ranking π_{opt} but, unlike ranking π_{opt}, belongs to the LWP class. For the objective of market share, it was possible to show that market share under π_{aux} (using the rounded weights $\tilde{w}_i$) is larger than or equal to that under π_{opt} (see Inequality (EC.9) and the discussion preceding it). For the objective of consumer welfare, however, such a relationship does not exist. This is because under π_{aux}, the consumer screens more products, and thus, she will pay an extra search cost.

7. Numerical Studies

In this section, we complement our theoretical developments with numerical studies using synthetic data. We start with developing heuristic versions of our algorithms that do not rely on model primitives, namely, the type probabilities and reservation prices. We then numerically evaluate the performance of our heuristics as well as our PTAS (Algorithm 1) in two informational settings. In the first setting, the model parameters are known, whereas in the second setting, we use the estimated versions of the model parameters.

Throughout this section, we assume that the platform can observe the last position the consumer screened. (Recall that in designing our algorithms, making such an assumption is not necessary.) This assumption is motivated by the fact that many platforms display their products in small devices where the consumers need to swipe in order to see the next product. In such small devices, the last screened product is observable by platforms. Under this assumption, the platform can observe consumers' consideration set. Once the consideration set is formed—which is observable to the platform—the consumer choice model reduces to a standard MNL model. This implies that estimating the preference weights of products follows the standard procedure in the literature. Given the above, in both informational settings considered in this section, we abstract away from estimating the preference weights and assume that the platform has perfect knowledge of the products' weights. However, in the second setting, we rely on the estimated type probabilities and reservation prices that we obtain following a maximum likelihood estimation. The details of the estimation process are deferred to Appendix EC.14. We only present our numerical exercise for the market share maximization problem. Similar results hold for the problem of maximizing consumer welfare.

7.1. Ranking Heuristics

Here, we develop heuristics that can be implemented if the platform (i) can run a reasonable number of experiments, or (ii) obtain an estimate of model parameters. Our heuristics build on the ideas we used in developing Algorithms 1 and 2.

Heuristic 1: This heuristic is motivated by Algorithm 1, where we discretize the weights and solve a DP to identify a ranking. In Algorithm 1, we start with filling up the positions in increasing order of their indices. When deciding about position j, Algorithm 1 (i) does not change its decisions about the first $j-1$ positions and (ii) approximates the value-to-go, i.e., the extra market share from the lower positions, using the discretized weights. Similarly, Heuristic 1 fills up the positions in increasing order of their indices. However, it does not use discretized weights to approximate the value-to-go; instead, it uses the w-ordered heuristic to obtain such an approximation. More precisely, when deciding about the first position, heuristic 1 chooses a product that results in maximum market share assuming that the rest of the positions (i.e., positions $2,\dots,n$) are filled according to the w-ordered heuristic. After fixing the product in the first position, heuristic 1 makes a decision about the second position in a similar way; it chooses one of the remaining products that results in maximum market share assuming that the rest of the positions (i.e., positions $3,\dots,n$) are filled according to the w-ordered heuristic, and so on. Note that when an estimate of the model parameters is not available, running heuristic 1 requires experimenting with $n(n-1)/2 = O(n^2)$ candidate rankings.

Heuristic 2: This heuristic uses the same idea as Algorithm 2.[6] Let M be a small number less than n. Then, this heuristic only optimizes the products in the first M positions. To do so, it fills the first M positions by any M products; then it fills the rest of the positions $(M+1,\dots,n)$ with the remaining products in decreasing order of their weights. This results in $\frac{n!}{(n-M)!} = O(n^M)$ candidate rankings, where the best one, i.e., the one with the highest market share, is returned. Note that when an estimate of the model parameters is not available, running this heuristic requires experimenting with $O(n^M)$ candidate rankings.

7.2. Settings and Evaluation

Settings: We consider a setting with nine products and two types of consumers. We assume that reserve prices have a parametric form. Specifically, the reserve price of a consumer of type $k \in [2]$ for position $j \in [9]$ is $z_k \exp(-q a_j)$ where $\{a_j\}_{j\in[9]}$ are nine equally spaced points in the interval $[0.2, 0.5]$. Here, $a_1 = 0.2$, $a_9 = 0.5$, and $a_j > a_{j-1}$ for $j = \{2,\dots,9\}$. We consider 100 problem instances, where

[6] We clarify that Algorithm 2 is defined for the problem of maximizing consumer welfare; however the same idea can be used for the market share maximization problem.

each problem instance is identified by (θ_1, q, z_1, z_2) with the constraint that $z_1 < z_2$. To generate each problem instance, we randomly draw $\theta_1 \sim U[0.25, 0.75]$, $q \sim U[0.25, 0.5]$, $z_1 \sim U[0.8, 1]$, and $z_2 = z_1 + \zeta_2$ where $\zeta_2 \sim U[0.04, 0.1]$. Finally, the weight of one of the products is randomly drawn from $U[0, 1]$; the weight of four other products is randomly drawn from $U[0, 0.3]$; the weight of the remaining four products is randomly drawn from $U[0, 0.1]$.

Recall that for evaluation, we consider two informational settings, where in the second informational setting, we need to estimate the reservation prices and type probabilities. To obtain the estimation results, for each problem instance, we generate a dataset consisting of $100,000$ queries, where in each query we display nine products to a consumer. These nine products are then displayed to the consumer in a random order. We evaluate our estimation error for the 100 problem instances generated as described above, i.e., each corresponding to a tuple (θ_1, q, z_1, z_2), via two metrics: mean absolute percentage error (MAPE) and mean absolute error (MAE). Our estimation procedure (as described in Appendix EC.14) leads to a MAPE of 3.8% with a standard error of 1.0%, and an MAE of 2.5 with a standard error of 1.0.[7] These numbers suggest that our model can be effectively estimated using a maximum likelihood estimator.

Having the estimation results, we compare the market share obtained from the following schemes: (i) w-ordered ranking, (ii) our PTAS solution (with $\epsilon = 0.25$), (iii) the PTAS based on estimated model parameters, (iv) heuristic 1, (v) heuristic 1 based on estimated model parameters, (vi) heuristic 2 (with $M = 2$), and finally (vii) heuristic 2 based on estimated model parameters. We generate 500 ranking problems where the weight of the nine products are sampled as described above. The set of model parameters (i.e., the tuple (θ_1, q, z_1, z_2)) are sampled out of the 100 instances generated for our estimation exercise (i.e., for which we have the estimated tuple $(\hat{\theta}_1, \hat{q}, \hat{z}_1, \hat{z}_2)$). Using these, for the seven schemes listed above, we compute the ratio of the achieved market share to the optimal one, and we report the mean, median, and standard deviation as tabulated in Table 1 in the same order as listed above. (Here, we compute the optimal solution via enumerating over all possible permutations.)

Based on Table 1 we make the following observations: (i) the market share achieved by our PTAS based on true model primitives and the estimated ones are very close, ensuring that the estimation errors do not result in a significantly inferior ranking. (ii) The same holds for our two heuristics. (iii) Our PTAS substantially outperforms its theoretical guarantee (which is $1 - \epsilon = 0.75$). (iv) W-ordered also outperforms its worst-case guarantee (0.5). (v) However, all of our schemes (which require us to estimate model parameters or experiment with a number rankings) offer improvement over w-ordered.

[7] Let $(\hat{\theta}_1, \hat{q}, \hat{z}_1, \hat{z}_2)$ be our estimation of (θ_1, q, z_1, z_2). Then, $\text{MAPE} = \frac{100}{4}\left(\frac{|\theta_1 - \hat{\theta}_1|}{\theta_1} + \frac{|q - \hat{q}|}{q} + \frac{|z_1 - \hat{z}_1|}{z_1} + \frac{|z_2 - \hat{z}_2|}{z_2}\right)$ and $\text{MAE} = \frac{1}{4}\left(|\theta_1 - \hat{\theta}_1| + |q - \hat{q}| + |z_1 - \hat{z}_1| + |z_2 - \hat{z}_2|\right)$.

	W-ordered	PTAS	PTAS w Est	Heuristic 1	Heuristic 1 w Est	Heuristic 2	Heuristic 2 w Est
Mean	0.8774	0.9994	0.9979	0.9349	0.9340	0.9402	0.9392
Median	0.8754	1.0000	1.0000	0.9316	0.9299	0.9455	0.9448
Std Dev	0.0675	0.0016	0.0101	0.0342	0.0347	0.0519	0.0528

Table 1 Summary statistics: the ratio of market share achieved by various schemes to the optimal market share. Here, we compute the optimal solution via enumerating over all possible permutations.

8. Concluding Remarks and Future Directions

Motivated by empirical evidence related to consumer search behavior on online platforms, we have developed a novel sequential search model. Our model constitutes a two-stage process: In the first stage, the consumer sequentially screens products, which are displayed in different positions, to observe their preference weights and form a consideration set. In the second stage, she observes the idiosyncratic part of utility for the products in her consideration set and then chooses the product with the highest utility. For such a model, we first characterize the optimal policy of a welfare-maximizing consumer. We then turn our attention to the ranking problem that online platforms with different objectives face.

The sequential search process that we introduced in this paper, along with its intuitive optimal policy, can be applicable to other sequential search problems—outside product ranking—where the initial inspection does not reveal the entire reward. Further, we believe that similar consumer search behavior exists in brick-and-mortar stores as well as online platforms that experiment with more sophisticated ways of displaying products. Thus, the present work can serve as a starting point to design product display in more complex settings.

Future Directions: In this work, we focus on designing a ranking algorithm under our novel two-stage search model. On the theoretical side, designing PTAS algorithms under a setting where ρ is not necessarily constant is an interesting direction for future work. To complement our theoretical results, as a future direction, one could study how our model can be estimated from sales data. As another interesting future direction, one could study a model in which consumers are heterogeneous in the intrinsic utility as well as their search costs. A common way to capture such heterogeneity is to *featurize* the products, i.e., define their intrinsic utility as a weighted sum of their features, such as price. (Indeed, such featurization can facilitate the problem of estimating our model.) Then, different consumer types will have different feature weights. Generalizing our model by defining two-dimensional types does not impact the structure of the consumer's optimal policy. Further, with such a generalization, both the problem of maximizing market share and the problem of maximizing consumer welfare is NP-complete. We highlight that with heterogeneity in intrinsic utility, sorting products based on intrinsic utility may not be well defined, as different consumer types may not agree on which product has the highest intrinsic utility, the second highest intrinsic utility, and so on.

Acknowledgments

The authors would like to thank the department editor, associate editor, and three anonymous referees for their valuable comments. Further, they would like to acknowledge Paat Rusmevichientong, Amin Saberi, and Hamid Nazerzadeh for their helpful suggestions.

References

Andrés Abeliuk, Gerardo Berbeglia, Manuel Cebrian, and Pascal Van Hentenryck. 2015. The benefits of social influence in optimized cultural markets. *PloS one* 10, 4 (2015), e0121934.

Andrés Abeliuk, Gerardo Berbeglia, Manuel Cebrian, and Pascal Van Hentenryck. 2016. Assortment optimization under a multinomial logit model with position bias and social influence. *4OR* 14, 1 (2016), 57–75.

Gagan Aggarwal, Jon Feldman, S Muthukrishnan, and Martin Pál. 2008. Sponsored search auctions with markovian users. *Internet and Network Economics* (2008), 621–628.

Ali Aouad, Vivek F Farias, and Retsef Levi. 2015. Assortment optimization under consider-then-choose choice models. (2015).

Ali Aouad, Jacob Feldman, Danny Segev, and Dennis Zhang. 2019. Click-Based MNL: Algorithmic Frameworks for Modeling Click Data in Assortment Optimization. *Available at SSRN 3340620* (2019).

Ali Aouad and Danny Segev. 2015. Display optimization for vertically differentiated locations under multinomial logit choice preferences. (2015).

Susan Athey and Glenn Ellison. 2011. Position auctions with consumer search. *The Quarterly Journal of Economics* 126, 3 (2011), 1213–1270.

Wenjia Ba, Haim Mendelson, and Mingxi Zhu. 2020. Sales Policies for a Virtual Assistant. *Available at SSRN 3718080* (2020).

Moshe E Ben-Akiva, Steven R Lerman, and Steven R Lerman. 1985. *Discrete choice analysis: theory and application to travel demand*. Vol. 9. MIT press.

Yuxin Chen and Song Yao. 2016. Sequential search with refinement: Model and application with click-stream data. *Management Science* (2016).

Leon Yang Chu, Hamid Nazerzadeh, and Heng Zhang. 2017. Position Ranking and Auctions for Online Marketplaces. (2017).

clavisinsight.com. 2015. A Definitive Guide to Optimizing Brand Performance on Amazon. *http://go.clavisinsight.com/rs/573-RCS-461/images/Clavis\%20Winning\%20on\%20Amazon_Final.pdf* (2015).

Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In *Proceedings of the 2008 international conference on web search and data mining*. ACM, 87–94.

James Davis, Guillermo Gallego, and Huseyin Topaloglu. 2013. Assortment planning under the multinomial logit model with totally unimodular constraint structures. (2013).

Babur De los Santos and Sergei Koulayev. 2013. Optimizing click-through in online rankings for partially anonymous consumers. *Unpublished manuscript* (2013).

Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. 2007. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American economic review* 97, 1 (2007), 242–259.

Guillermo Gallego, Anran Li, Van-Anh Truong, and Xinshang Wang. 2016. Approximation algorithms for product framing and pricing. *Submitted to Operations Research* (2016).

Anindya Ghose, Panagiotis G Ipeirotis, and Beibei Li. 2014. Examining the impact of ranking on consumer behavior and search engine revenue. *Management Science* 60, 7 (2014), 1632–1654.

Matthew Goldman and Justin Rao. 2016. Experiments as Instruments: Heterogeneous Position Effects in Sponsored Search Auctions. *EAI Endorsed Trans. Serious Games* 3, 11 (2016), e2.

John R Hauser. 2014. Consideration-set heuristics. *Journal of Business Research* 67, 8 (2014), 1688–1699.

John R Hauser and Birger Wernerfelt. 1990. An evaluation cost model of consideration sets. *Journal of consumer research* 16, 4 (1990), 393–408.

Srikanth Jagabathula and Paat Rusmevichientong. 2016. A nonparametric joint assortment and price choice model. *Management Science* (2016).

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

Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. 2007. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Transactions on Information Systems (TOIS)* 25, 2 (2007), 7.

David Kempe and Mohammad Mahdian. 2008. A cascade model for externalities in sponsored search. *Internet and Network Economics* (2008), 585–596.

Jun B Kim, Paulo Albuquerque, and Bart J Bronnenberg. 2010. Online demand under limited consumer search. *Marketing science* 29, 6 (2010), 1001–1023.

Jon Kleinberg and É Tardos. 2006. Algorithm Design. ed.

Yanzhe Murray Lei, Stefanus Jasin, Joline Uichanco, and Andrew Vakhutinsky. 2018. Randomized Product Display (Ranking), Pricing, and Order Fulfillment for E-commerce Retailers. *Available at SSRN: 3282019* (2018).

Pierre L'Ecuyer, Patrick Maillé, Nicolás E Stier-Moses, and Bruno Tuffin. 2017. Revenue-maximizing rankings for online platforms with quality-sensitive consumers. *Operations Research* 65, 2 (2017), 408–423.

Siddharth Mahajan and Garrett Van Ryzin. 2001. Inventory competition under dynamic consumer choice. *Operations Research* 49, 5 (2001), 646–657.

Donald Ngwe. 2017. Why outlet stores exist: Averting cannibalization in product line extensions. *Marketing Science* 36, 4 (2017), 523–541.

Donald Ngwe and Thales Teixeira. 2017. The impact of increasing search frictions on online shopping behavior: Evidence from a field experiment. (2017).

Donald Ngwe, Thales S Teixeira, and Kris Ferreira. 2018. *The impact of increasing search frictions on online shopping behavior: Evidence from a field experiment.* Harvard Business School.

John H Roberts and James M Lattin. 1991. Development and testing of a model of consideration set composition. *Journal of Marketing Research* (1991), 429–440.

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

Kenneth E Train. 2009. *Discrete choice methods with simulation.* Cambridge university press.

Raluca Mihaela Ursu. 2016. The Power of Rankings: Quantifying the Effect of Rankings on Online Consumer Search and Purchase Decisions. *Browser Download This Paper* (2016).

Stefano Vaccari, Costis Maglaras, and Marco Scarsini. 2018. Social Learning from Online Reviews with Product Choice. (2018).

Hal R Varian. 2007. Position auctions. *international Journal of industrial Organization* 25, 6 (2007), 1163–1178.

Ruxian Wang and Ozge Sahin. 2017. The impact of consumer search cost on assortment planning and pricing. *Management Science* (2017).

Martin L Weitzman. 1979. Optimal search for the best alternative. *Econometrica: Journal of the Econometric Society* (1979), 641–654.

WSJ. 2017. Amazon's Early Christmas Bonus. *https://www.wsj.com/articles/amazons-early-christmas-bonus-1511891465* (2017).

EC.1. Empirical Evidence

In this section, we present empirical evidence in support of our model for consumer behavior. Utilizing a small dataset from the platform discussed in Section 1, we investigate how well our model explains the data and compare that to the performance of alternative models. In this exercise, we restrict our attention to all the queries in which the number of products displayed, n, is 2. As described in Section 1, the platform displays rankings with a larger size; however, we are unable to extend our analysis to $n \geq 3$ due to the sparsity of our dataset.

We wish to follow the estimation strategy laid out in Section EC.14 to estimate model parameters. However, even for rankings of size two, our dataset is not rich enough: for each category of products, we do not have enough queries in which preference weights vary noticeably. Further, we do not have data on where the consumer stopped screening. Instead, we only have data on "purchase events." Namely, for each query, we know whether the consumer purchased any of the products, and if so, which one. [8] Lastly, we have estimates for preference weights only relative to an outside option whose weight needs to be jointly estimated along with the search costs and type probabilities. Therefore, we slightly modify our estimation strategy, which we describe next.

Our dataset includes several pairs of products that have been displayed under both possible rankings. For a pair of products 1 and 2, the input into the estimation can be summarized as a vector $p = [p_{1,0}\ p_{1,1}\ p_{1,2}\ p_{2,0}\ p_{2,1}\ p_{2,2}]$, where $p_{1,i}$ represents the purchase probability of product $i = 0, 1, 2$ under ranking $(1, 2)$, and $p_{2,i}$ represents the purchase probability of product i under ranking $(2, 1)$. Note that product 0 represents the no-purchase option.

The estimation in the general form involves estimating the consumer-type distribution $(\theta^1, \ldots, \theta^K)$ and the associated search costs (or, equivalently, reservation prices r_j^k) and w_0. The formation of the consideration set in the first stage implies that the consideration set itself and the purchase probabilities only depend on the relation between the preference weights and the reservation prices, and not on the values of the reservation prices. This implies that with three products (the two products plus the no-purchase option), it suffices to determine whether r_1—the reservation price for the highest position—belongs to one of four regions: larger than the preference weight of all three products, between the largest and the second largest preference weights, between the second largest and the smallest preference weight, and smaller than the preference weights of all the products. Assuming one of these cases for r_1, we can consider all cases for r_2 with the constraint $r_2 \leq r_1$. Therefore, in our estimation process, we do not determine the exact values

[8] We exclude (a small number of) queries in which the consumer purchased both products.

of reservation prices. Alternatively, we estimate the probability that reservation prices belong to a specific region. We point out that each combination of (r_1, r_2) discussed above corresponds to a potential consumer type. However, to keep the notation simple, we do not explicitly index the reservation prices.

For the above model, we are faced with the problem of estimating a mixture model with the consumer type as the latent variable. Utilizing the Expectation Maximization (EM) algorithm, we compute the maximum likelihood estimation of w_0 and the consumer-type distribution. In Figure EC.1, we present the estimated purchase probabilities resulting from running the EM algorithm along with the actual probabilities for two pairs of products that we call Pair 1 and Pair 2. Pair 1 has already been defined in Section 1. Pair 2 is also among the most frequent pairs shown. The products in Pair 1 are different from those in Pair 2. As mentioned in Section 1, the first product in Pair 1, denoted by Product 1, is significantly more popular than the second product, denoted by Product 2. In Pair 2, however, the first product, which (with a slight abuse of notation) we also call Product 1, is only slightly more popular than the second product. In the figure, the absolute values of the purchase probabilities are hidden for confidentiality, but the ratios are preserved, and they indicate the closeness of our estimated probabilities to the actual ones.

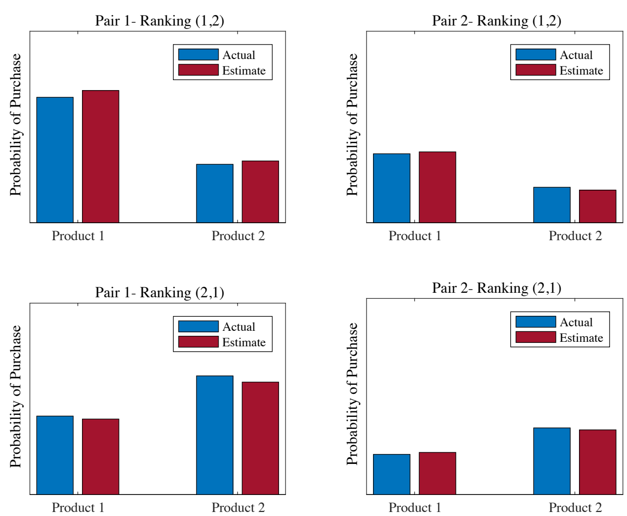


Figure EC.1 The purchase probabilities along with their estimates. The absolute values are hidden for confidentiality, but the ratios are preserved.

Finally, in Table EC.1, we report the estimation error for Pairs 1 and 2 via two metrics: (MAPE) and (MAE).[9] We further report the Akaike information criterion (AIC) value of our model.[10]

[9] Let us denote our estimation of purchase probabilities $p = [p_{1,0} \ p_{1,1} \ p_{1,2} \ p_{2,0} \ p_{2,1} \ p_{2,2}]$ by $\hat{p} = [\hat{p}_{1,0} \ \hat{p}_{1,1} \ \hat{p}_{1,2} \ \hat{p}_{2,0} \ \hat{p}_{2,1} \ \hat{p}_{2,2}]$. Then, $\text{MAPE} = \frac{100}{6}\left(\sum_{\ell=1}^{2}\sum_{i=1}^{3}\frac{|\hat{p}_{\ell,i}-p_{\ell,i}|}{p_{\ell,i}}\right)$ and $\text{MAE} = \frac{1}{6}\left(\sum_{\ell=1}^{2}\sum_{i=1}^{3}|\hat{p}_{\ell,i}-p_{\ell,i}|\right)$.

[10] Let l be the number of estimated parameters in a model and $\hat{L}$ be the maximum value of the likelihood function for the model. Then the AIC value of the model is given by $2l - 2\log(\hat{L})$. We prefer the AIC value to be small.

Model	Pair 1			Pair 2		
	MAPE	MAE	AIC	MAPE	MAE	AIC
Our Two-Stage Model	3.8%	6.9e-3	1972	3.2%	1.6e-3	1111
Pos-Ind Model	17.0%	1.8e-2	2019	14.7%	1.0e-2	1172
MNL	21.0%	2.1e-2	1973	20.9%	1.1e-2	1106

Table EC.1 Comparing different models in terms of MAPE, MAE, and AIC.

Additionally, along with our model, we compute the estimation error and the AIC value of the following two models.

MNL Model: In this model, there is no search cost, and the consumer faces a standard MNL choice model. This model is rank oblivious; that is, the probability of purchase of product $i = 0, 1, 2$ does not depend on how products are displayed.

Pos-Ind Model: This model is a special case of our model with a small twist. Here, the search cost does not depend on the position; that is, $s_1^k = s_2^k$ for any type k.

Table EC.1 shows that, compared with these two models, our model results in significantly lower estimation errors. This highlights the fact that even with two products, search costs play an important role in explaining consumers' choice. Moreover, the finding implies that consumers' search costs increase as they move away from the higher position and that failing to consider this can lead to a bad estimation. Finally, observe that the Pos-Ind model, which considers search costs to some extent, performs better than the MNL model, which completely ignores them. Table EC.1 also shows that the issue of overfitting in our model is under control, as the AIC value of our model is comparable to that of the MNL model, which has two parameters to estimate.

EC.2. Uniqueness of Reservation Prices

LEMMA EC.1. *For any $k \in [K]$ and $j \in [n]$ such that $s_j^k \in (0, \infty)$, there exists a unique $r_j^k \in (-1, \infty)$ that solves Equation* (5). *If $s_j^k = 0$, then $r_j^k = \infty$.*

Proof: For any $r \in (-1, \infty)$ define random variable $\Phi(r) := \log\left(\frac{1+r+W}{1+r}\right)$ and recall that $W \sim f_W$.Now all of the following hold *almost surely*, because of the continuity and strict monotonicity of the $\log(\cdot)$ function and our modeling assumption that f_W has support of $[w_{\min}, w_{\max}]$ where $w_{\min}$ and $w_{\max}$ are both positive and finite:

1. $\lim_{r \to -1^+} \Phi(r) = \infty$ (a.s.)
2. $\lim_{r \to \infty} \Phi(r) = 0$ (a.s.)
3. For $r_1 > r_2$, we have $\Phi(r_1) < \Phi(r_2)$ (a.s.)
4. For any $\epsilon > 0$, there exists $\delta > 0$ such that if $r_1 - r_2 \le \epsilon$ then $\Phi(r_2) - \Phi(r_1) \le \delta$ (a.s.)

Roughly speaking, the AIC gets smaller as the model better describes the observed samples using a smaller number of parameters; that is, while the AIC encourages a better fit, it discourages overfitting.

With these almost sure results, all we need to do is to use the observation that by taking expectation (over the support of f_W which is $[w_{\min}, w_{\max}]$), we have the same properties, i.e.,

1. $\lim_{r\to -1^+} \mathbb{E}[\Phi(r)] = \infty$
2. $\lim_{r\to\infty} \mathbb{E}[\Phi(r)] = 0$
3. For $r_1 > r_2$ we have $\mathbb{E}[\Phi(r_1)] < \mathbb{E}[\Phi(r_2)]$. Namely, $\mathbb{E}[\Phi(r)]$ is strictly decreasing in r.
4. For any $\epsilon > 0$, there exists $\delta > 0$ such that if $r_1 - r_2 \le \epsilon$ then $\mathbb{E}[\Phi(r_2) - \Phi(r_1)] \le \delta$. Namely, $\mathbb{E}[\Phi(r)]$ is continuous in r.

Finally, note that reservation price, r_j^k, solves $\mathbb{E}[\Phi(r_j^k)] = s_j^k$. Thus, because $\mathbb{E}[\Phi(r)] : (-1,\infty) \mapsto (0,\infty)$ is a continuous and strictly decreasing function of r, if s_j^k is a finite and positive number, then there exists a unique r_j^k that solves $\mathbb{E}[\Phi(r_j^k)] = s_j^k$. On the other hand, if $s_j^k = 0$, then $r_j^k = \infty$.

EC.3. Proof of Lemma 1

Let $Z = \max_{i\in\mathcal{C}\cup\{0\}} \{\log(w_i) + Z_i\}$, where Z_i's are i.i.d. We prove that if $Z_i \sim \text{Gumbel}(0,1)$, for $i \in \mathcal{C}\cup\{0\}$, then $Z \sim \text{Gumbel}(\log(1 + w(\mathcal{C})), 1)$. Then, the result follows because $\mathbb{E}[Z] = \log(1 + w(\mathcal{C})) + \gamma$.

In order to show that $Z \sim \text{Gumbel}(\log(1 + w(\mathcal{C})), 1)$, we compute the logarithm of the CDF of Z:

$$\begin{aligned}
\log(\mathbb{P}(Z \le x)) &= \log\Big(\prod_{i\in\mathcal{C}\cup\{0\}} \mathbb{P}(\log(w_i) + Z_i \le x)\Big) \\
&= \sum_{i\in\mathcal{C}\cup\{0\}} \log(\mathbb{P}(Z_i \le x - \log(w_i))) \\
&= -\sum_{i\in\mathcal{C}\cup\{0\}} e^{-(x-\log(w_i))} \\
&= -e^{-x} \sum_{i\in\mathcal{C}\cup\{0\}} w_i \\
&= -e^{-x}(1 + w(\mathcal{C})) \\
&= -e^{-[x-\log(1+w(\mathcal{C}))]}
\end{aligned}$$

Thus, $Z \sim \text{Gumbel}(\log(1 + w(\mathcal{C})), 1)$, completing the proof.

EC.4. Proof of Theorem 1

We prove the theorem by induction on the number of unscreened positions, which we denote by m. The base of our induction shows that if only one position is unscreened, the consumer follows our index-based policy; that is, she screens that position if the sum of the weights of the products screened so far is less than the reservation price of that position. Then, for our induction, we show that if our index-based policy is optimal when m positions are unscreened, then our index-based policy is optimal when $m+1$ positions are unscreened. The optimality of our index-based policy implies that (i) the consumer does not screen another position if the sum of the weights of the products screened so far is larger than the reservation prices of all the remaining unscreened positions and that (ii) if the consumer screens another position, then she will screen the position with the highest reservation price.

Base case ($m = 1$): First, we provide proof for the base case of $m = 1$. In other words, suppose that the consumer has screened all positions except for position $j \in [n]$, i.e., $\mathcal{C} = [n] \setminus \{\pi(j)\}$. Let us denote the item in position j by i, i.e., $\pi(j) = i$. If the consumer does not screen position j, her expected welfare is

$$\mathbb{E}\left[\max_{i' \in \mathcal{C} \cup \{0\}} \{\log(w_{i'}) + Z_{i'}\}\right] - \sum_{i' \in \mathcal{C}} s^k_{\pi^{-1}(i')} = \log\big(1 + w(\mathcal{C})\big) + \gamma - \sum_{i' \in \mathcal{C}} s^k_{\pi^{-1}(i')} ,$$

where the equality holds because of Lemma 1. In contrast, if the consumer screens position j, her expected welfare is given by

$$\mathbb{E}\left[\max\left\{\log(W_i) + Z_i, \max_{i' \in \mathcal{C} \cup \{0\}} \{\log(w_{i'}) + Z_{i'}\}\right\}\right] - \sum_{i' \in [n]} s^k_{\pi^{-1}(i')} ,$$

where the expectation is w.r.t. $Z_{i'}$, $i' \in [n] \cup \{0\}$, and W_i. Applying Lemma 1, the above expression can be written as

$$\mathbb{E}\Big[\log\Big(1 + w(\mathcal{C}) + W_i\Big)\Big] + \gamma - \sum_{i' \in [n]} s^k_{\pi^{-1}(i')} .$$

Putting these together, the consumer screens position j if

$$\log\Big(1 + w(\mathcal{C})\Big) < \mathbb{E}\Big[\log\Big(1 + w(\mathcal{C}) + W_i\Big)\Big] - s^k_j .$$

Then, by definition of reservation prices given in (5), the consumer screens position j if $w(\mathcal{C}) < r^k_j$, which is the same decision that our index-based policy makes.

General case ($1 \le m < n - 1$): Assume that following the index-based policy is optimal when there are m unscreened positions. We will show that the index-based policy is also optimal when there are $m + 1$ unscreened positions. To do so, we first show that the consumer stops screening if the sum of the weights of the products screened so far exceeds the reservation price of all the unscreened positions. Then, we show that if the consumer continues screening, she will screen the position with the highest reservation price.

Let j^* be the position among the unscreened ones that has the highest reservation price. Let $\mathcal{J}$ denote the set of positions that the consumer has screened so far (we know $|\mathcal{J}| = n - (m+1)$). Then, $j^* = \arg\max_{j \in [n] \setminus \mathcal{J}} \{r^k_j\}$. We denote the product in position j^* by i^*. Further, we use $\mathcal{C}$ to denote the set of products in positions $\mathcal{J}$; that is, $\mathcal{C} = \pi(\mathcal{J})$. Thus, $i^* := \pi(j^*)$, and $i \in \mathcal{C}$ iff $\pi^{-1}(i) \in \mathcal{J}$. We consider the following two cases:

Case 1 ($w(\mathcal{C}) \geq r^k_{j^*}$): In this case, we show that it is optimal for the consumer to stop screening. Note that this is the same decision that our index-based policy makes. To verify this result, we need to show that the welfare expected by the consumer when she stops is greater than that when she does not stop.

Suppose that the consumer does not stop and screens position $g \in [n] \setminus \mathcal{J}$. After screening position g, we are left with only m unscreened positions, and thus, by our induction assumption, the index-based policy is optimal. According to the index-based policy, after screening position g, it is optimal for the consumer to stop. This is so because by screening position g, the total weight seen so far is $w_{\pi^{-1}(g)} + w(\mathcal{C})$, which is greater than the sum of the weights before screening position g, i.e., $w(\mathcal{C})$. Then, considering that fact that under Case 1, $w(\mathcal{C}) \geq r^k_{j^*}$, we have

$$w_{\pi^{-1}(g)} + w(\mathcal{C}) \;\geq\; w(\mathcal{C}) \;\geq\; r^k_{j^*} \;\geq\; \max_{j \in [n]\setminus(\mathcal{J}\cup\{g\})} \{r^k_j\}\,,$$

which implies that the consumer stops after screening position g. Hence, in this case, by Lemma 1, the consumer welfare is given by

$$\mathbb{E}\Big[\log\Big(1 + w(\mathcal{C}) + W_{\pi^{-1}(g)}\Big)\Big] + \gamma - \sum_{j \in \mathcal{J}\cup\{g\}} s^k_j\,,$$

where the expectation is w.r.t. $W_{\pi^{-1}(g)} \sim f_W$. The value of the above expression is less than the consumer welfare when she stops her search; i.e., $\log\Big(1 + w(\mathcal{C})\Big) + \gamma - \sum_{j\in\mathcal{J}} s^k_j$. This is so because $w(\mathcal{C}) \geq r^k_{j^*} > r^k_g$, and by definition, $\mathbb{E}\Big[\log\Big(1 + r^k_g + W\Big)\Big] - s^k_g = \log(1 + r^k_g)$.

Case 2 ($w(\mathcal{C}) < r^k_{j^*}$): In this case, first note that we can rule out the optimality of stopping. To see why, note that by definition of $r^k_{j^*}$, if the consumer just screens j^* and then stops, her expected welfare would be higher than that if she does stop. Next, we show that it is optimal to first screen position j^*, i.e., to follow our index-based policy.

To show that, we construct the following two policies. In the first policy, the consumer first screens another position $g \neq j^*$ and follows the index-based policy after that. Note that because after screening g, there are only m unscreened positions, by our induction assumption, following the index-based policy is indeed optimal. We represent the (random) welfare of this policy by Wel_1. In the second policy, the consumer screens j^* first. Then, if $w(\mathcal{C}) + w_{i^*} \geq \max_{j\in[n]\setminus(\mathcal{J}\cup\{j^*\})}\{r^k_j\}$, she stops. Otherwise, she screens product g next and follows the index-based policy afterward. We represent the (random) welfare of this policy by Wel_2. In order to establish the optimality of the index-based policy, it suffices to show that

$$\mathbb{E}[\mathsf{Wel}_1] \leq \mathbb{E}[\mathsf{Wel}_2]\,. \tag{EC.1}$$

This is the case because by our induction assumption, the consumer welfare under the index-based policy is greater than or equal to the consumer welfare under the second policy. Inequality (EC.1) is proved in Lemma EC.2.

LEMMA EC.2. *Let* $\mathbb{E}[\mathsf{Wel}_1]$ *and* $\mathbb{E}[\mathsf{Wel}_2]$ *respectively be the expected consumer welfare under the first and second policies defined above. Then, we have* $\mathbb{E}[\mathsf{Wel}_1] \leq \mathbb{E}[\mathsf{Wel}_2]$.

The proof of Lemma EC.2 is presented in Section EC.4.1. The above lemma completes the proof of Theorem 1.

EC.4.1. Proof of Lemma EC.2

We show inequality (EC.1) by analyzing different cases that can arise under each policy. To simplify notation, we define position $h := \arg\max_{j\in[n]\setminus(\mathcal{J}\cup\{j^*\})}\{r_j^k\}$. That is, position h has the second highest reservation prices among the set $[n]\setminus\mathcal{J}$ positions. Finally, we define y to be the product in position g, i.e., $\pi(g) = y$.

Let us enumerate all the cases that happen under the first policy:

- Event $\mathcal{A}_1$: $w(\mathcal{C}) + w_y \geq r_{j^*}^k$. Under this event the consumer stops. This is so because under this policy, the consumer first screens position g and follows the index-based policy afterwards.
- Event $\mathcal{B}_1$: $w(\mathcal{C}) + w_y \in [r_h^k, r_{j^*}^k)$. Under this event, the consumer screens position j^* next, but then stops because $w(\mathcal{C}) + w_y + w_{i^*} > w(\mathcal{C}) + w_y \geq r_h^k$.
- Event $\mathcal{C}_1$: $w(\mathcal{C}) + w_y < r_h^k$. Under this event, the consumer screens position j^* and then continues to follow the index-based policy.

Similarly, we enumerate all the cases that happen under the second policy:

- Event $\mathcal{A}_2$: $w(\mathcal{C}) + w_{i^*} \geq r_{j^*}^k$. Under this event, the consumer stops.
- Event $\mathcal{B}_2$: $w(\mathcal{C}) + w_{i^*} \in [r_h^k, r_{j^*}^k)$. Under this event, the consumer stops.
- Event $\mathcal{C}_2$: $w(\mathcal{C}) + w_{i^*} < r_h^k$. Under this event, the consumer screens position g and then continues to follow the index-based policy.

First note that because the weights of all the products have the same distribution (regardless of their positions), we have $\mathbb{P}(\mathcal{A}_1) = \mathbb{P}(\mathcal{A}_2)$, $\mathbb{P}(\mathcal{B}_1) = \mathbb{P}(\mathcal{B}_2)$, and $\mathbb{P}(\mathcal{C}_1) = \mathbb{P}(\mathcal{C}_2)$. Next, note that

$$\mathbb{E}[\mathsf{Wel}_1 \cdot \mathbb{I}(\mathcal{C}_1)] = \mathbb{E}[\mathsf{Wel}_2 \cdot \mathbb{I}(\mathcal{C}_2)]\,, \tag{EC.2}$$

where $\mathbb{I}(\cdot)$ is an indicator function: $\mathbb{I}(\mathcal{C}_1) = 1$ when event $\mathcal{C}_1$ happens and zero otherwise. This is because the set of screened products under the first policy and event $\mathcal{C}_1$ will be the same as that under the second policy and event $\mathcal{C}_2$. Thus we are left to show that:

$$\mathbb{E}[\mathsf{Wel}_1 \cdot \mathbb{I}(\mathcal{A}_1)] \leq \mathbb{E}[\mathsf{Wel}_2 \cdot \mathbb{I}(\mathcal{A}_2)]\,, \tag{EC.3}$$

and

$$\mathbb{E}[\mathsf{Wel}_1 \cdot \mathbb{I}(\mathcal{B}_1)] \leq \mathbb{E}[\mathsf{Wel}_2 \cdot \mathbb{I}(\mathcal{B}_2)]\,. \tag{EC.4}$$

We first verify inequality (EC.3). We start by writing the definitions of $\mathbb{E}[\mathsf{Wel}_1 \cdot \mathbb{I}(\mathcal{A}_1)]$ and $\mathbb{E}[\mathsf{Wel}_2 \cdot \mathbb{I}(\mathcal{A}_2)]$:

$$\mathbb{E}[\mathsf{Wel}_1 \cdot \mathbb{I}(\mathcal{A}_1)] = \mathbb{E}\Big[\log(1+w(\mathcal{C})+W_y)\ \mathbb{I}(w(\mathcal{C})+W_y \geq r^k_{j^*})\Big] + \mathbb{P}(\mathcal{A}_1)\Big[\gamma - \sum_{j\in\mathcal{J}\cup\{g\}} s^k_j\Big].$$

Similarly for the second policy, we have:

$$\mathbb{E}[\mathsf{Wel}_2 \cdot \mathbb{I}(\mathcal{A}_2)] = \mathbb{E}\Big[\log(1+w(\mathcal{C})+W_{i^*})\ \mathbb{I}(w(\mathcal{C})+W_{i^*} \geq r^k_{j^*})\Big] + \mathbb{P}(\mathcal{A}_2)\Big[\gamma - \sum_{j\in\mathcal{J}\cup\{j^*\}} s^k_j\Big].$$

With the above definitions, we show (EC.3) by making the following two observations: (1) The weights of products y and i^* are identically distributed; therefore we have: $\mathbb{E}\Big[\log(1+w(\mathcal{C})+W_y)\ \mathbb{I}(w(\mathcal{C})+W_y \geq r^k_{j^*})\Big] = \mathbb{E}\Big[\log(1+w(\mathcal{C})+W_{i^*})\ \mathbb{I}(w(\mathcal{C})+W_{i^*} \geq r^k_{j^*})\Big]$. (2) The reservation price is decreasing in cost. Thus $r^k_{j^*} \geq r^k_g$ implies that $s^k_{j^*} \leq s^k_g$.

Now we proceed to show (EC.4). We start by writing $\mathbb{E}[\mathsf{Wel}_1 \cdot \mathbb{I}(\mathcal{B}_1)]$ as follows

$$\mathbb{E}[\mathsf{Wel}_1 \cdot \mathbb{I}(\mathcal{B}_1)] = \mathbb{E}\Big[\log(1+w(\mathcal{C})+W_y+W_{i^*})\ \mathbb{I}\big((w(\mathcal{C})+W_y) \in [r^k_h, r^k_{j^*})\big)\Big] + \mathbb{P}(\mathcal{B}_1)\Big[\gamma - \sum_{j\in\mathcal{J}\cup\{g,j^*\}} s^k_j\Big]. \tag{EC.5}$$

Note that the expectation above is w.r.t. two random variables W_y and W_{i^*}. By the law of iterated expectation, we re-write the first term of the above expectation as follows:

$$\begin{aligned}&\mathbb{E}\Big[\log(1+w(\mathcal{C})+W_y+W_{i^*})\ \mathbb{I}((w(\mathcal{C})+W_y) \in [r^k_h, r^k_{j^*}))\Big]\\ &= \mathbb{E}_{W_y}\big[\mathbb{E}_{W_{i^*}}[\log(1+w(\mathcal{C})+W_y+W_{i^*})\ \mathbb{I}(w(\mathcal{C})+W_y \in [r^k_h, r^k_{j^*}))\mid W_y]\big] = \mathbb{E}_{W_y}[f(W_y)]\end{aligned}$$

where we define function $f(W_y) := \mathbb{E}_{W_{i^*}}[\log(1+w(\mathcal{C})+W_y+W_{i^*})\ \mathbb{I}(w(\mathcal{C})+W_y \in [r^k_h, r^k_{j^*}))\mid W_y]$. We have:

$$f(W_y) = \begin{cases}\mathbb{E}_{W_{i^*}}[\log(1+w(\mathcal{C})+W_y+W_{i^*})] & w(\mathcal{C})+W_y \in [r^k_h, r^k_{j^*})\\ 0 & \text{otherwise}\end{cases}$$

Next we claim that for $w(\mathcal{C})+W_y \geq r^k_h$:

$$-s^k_g + \mathbb{E}_{W_{i^*}}[\log(1+w(\mathcal{C})+W_y+W_{i^*})] \leq \log(1+w(\mathcal{C})+W_y) \tag{EC.6}$$

Note that the above follows from the definition of reservation price r^k_g, the fact that the weights of products i^* and y are identically distributed, and the fact that $r^k_g \leq r^k_h$. Plugging (EC.6) back into (EC.5), we have:

$$\begin{aligned}\mathbb{E}[\mathsf{Wel}_1 \cdot \mathbb{I}(\mathcal{B}_1)] &\leq \mathbb{E}_{W_y}\Big[\log(1+w(\mathcal{C})+W_y)\ \mathbb{I}(w(\mathcal{C})+W_y \in [r^k_h, r^k_{j^*}))\Big] + \mathbb{P}(\mathcal{B}_1)\Big[s^k_g + \gamma - \sum_{j\in\mathcal{J}\cup\{g,j^*\}} s^k_j\Big]\\ &= \mathbb{E}_{W_y}\Big[\log(1+w(\mathcal{C})+W_y)\ \mathbb{I}(w(\mathcal{C})+W_y \in [r^k_h, r^k_{j^*}))\Big] + \mathbb{P}(\mathcal{B}_1)\Big[\gamma - \sum_{j\in\mathcal{J}\cup\{j^*\}} s^k_j\Big]\\ &= \mathbb{E}[\mathsf{Wel}_2 \cdot \mathbb{I}(\mathcal{B}_2)],\end{aligned}$$

where the last equality follows from two observations: (1) Under event $\mathcal{B}_2$, the second policy stops after screening position j^*. (2) The weights of products i^* and y are identically distributed.

EC.5. Discussion on Uniform Knapsack

In the proof of our hardness results (Theorem 2 and Proposition 1), we reduce our problems to a general instance of a version of uniform knapsack which is NP-complete. Thus, before presenting our hardness proofs, we clearly state our definition of the uniform knapsack problem, and we show it is NP-complete.

DEFINITION EC.1 (c-BOUNDED UNIFORM KNAPSACK PROBLEM). For any given constant number $c<1$, we define the *c-bounded uniform knapsack problem* as follows. We are given a knapsack of size K and n items with weights $0 \le w_i < K$, $i \in [n]$ where $\frac{\min_{i\in[n]} w_i}{\max_{i\in[n]} w_i} > c$. The goal in this problem is to find a subset A of items whose total size is strictly less than K and is maximized; that is, we maximize $w(A) = \sum_{i\in A} w_i$ subject to $w(A) < K$.

To show that this problem is NP-complete, we consider any instance of the subset sum problem, and we solve it using polynomially many instances of the c-bounded uniform knapsack problem. In the subset sum problem, we are given a set $S = \{s_1, \ldots, s_n\}$ of non-negative integers and a number $M > 1$ and the goal is to determine whether or not there is a subset of S whose total sum is equal to M. (Note that if $M = 1$ the problem trivially reduces to checking whether there is any $s_i = 1$.) Without loss of generality, we assume that for any $i \in [n]$ we have $s_i < M$. Note that this version of subset sum problem is known to be NP-complete (Kleinberg and Tardos 2006). Thus reducing it to the c-bounded uniform knapsack problem proves the hardness of this problem as well.

Let us fix a constant c' such that $1 > c' > c$ and $\frac{c'}{1-c'}$ is a positive integer which we denote by x. We create n instances of c'-bounded uniform knapsack problem such that the solution to the subset sum problem can be retrieved in polynomial time given the solution to these instances.

For any $j \in [n]$, we create an instance of c'-bounded uniform knapsack problem as follows. The capacity of the knapsack in this instance is $2(xnj+1)M+1$. For any item $s_i \in S$, we have an item in this instance with size $2(xnM + s_i)$. Let Ω denote the set of these items. The ratio of the minimum size item to the maximum size one in Ω is lower-bounded by $\frac{2xnM}{2(x+1)nM} = c'$. To see why note that

$$\frac{\min_{i\in[n]}\left(2(xnM+s_i)\right)}{\max_{i\in[n]}\left(2(xnM+s_i)\right)} \ge \frac{xnM}{(x+1)nM} = \frac{x}{(x+1)} = c'\,.$$

We claim that the optimal solution of the subset sum problem can be found using the solutions of these n instances, in polynomial time. Remark that proving this claim shows the NP-completeness of the c-bounded knapsack problem.

Now suppose that the optimal solution of the subset sum problem, denoted by S^*, has size j^* (i.e., $|S^*| = j^*$). We claim that the problem instance j^* must have the same optimal solution.

Proving this claim will complete the proof. We show the claim in three steps: (1) Note that any feasible solution of the instance j^* has at most j^* items. To see why note that size of any item is an integer greater than $2xnM$ and the capacity of the knapsack is $2(xnj^*+1)M+1$, (2) Subset S^* is a feasible solution to the problem instance j^*. This is because:

$$\sum_{i\in S^*} s_i = M \Longleftrightarrow \sum_{i\in S^*} 2(xnM+s_i) = 2(xnj^*+1)M < 2(xnj^*+1)M+1.$$

Clearly, S^* is the best solution to the problem instance j^* among the feasible solutions with size j^*. Finally, (3) any feasible solution to instance j^* that has less than j^* items is dominated by S^*. The reason is that (i) any set of at most j^*-1 items in Ω has a total weight of at most $2xn(j^*-1)M+2(j^*-1)M$ (because $s_i < M$, $i\in[n]$), (ii) any set of j^* items has a total weight of at least $2xnj^*M$ (because $s_i \geq 0$, $i\in[n]$), and (iii) $xn(j^*-1)M+(j^*-1)M < xnj^*M$ (because $x\geq 1$).

EC.6. Proof of Theorem 2

We will reduce a general instance of the c-bounded uniform knapsack problem, as stated in Definition EC.1 in Section EC.5, to our market share maximization problem in which $\rho = 2/c$. Recall that in Section EC.5 we showed that the c-bounded uniform knapsack problem is NP-complete. Further we remind the reader that $\rho > 1$ is the ratio of maximum weight to the minimum weight in the range of the distribution of product weights, f_W. We specify ρ and confirm that it is $2/c$ when defining our instance of market share maximization problem.

In the instance of the c-bounded uniform knapsack problem, we are given a set of n items with size $w_1,\ldots,w_n$ where $c = \frac{\min_{i\in[n]} w_i}{\max_{i\in[n]} w_i}$ and a knapsack of capacity $M>0$, with $0\leq w_i < M$ for any $i\in[n]$. The goal in this problem is to find a subset A of items whose total size is strictly less than M and is maximized; that is, we maximize $w(A) = \sum_{i\in A} w_i$ subject to $w(A)<M$.

Using this general instance of the c-bounded uniform knapsack problem (which is equivalent to Definition EC.1, with $M=K$), we create an instance of the rank optimization problem with a set of $n+1$ products denoted by P and one consumer type. (To simplify notation, when there is only one type of consumer, we drop the superscript k.) In this instance of the rank optimization problem, the reservation price r_j for any position $j\in[n+1]$ is M. That is, all positions have the same search cost. (Later in the proof, we define the search cost, s, and the weight distribution, f_W, in a way that leads to having $r_j = M$, for any $j\in[n+1]$.) For any item with size w_i in the c-bounded knapsack problem instance defined above, we have a product with the same weight w_i in set P. We also have a $n+1$-st product with weight $w_{n+1} = \bar{w}$ where we define $\bar{w} := 2\max_{i\in[n]} w_i$. Note that w_{n+1} has the highest weight in P.

We finish the description of the market share maximization problem instance by defining the weight distribution, f_W, and the search cost s as follows:

$$f_W(w_i) = \frac{1}{n+1}, \quad i \in [n+1] \quad \text{and} \quad s = \frac{1}{n+1}\sum_{i=1}^{n+1} \log\left(1 + \frac{w_i}{1+M}\right).$$

To see why the reservation price, r_j, equals M for the above weight distribution and search cost, note that we can re-write the definition of reservation prices given in (5) as $s = s_j = \mathbb{E}_W[\log(1 + \frac{W}{1+r_j})]$. Finally we confirm that $\rho = 2/c$: By definition $\rho = \frac{w_{n+1}}{\min_{i\in[n]} w_i} = \frac{2\max_{i\in[n]} w_i}{\min_{i\in[n]} w_i} = 2/\mathrm{c}$.

Having described the market maximization problem instance, we now focus on the properties of the rank optimization problem (Ms) to show that this problem is NP-complete. When there is one type of consumer, we would like to find a ranking π that maximizes $\frac{w(\mathcal{C}_\pi)}{1+w(\mathcal{C}_\pi)}$. Then, considering the fact that function $x \mapsto \frac{x}{1+x}$ is increasing in x, we would like to choose a ranking π under which $w(\mathcal{C}_\pi)$, i.e., the sum of the weight of the products in the consumer's consideration set, is maximized. By the stopping rule in (6), and that $r_j = M$ for any $j \in [n+1]$, the consideration set under ranking π only consists of the j products in the top positions if $\sum_{j'\in[j-1]} w_{\pi^{-1}(j')} < M$ and $\sum_{j'\in[j]} w_{\pi^{-1}(j')} \geq M$. This implies that when there is a single type of consumer, without loss of generality, we can assume that in the optimal solution of the ranking problem, product $n+1$ with weight $\bar{w}$ is placed in the last position that the consumer screens. If the optimal solution to the rank optimization problem (Ms) includes $\bar{w}$, then moving it to the last position does not change the consideration set. We argue that the optimal solution always includes $\bar{w}$. Contrary to our claim, suppose that the optimal solution does not include $\bar{w}$. Then, when the product in the last screened position is replaced with $\bar{w}$, the size of the consideration set does not change, and the market share increases.

Having shown that, without loss of generality, in the optimal solution, we can place product $n+1$ (with weight $\bar{w}$) in the last position that the consumer screens. Then, in light of the stopping rule in (6), we conclude that the rank optimization problem can be written as the following optimization problem: $\max_{A\subseteq[n]} w(A)$ subject to $w(A) < M$. Here, A can be seen as the set $\mathcal{C}_\pi \setminus \{n+1\}$. Specifically, let A^* be the optimal solution of this optimization problem. Then, we put the products in set A^* in the first $|A^*|$ positions, and we place the product with weight $\bar{w}$ in position $|A^*|+1$. The fact that the rank optimization problem can be written as $\max_{A\subseteq[n]} w(A)$ subject to $w(A) < M$, which is a general c-bounded uniform Knapsack problem as stated in Definition EC.1, shows that this problem is NP-complete.

EC.7. Proof of Theorem 3

Consider a consumer of type $k \in [K]$. Let J_w and J_o be the positions at which the consumer stops under the w-ordered algorithm and the optimal ranking, respectively. Recall that under any ranking π, the consumer screens positions downward and stops her search the first time that the stopping condition in (6) holds. Let x_j, $j \in [J_w]$, be the weight of the product in position j under the w-ordered algorithm. Similarly, let y_j, $j \in [J_o]$, be the weight of the product in position j under the optimal ranking. By Equation (3), the platform's market share—which is equivalent to the purchase probability of the consumer—under the w-ordered algorithm and the optimal ranking is respectively $\frac{\sum_{j\in[J_w]} x_j}{1+\sum_{j\in[J_w]} x_j}$ and $\frac{\sum_{j\in[J_o]} y_j}{1+\sum_{j\in[J_o]} y_j}$. We will show that $2\sum_{j\in[J_w]} x_j \geq \sum_{j\in[J_o]} y_j$. This gives us the desired results because for any $z > 0$, we have

$$\frac{2z}{2z+1} \leq \frac{2z}{z+1} \quad \text{and} \quad \frac{2z}{2z+1} - \frac{z}{z+1} \leq \frac{\sqrt{2}}{(\sqrt{2}+2)(\sqrt{2}+1)}\,.$$

To see why the second inequality holds, note that $\arg\max_{z>0}\left\{\frac{2z}{2z+1} - \frac{z}{z+1}\right\} = \frac{1}{\sqrt{2}}$.

We now proceed to verify $2\sum_{j\in[J_w]} x_j \geq \sum_{j\in[J_o]} y_j$. First note that since the w-ordered algorithm puts the products in decreasing order of their weights, it is not possible for another ranking to stop the consumer before position J_w (existence of such a ranking would imply that there exists a subset of fewer than J_w products whose sum of weights is larger than $\sum_{j\in[J_w]} x_j$, which cannot happen). Therefore, $J_o \geq J_w$. Now we consider two cases separately: $J_o = J_w$ and $J_o > J_w$. In the former case, by definition of w-ordered, we have $\sum_{j\in[J_w]} x_j \geq \sum_{j\in[J_o]} y_j$, which implies $2\sum_{j\in[J_w]} x_j \geq \sum_{j\in[J_o]} y_j$. Thus, we focus on the case of $J_o > J_w$.

By the stopping rule presented in (6), under the w-ordered algorithm, we have

$$\sum_{j\in[J_w]} x_j \geq r^k_{J_w+1}\,. \tag{EC.7}$$

Similarly, we get

$$\sum_{j\in[J_o-1]} y_j < r^k_{J_o} \leq r^k_{J_w+1}\,,$$

where the second inequality holds because $J_o < J_w$, and hence, $r^k_{J_o} \leq r^k_{J_w+1}$. Then, by Equation (EC.7), we get

$$\sum_{j\in[J_w]} x_j \geq \sum_{j\in[J_o-1]} y_j\,.$$

Observe that we have $y_{J_o} \leq x_1$ and $\sum_{j\in[J_w]} x_j \geq x_1$. The former holds because x_1 has the highest weight among all products. This implies that

$$2\sum_{j\in[J_w]} x_j \geq \sum_{j\in[J_o]} y_j\,.$$

The above equation completes the proof by using the claim above for all consumer types.

EC.8. Proof of Claim 1

The proof has two parts. In the first part, we show that the market share obtained by the ranking of Algorithm 1, denoted by ALG, is at least $(1-\epsilon)$ fraction of the maximum market share Ms; that is, $\mathsf{ALG} \geq (1-\epsilon)\mathsf{Ms}$. In the second part, we characterize the computational complexity of our DP.

First part: Let $\pi_{\rm opt}$ be the optimal ranking with actual weights and $\pi_{\rm alg}$ be the ranking returned by Algorithm 1. We note that Algorithm 1's search space is the class of LWP rankings. Thus, by optimality of $\pi_{\rm alg}$ within the LWP class, for any ranking π that belongs to this class, we have

$$\sum_{k\in[K]} \theta^k \frac{\tilde{w}(\mathcal{C}^k_{\pi_{\rm alg}})}{1+\tilde{w}(\mathcal{C}^k_{\pi_{\rm alg}})} \geq \sum_{k\in[K]} \theta^k \frac{\tilde{w}(\mathcal{C}^k_{\pi})}{1+\tilde{w}(\mathcal{C}^k_{\pi})}\,, \tag{EC.8}$$

where consideration sets are derived based on Equation (6) using the original weights, and not the rounded ones. (Recall that in (9), we use actual weights in determining stopping decisions.) If ranking $\pi_{\rm opt}$ belongs to the aforementioned class, the above equation holds when π is replaced with $\pi_{\rm opt}$. Now, assume that $\pi_{\rm opt}$ does not belong to that class. In that case, we construct an auxiliary ranking denoted by $\pi_{\rm aux}$ that has the same bucket representation as ranking $\pi_{\rm opt}$. However, unlike ranking $\pi_{\rm opt}$, ranking $\pi_{\rm aux}$ belongs to the LWP class. (Namely, in $\pi_{\rm aux}$, within the positions of a given bucket, products are internally sorted in the increasing order of their weights.) Here, bucket representation of a ranking π is a vector whose j-th element is the bucket of product $\pi(j)$, i.e., the product placed in position j under ranking π.

To make this definition clear, let us provide an example. Assume that we have 6 products with weights $w_1 < w_2 < \ldots < w_6$. Suppose that we have two buckets, where products $1,2,3$ belong to bucket 1 and products $4,5,6$ belong to bucket 2. Suppose that $\pi_{\rm opt} = (5,6,1,3,4,2)$. Obviously, this ranking does not belong to the LWP class because products 4 and 6 belong to the same bucket, but product 6, which has a higher weight than product 4, appears first in this ranking. The bucket representation of $\pi_{\rm opt}$ is given by $(2,2,1,1,2,1)$. Now define ranking $\pi_{\rm aux} = (4,5,1,2,6,3)$. Ranking $\pi_{\rm aux}$ belongs to the LWP class while having the same bucket representation. We point out that when faced with ranking $\pi_{\rm aux}$, consumers of each type k screen more products compared to when faced with ranking $\pi_{\rm opt}$. This holds because if the stopping condition, given by (6), holds in a position j under ranking $\pi_{\rm aux}$, it must hold under ranking $\pi_{\rm opt}$ as well. Here, we crucially use the LWP property of $\pi_{\rm aux}$. As a result, we have

$$\sum_{k\in[K]} \theta^k \frac{\tilde{w}(\mathcal{C}^k_{\pi_{\rm aux}})}{1+\tilde{w}(\mathcal{C}^k_{\pi_{\rm aux}})} \geq \sum_{k\in[K]} \theta^k \frac{\tilde{w}(\mathcal{C}^k_{\pi_{\rm opt}})}{1+\tilde{w}(\mathcal{C}^k_{\pi_{\rm opt}})}\,. \tag{EC.9}$$

By Inequalities (EC.8) and (EC.9), we have

$$\sum_{k\in[K]} \theta^k \frac{\tilde{w}(\mathcal{C}^k_{\pi_{\rm alg}})}{1+\tilde{w}(\mathcal{C}^k_{\pi_{\rm alg}})} \geq \sum_{k\in[K]} \theta^k \frac{\tilde{w}(\mathcal{C}^k_{\pi_{\rm opt}})}{1+\tilde{w}(\mathcal{C}^k_{\pi_{\rm opt}})} \geq \sum_{k\in[K]} \theta^k \frac{w(\mathcal{C}^k_{\pi_{\rm opt}})}{1+w(\mathcal{C}^k_{\pi_{\rm opt}})} = \mathsf{Ms}\,, \tag{EC.10}$$

where the second inequality holds because $\tilde{w}_i \geq w_i$ for any $i \in [n]$. The proof is completed by recalling that $\tilde{w}_i \leq (1+\epsilon)w_i$ and observing

$$\sum_{k\in[K]} \theta^k \frac{\tilde{w}(\mathcal{C}^k_{\pi_{\text{alg}}})}{1+\tilde{w}(\mathcal{C}^k_{\pi_{\text{alg}}})} \leq (1+\epsilon) \sum_{k\in[K]} \theta^k \frac{w(\mathcal{C}^k_{\pi_{\text{alg}}})}{1+w(\mathcal{C}^k_{\pi_{\text{alg}}})} = (1+\epsilon)\mathsf{ALG}.$$

Thus, we have $\mathsf{ALG} \geq \frac{1}{1+\epsilon}\mathsf{Ms} \geq (1-\epsilon)\mathsf{Ms}$.

Second part: We now discuss the computational complexity of our DP. Note that our state $\mathcal{S}$ has two components $\mathcal{B}$ and κ. $\mathcal{B}$ is a B dimensional vector where each element takes integer values between 0 and n, whereas κ is an integer number between 0 and K. Solving each step of our DP, given in (10), requires searching over at most B buckets, where computing the per-step reward for each bucket has complexity of $O(K)$. Thus, the complexity of Algorithm 1 is at most $O\Big(K^2\Big(\log(\rho)/\epsilon+1\Big)n^{\log(\rho)/\epsilon+1}\Big)$.

EC.9. Proof of Proposition 1

Similar to the proof of Theorem 2, we will reduce a general instance of the c-bounded uniform knapsack problem, as stated in Definition EC.1, to our problem.

We create an instance of the rank optimization problem with a set of $n+1$ products denoted by P and one consumer type such that for any $j \in [n+1]$, we have $r_j = M$. More precisely, the search cost, $s_j = s$, and the distribution f_W are such that we get $r_j = M$, where M equals K which denotes the capacity of the knapsack in Definition EC.1.(Note that to simplify notation, we drop the superscript k when we have one type.) For any item with size w_i in the knapsack problem, we have a product with the same weight w_i in set P. We also have a $n+1$-st product as the product with the highest weight in P. The weight of this product $w_{n+1} = \bar{w}$ where we define $\bar{w} := 2\max_{i\in[n]} w_i$. Further, we define

$$\tilde{\rho} := \frac{\bar{w}}{\min_{i\in[n]} w_i}. \tag{EC.11}$$

We assume that weights are drawn from distribution f_W. Under this distribution, for any $i \in [n]$, $f_W(w_i) = \frac{\delta}{n+1}$, where $\delta \in (0,1)$ will be defined later. Further, $f_W(\bar{w}) = \frac{\delta}{n+1}$ and $f_W(\bar{w}/\rho) = 1-\delta$, where $\rho > \tilde{\rho} > 1$ will also be defined later in the proof.

Following the same setting and notation in the proof of Theorem 2, we will show that for the considered instance, the rank optimization problem in (Wel) is equivalent to the following optimization problem: $\max_{A\subseteq[n]} w(A)$ subject to $w(A) < M$, which is the knapsack instance as stated in Definition EC.1. This will complete the proof. Here, $w(A) = \sum_{i\in A} w_i$.

If the search costs were all zero, establishing the equivalency would have been identical to that of the proof of Theorem 2. This is because both objectives of market share and consumer welfare are increasing functions of the summation of the weights of the products in the consumer's consideration set. However, the costs cannot be exactly zero because reservation prices are finite. Recall that we assume $r_j = M$ for $j \in [n+1]$. In the following, we show that there exists a small enough search cost s and a distribution f_W, parameterized by δ and ρ, such that: (1) we have $r_j = M$ for $j \in [n+1]$ and (2) maximizing consumer welfare is equivalent to maximizing the summation of the weights of the products in the consumer's consideration set. This, in turn, implies that solving the ranking problem is equivalent to solving the knapsack one. To establish the existence of parameters ρ and δ, in the following claim, we present an upper bound on the search cost.

Claim EC.1 **(Upper Bounding Search Cost)**. *Suppose $r_j = M$ for $j \in [n+1]$. If the search cost s satisfies the following condition*

$$s \leq \min_{m \leq \tilde{\rho}} \left\{ \frac{1}{m} \log \left(1 + \frac{m \frac{\bar{w}}{\tilde{\rho}}}{1 + \bar{w} + M} \right) \right\}. \tag{EC.12}$$

then maximizing consumer welfare is equivalent to maximizing the summation of the weights of the products in the consumer's consideration set. Here, $\tilde{\rho}$ is defined in (EC.11) *and the minimum is taken over positive integers, i.e., $m \in \{1, 2, \ldots, \lfloor \tilde{\rho} \rfloor\}$.*

Proof: Similar to the proof of Theorem 2, without loss of generality, we can assume that the optimal ranking puts product $\bar{w}$ in the last position because there is only a single type of consumer. Next, we aim to show that if Condition (EC.12) holds then the consumer benefits from screening as many products as possible before seeing $\bar{w}$. To that end, suppose that the consumer has already screened the products in set A; that is, $w(A) < M$, and she will stop if we place $\bar{w}$ next, i.e., $\bar{w} + w(A) \geq M$. Further, suppose that there exists another set A' with size m (i.e., $|A'| = m$) that can be screened by the consumer; that is, $w(A \cup A') < M$ with $A \cap A' = \emptyset$. In the rest of the proof we show if Condition (EC.12) holds then consumer welfare under the latter consideration set is always greater than that under the former. Note that when the consumer screens $A \cup A'$ before seeing $\bar{w}$, she pays an extra search cost of ms. Thus, we need to show that if Condition (EC.12) holds

$$\log(1 + \bar{w} + w(A \cup A')) - \log(1 + \bar{w} + w(A)) - ms \geq 0 \tag{EC.13}$$

The above inequality is equivalent to

$$s \leq \frac{1}{m} \log \left(1 + \frac{w(A')}{1 + \bar{w} + w(A)} \right) \tag{EC.14}$$

In order to show that (EC.14) holds under Condition (EC.12), we make the following observations:

1. $m \leq \tilde{\rho}$. (Recall that $m = |A'|$.) This is so because $w(A') \geq |A'| \cdot \min_{i \in [n]} w_i$, $w(A') \leq \bar{w}$, and by definition, $\tilde{\rho} = \frac{\bar{w}}{\min_{i \in [n]} w_i}$.

2. $w(A') \geq m\frac{\bar{w}}{\tilde{\rho}}$. This follows from the definition of $\bar{w}$ and $\tilde{\rho}$. (Recall that $\tilde{\rho} = \frac{\bar{w}}{\min_{i\in[n]} w_i}$ and $w(A') \geq m \cdot \min_{i\in[n]} w_i = m\frac{\bar{w}}{\tilde{\rho}}$)

3. $\bar{w} + w(A) < \bar{w} + M$. This holds $w(A \cup A') < M$, and hence $w(A) < M$.

Putting these observations together we have:

$$\frac{1}{m}\log\left(1 + \frac{w(A')}{1+\bar{w}+w(A)}\right) \geq \min_{m\leq\tilde{\rho}}\left\{\frac{1}{m}\log\left(1+\frac{m\frac{\bar{w}}{\tilde{\rho}}}{1+\bar{w}+M}\right)\right\} \tag{EC.15}$$

Observe that the above inequality implies that if Condition (EC.12) holds then (EC.13) also holds. This completes the proof of the claim. □

Having established an upper bound on search cost s, we proceed to define the parameters ρ and δ of the distribution f_W such that we have $r_j = M$ for $j \in [n+1]$ and the resulting search cost satisfies Condition (EC.12). Let us defined λ as follows:

$$\lambda := \min_{m\leq\tilde{\rho}}\left\{\frac{1}{m}\log\left(1+\frac{m\frac{\bar{w}}{\tilde{\rho}}}{1+\bar{w}+M}\right)\right\}. \tag{EC.16}$$

Note that λ is finite, positive, and independent of n. Further, we remark that λ can be computed in polynomial time because it is a minimum of a finite number of alternatives (recall that $\tilde{\rho}$ is finite and m is an integer. Next, we define ρ as follows:

$$\rho := \max\{\tilde{\rho}+1, \rho_1\} \quad \text{where } \rho_1 \text{ solves} \quad \log(1+\frac{\bar{w}/\rho_1}{1+M}) = \lambda/2 \tag{EC.17}$$

Note that such a ρ_1 must exists because $\log(1+\frac{\bar{w}/x}{1+M})$ is continuous and strictly decreasing in x, and $\lim_{x\to\infty}\log(1+\frac{\bar{w}/x}{1+M}) = 0$. Having defined ρ, we define

$$\delta := \frac{\lambda}{2\rho} \tag{EC.18}$$

In the following claim, we show that the resulting distribution has the desired properties.

Claim EC.2 **(Properties of f_W)**. *Suppose that $r_j = M$ for $j \in [n+1]$, and $f_W(w_i) = \frac{\delta}{n+1}$ for $i \in [n+1]$ and $f_W(\bar{w}/\rho) = 1-\delta$, where ρ and δ are defined in* (EC.17) *and* (EC.18)*. Then,*

$$s = \mathbb{E}\left[\log(1+\frac{W}{1+M})\right] \leq \lambda = \min_{m\leq\tilde{\rho}}\left\{\frac{1}{m}\log\left(1+\frac{m\frac{\bar{w}}{\tilde{\rho}}}{1+\bar{w}+M}\right)\right\}.$$

Proof: First note that by the definition of reservation price and the fact that $r_j = M$, we have $s_j = s = \mathbb{E}[\log(1+\frac{W}{1+M})]$, for $j \in [n+1]$. Our goal here is to show $s \leq \lambda$. To do so, we express $s = \mathbb{E}[\log(1+\frac{W}{1+M})]$ as follows

$$\mathbb{E}\left[\log(1+\frac{W}{1+M})\right] = \log(1+\frac{\bar{w}/\rho}{1+M}) + \zeta. \tag{EC.19}$$

where we define ζ as

$$\zeta := \frac{\delta}{n+1}\sum_{i\in[n+1]}\Big(\log(1+\frac{w_i}{1+M}) - \log(1+\frac{\bar{w}/\rho}{1+M})\Big).$$

The first term in Equation (EC.19), i.e., $\log(1+\frac{\bar{w}/\rho}{1+M})$ is less than or equal to $\lambda/2$. To see why recall that $\rho = \max\{\tilde{\rho}+1, \rho_1\}$. Then, if $\rho = \rho_1$, by definition of ρ_1, $\log(1+\frac{\bar{w}/\rho}{1+M}) = \lambda/2$. If $\rho = \tilde{\rho}+1$, then because $\log(1+\frac{\bar{w}/x}{1+M})$ is decreasing in x, we have $\log(1+\frac{\bar{w}/\rho}{1+M}) \le \log(1+\frac{\bar{w}/\rho_1}{1+M}) = \lambda/2$. Next, we show that the second term in Equation (EC.19), i.e., ζ, lies in the interval of $(0, \lambda/4]$. This completes the proof. Note that $\zeta > 0$, because for any $i \in [n+1]$, $w_i > \bar{w}/\rho$. (Recall that $\frac{\bar{w}}{w_i} \le \tilde{\rho} < \rho$.) First observe that $\rho > \tilde{\rho}$. Next, we show that $\zeta \le \lambda/4$. Note that utilizing $w_i > \bar{w}/\rho$, we have:

$$\zeta \le \frac{\delta}{n+1}\sum_{i\in[n+1]}\frac{\rho w_i}{\bar{w}} \le \frac{\delta}{n+1}\times(n+1)\frac{\rho}{2} = \frac{\delta\rho}{2} = \frac{\lambda}{4}, \tag{EC.20}$$

where the first inequality follows from the algebraic observation that for any $x > y > 0$ and $z > 0$, we have: $\log(\frac{z+x}{z+y}) \le \log(\frac{x}{y}) \le \frac{x}{y}$. The second inequality follows from the definition of $\bar{w}$: $w_i \le \max_{i\in[n]} w_i = \bar{w}/2$. Finally the last one simply follows from the definition of δ (given in (EC.18)). □

Putting Claims EC.1 and EC.2 together completes the proof of Proposition 1.

EC.10. Proof of Proposition 2

The basic idea of the proof is similar to that of Theorem 3. In particular, fix a consumer type k and define J_w and J_o be the position at which the consumer stops under the w-ordered ranking and the optimal welfare maximizing ranking, respectively. Also, define x_j, $j \in [J_w]$ and y_j, $j \in [J_o]$ in a similar way as we did in the proof of Theorem 3. Following the same steps, we can show that:

$$2\sum_{j\in[J_w]} x_j \ge \sum_{j\in[J_o]} y_j\,. \tag{EC.21}$$

Under the w-ordered ranking, consumer search cost will be always less than or equal to that under the optimal ranking. This is because $J_w \le J_o$ for the same reason explained in the proof of Theorem 3. Therefore, we have

$$\begin{aligned}\mathsf{Wel}_w - \mathsf{Wel} &= \log(1+\sum_{j\in[J_w]} x_j) + \gamma - \sum_{j\in[J_w]} s_j^k - \Big[\log(1+\sum_{j\in[J_o]} y_j) + \gamma - \sum_{j\in[J_o]} s_j^k\Big]\\ &\ge \log(1+\sum_{j\in[J_w]} x_j) - \log(1+\sum_{j\in[J_o]} y_j)\end{aligned} \tag{EC.22}$$

With inequalities (EC.21) and (EC.22), the proof is completed by noting that for any $z > 0$, we have

$$\log(1+z) \ge \log(1+2z) - \log(2)\,.$$

To see why the second inequality holds, note that $\log(1+2z) - \log(1+z) = \log(\frac{1+2z}{1+z})$ is increasing in $z \ge 0$; and as z approaches ∞, it approaches $\log(2)$.

EC.11. Proof of Claim 2

The proof has two parts. In the first part, we show that the consumer welfare obtained by the ranking of Algorithm 2, denoted by ALG, is at most ϵ away from the maximum consumer welfare Wel; that is, $\mathsf{ALG} \geq \mathsf{Wel} - \epsilon$. In the second part, we argue that for a fixed ϵ, the computational complexity of Algorithm 2 is polynomial in n.

First part: Let S_M be the ordered set that contains the first M products under the optimal ranking, denoted by π_{opt}. We construct a ranking using the ordered set S_M. To do so, we follow the steps in Algorithm 2: for any $i \in [M]$, we place the i^{th} element of the ordered set S_M in position i. Then, we fill the next $n - M$ positions by ranking the remaining products, i.e., $[n]\backslash S_M$, in decreasing order of their weights. Let us call this ranking π_M. Further, define π_{alg} as the ranking returned by Algorithm 2. With a slight abuse of notation, let $\mathsf{Wel}(\pi)$ be the consumer welfare under ranking π. Then, by optimality of π_{alg}, we have

$$\mathsf{ALG} = \mathsf{Wel}(\pi_{\text{alg}}) \geq \mathsf{Wel}(\pi_M)\,.$$

We will show that $\mathsf{Wel}(\pi_M) \geq \mathsf{Wel}(\pi_{\text{opt}}) - \epsilon$. This gives us the desired result. Define $\mathsf{Wel}^k(\pi)$ as the welfare of consumers of type k under ranking π. Then, if the size of the consideration set of a consumer of type k under optimal ranking π_{opt} is less than or equal to M, i.e., $|\mathcal{C}^k_{\pi_{\text{opt}}}| \leq M$, we have $\mathsf{Wel}^k(\pi_{\text{opt}}) = \mathsf{Wel}^k(\pi_M)$. Thus, let us consider any consumer of type k that screens more than M positions under the optimal ranking π_{opt}. Note that by the stopping condition (6), if $|\mathcal{C}^k_{\pi_{\text{opt}}}| > M$, then $|\mathcal{C}^k_{\pi_M}| > M$. Let J_m and J_o be the positions at which the consumer stops under ranking π_M and π_{opt}, respectively. Since under π_M, the products in position $M+1,\ldots,n$ are ranked in decreasing order of their weights, similar to the proof of Theorem 3, we can show that $J_m \leq J_o$. Let x_j, $j \in [J_m]$, be the weight of the product in position j under ranking π_M. Similarly, let y_j, $j \in [J_o]$, be the weight of the product in position j under ranking π_{opt}. First note that in order show $\mathsf{Wel}^k(\pi_M) - \mathsf{Wel}^k(\pi_{\text{opt}}) \geq -\epsilon$, it suffices to show

$$(1+\epsilon)\sum_{j\in[J_m]} x_j \geq \sum_{j\in[J_o]} y_j\,. \tag{EC.23}$$

This is so because if (EC.23) holds, then we have

$$\begin{aligned}
\mathsf{Wel}^k(\pi_M) - \mathsf{Wel}^k(\pi_{\text{opt}}) &= \Big[\log(1+\sum_{j\in[J_m]} x_j) + \gamma - \sum_{j\in[J_m]} s_j^k\Big] - \Big[\log(1+\sum_{j\in[J_o]} y_j) + \gamma - \sum_{j\in[J_o]} s_j^k\Big] \\
&\geq \log(1+\sum_{j\in[J_m]} x_j) - \log(1+\sum_{j\in[J_o]} y_j) \\
&\geq \log\Big(1+\frac{\sum_{j\in[J_o]} y_j}{1+\epsilon}\Big) - \log(1+\sum_{j\in[J_o]} y_j) \\
&= \log\Big(\frac{1+\epsilon+\sum_{j\in[J_o]} y_j}{(1+\epsilon)(1+\sum_{j\in[J_o]} y_j)}\Big) \geq -\epsilon\,,
\end{aligned}$$

where the first inequality follows from $J_m \le J_o$, the second holds if (EC.23) is satisfied, and the third one holds because function $\log\left(\frac{1+\epsilon+z}{1+\epsilon+(1+\epsilon)z}\right)$ is decreasing for $z \ge 0$ and $\lim_{z\to\infty} \log\left(\frac{1+\epsilon+z}{1+\epsilon+(1+\epsilon)z}\right) = \log\left(\frac{1}{1+\epsilon}\right) \ge -\epsilon$. Thus, we are left to show that (EC.23) holds.

We consider the following two cases. In the first case, $J_m = J_o$. Under this case, (EC.23) holds because by construction of π_M, we have $\sum_{j\in[J_m]} x_j \ge \sum_{j\in[J_o]} y_j$. In the second case, $J_m < J_o$. Note that because under ranking π_M, the consumer of type k stops at J_m, we must have: $\sum_{j\in[J_m]} x_j \ge r^k_{J_m+1}$. On the other hand, because under ranking π_{opt}, the consumer of type k does not stop J_m, we have $\sum_{j\in[J_o]} y_j - \max_{j\in[J_o]}\{y_j\} < r^k_{J_m+1}$. Putting these two together, we have:

$$\sum_{j\in[J_m]} x_j \ge \sum_{j\in[J_o]} y_j - \max_{j\in[J_o]}\{y_j\}\,.$$

Further, note that $\max_{j\in[J_o]}\{y_j\} \le w_{\max}$. Thus, we have

$$\sum_{j\in[J_m]} x_j + w_{\max} \ge \sum_{j\in[J_m]} x_j + \max_{j\in[J_o]}\{y_j\} \ge \sum_{j\in[J_o]} y_j\,.$$

In order to complete the proof of (EC.23), it suffices to show that $w_{\max} \le \frac{\rho}{M}\sum_{j\in[J_m]} x_j$. This is true because

$$\sum_{j\in[J_m]} x_j \ge M w_{\min} = \frac{M}{\rho} w_{\max}\,,$$

where the inequality holds because $J_m \ge M$. Further, we use the definition of $\rho = \frac{w_{\max}}{w_{\min}}$.

Second part: The number of possible ordered sets S is $\frac{n!}{(n-M)!} = O(n^M)$. Fixing an ordered set S, constructing the ranking and computing its consumer welfare has complexity of $O(Kn)$; thus, the complexity is $O(Kn^{1+\rho/\epsilon})$.

EC.12. Tightness of our Analysis in Theorem 3

EXAMPLE EC.1. Suppose that we have 2 positions and 2 products with weights $w_1 = x$ and $w_2 = x(1-2\epsilon)$ (where $x, \epsilon \in \mathbb{R}^+$ and $\epsilon < 1/2$.) Further, the number of consumer types, K, is one. Suppose that $s_1 = 0$ and thus $r_1 = \infty$. Further, suppose that s_2 and the distribution of weights are such that $r_2 = x(1-\epsilon)$.

The optimal solution is to first place w_2 in position 1 and then w_1 in position 2, i.e., the optimal ranking is $(2,1)$. Following the index-based optimal policy, a consumer facing ranking $(2,1)$ will screen both positions, and thus, the market share corresponding to this ranking will be

$$\mathsf{Ms} = \frac{2x(1-\epsilon)}{1+2x(1-\epsilon)}\,.$$

In contrast, for the w-ordered ranking, given by ranking $(1,2)$, the consumer only screens the first position. This is because the weight of the product placed in the first position is larger than the reservation price for the second position, i.e., r_2. Consequently, the market share of w-ordered ranking will be

$$\mathsf{Ms}_w = \frac{x}{1+x}\,.$$

The ratio $\mathsf{Ms}_w/\mathsf{Ms}$ achieves its minimum when $\epsilon \to 0$ and $x \to 0$, and we have

$$\lim_{x\to 0}\lim_{\epsilon\to 0}\frac{\mathsf{Ms}_w}{\mathsf{Ms}} = \frac{1}{2}\,.$$

In contrast, the gap $\mathsf{Ms}_w - \mathsf{Ms}$ achieves its minimum when $\epsilon \to 0$ and $x = 1/\sqrt{2}$. The minimum is $-\frac{\sqrt{2}}{(\sqrt{2}+2)(\sqrt{2}+1)}$. □

EC.13. Tightness of our Analysis in Proposition 2

Example EC.2. Suppose that we have 2 positions and 2 products with weights $w_1 = x$ and $w_2 = x(1-2\epsilon)$ (where $x, \epsilon \in \mathbb{R}^+$ and $\epsilon < 1/2$.) Further, the number of consumer types, K, is one. Suppose that the distribution of weights is given by

$$W = \begin{cases} x & \text{w.p. } \delta \\ x(1-2\epsilon) & \text{w.p. } \delta \\ \zeta & \text{w.p. } 1-2\delta, \end{cases}$$

where δ and ζ are small positive numbers. Further, we set $s_1 = 0$, which means $r_1 = \infty$. We set the search cost for the second position to be

$$s_2 = \mathbb{E}\Big[\log\Big(1+\frac{W}{1+r_2}\Big)\Big] = \delta\Big(\log\Big(1+\frac{x}{1+r_2}\Big) + \log\Big(1+\frac{x(1-2\epsilon)}{1+r_2}\Big)\Big) + (1-2\delta)\log\Big(1+\frac{\zeta}{1+r_2}\Big),$$

where $r_2 := x(1-\epsilon)$. Note that r_2 is the reservation price for the second position. For the above instance, we study the consumer welfare under the w-ordered and optimal rankings. First note that under the w-ordered ranking, given by ranking $(1,2)$, the consumer only screens the first position. This is because the weight of the product placed in the first position is larger than the reservation price for the second position. Consequently, we have

$$\mathsf{Wel}_w = \log(1+x) + \gamma\,.$$

Next, let us consider the consumer welfare under the alternative ranking $(2,1)$. Here, the consumer screens both products. Thus, her welfare will be

$$\log(1+2x(1-\epsilon)) + \gamma - s_2\,.$$

Now, if $\delta \to 0$ and $\zeta \to 0$, then we have $s_2 \to 0$ for any $x \in \mathbb{R}^+$. Thus, ranking $(2,1)$ will be optimal. Further as $\epsilon \to 0$ and $x \to \infty$, we have $\lim_{x\to\infty}\lim_{\delta,\zeta,\epsilon\to 0}\Big(\mathsf{Wel}_w - \mathsf{Wel}\Big) = -\log(2)$.

□

EC.14. An Estimation Procedure

In this section, we describe a maximum likelihood procedure for estimating reservation prices and consumer type probabilities. Each data point represents a search query, where the t-th search query is associated with a list of n products with weights $(w_{1,t}, \ldots, w_{n,t})$. The list of products varies across queries, and we assume the platform's set of products—from which it chooses n products for each query—is diverse, covering the entire range supported by the weight distribution f_W which we assume to be atomless and known by the platform. We remark that when focusing on the objective of market share, it only suffices to estimate the reservation prices. However, once we estimate the reservation prices, the search costs—needed for the objective of consumer welfare—can be estimated using their relation as specified in (5).

We estimate reservation prices and type probabilities using our stopping rules (6). According to this rule, in search query t and under ranking π, the consumer of type k stops at position j iff (i) $\mathbb{I}\{\sum_{j'\in[j-1]} w_{\pi(j'),t} < r_j^k\} = 1$ and (ii) $\mathbb{I}\{\sum_{j'\in[j]} w_{\pi(j'),t} \geq r_{j+1}^k\} = 1$. Observe that this stopping rule is non-continuous, and this makes the estimation process of reservation prices challenging. To overcome this challenge, we smooth out the stopping rule by adding an independent Gaussian noise to the weight of the product placed in the first position. Namely, we assume that the consumer stops at position j if $\mathbb{I}\{\epsilon + \sum_{j'\in[j-1]} w_{\pi(j'),t} < r_j^k\}\mathbb{I}\{\epsilon + \sum_{j'\in[j]} w_{\pi(j'),t} \geq r_{j+1}^k\} = 1$, where $\epsilon \sim N(0,\sigma)$ and $\sigma > 0$ is a small positive number. We highlight that the noise term is only added for estimation purposes; in the synthetic data, the consumer still follows our two-stage search model and stops at position j if $\mathbb{I}\{\sum_{j'\in[j-1]} w_{\pi(j'),t} < r_j^k\}\mathbb{I}\{\sum_{j'\in[j]} w_{\pi(j'),t} \geq r_{j+1}^k\} = 1$. Considering the added noise, in estimating reserve prices, we assume that the consumer stops at position j with probability

$$\Phi\Big(\frac{r_j^k - \sum_{j'\in[j-1]} w_{\pi(j'),t}}{\sigma}\Big) - \Phi\Big(\frac{r_{j+1}^k - \sum_{j'\in[j]} w_{\pi(j'),t}}{\sigma}\Big),$$

where Φ is the CDF of the standard normal distribution. Then, suppose that in search query t and under ranking π, the consumer stops at position $j_t = 2, \ldots, n-1$. The likelihood of this event can be written as

$$q_t := \sum_{k\in[K]} \theta^k \Big(\Phi\Big(\frac{r_{j_t}^k - \sum_{j'\in[j_t-1]} w_{\pi(j'),t}}{\sigma}\Big) - \Phi\Big(\frac{r_{j_t+1}^k - \sum_{j'\in[j_t]} w_{\pi(j'),t}}{\sigma}\Big)\Big),$$

Similar expressions can be obtained for $j_t = 1, n$. Then, to estimate reserve prices and type probabilities, we solve the following maximum log-likelihood optimization problem: $\max \sum_{t\in[T]} \log(q_t)$, where T is the number of search queries and maximization is over reservation prices and type probabilities with the constraints that (1) fixing the type, reservation prices decrease for lower positions, and (2) fixing the position, reservation prices increase for types with larger indices. We use the "fmincon" function in Matlab to carry out this estimation exercises.

Finally, we remark that under our model, consideration sets are deterministic for each type. Therefore, identifying the number of consumer types can be done prior to the estimation process: if we show the same ranking to a small subset of consumers, the number of resulting consideration sets will be a lower bound on the number of types (because multiple types can form the same consideration sets). By repeating this for a small number of rankings, we can determine the number of types.