The Value of Robust Assortment Optimization Under Ranking-based Choice Models

Bradley Sturt
Department of Information and Decision Sciences
University of Illinois at Chicago, bsturt@uic.edu

The ranking-based choice model is a popular model in revenue management for predicting demand for a firm’s products based on the assortment of products that the firm offers to their customers. Because this model has a huge number of parameters, many different ranking-based choice models can be consistent with the historical sales data generated by a firm’s past assortments. Motivated by the use of ranking-based choice models in assortment planning, we consider the following identification question: Is it possible to identify an assortment with an expected revenue that is strictly greater than the expected revenues of the firm’s past assortments under all ranking-based choice models that are consistent with the firm’s historical sales data?

In this work, we provide the first answers to the identification question by using robust optimization. We begin by characterizing the structure of optimal assortments for a class of robust assortment optimization problems proposed by Farias, Jagabathula, and Shah (2013). We then leverage this structure to develop the first algorithms for answering the identification question that run in polynomial time for any fixed number of past assortments. We use our algorithms to prove that it is possible to have affirmative answers to the identification question with as few as two past assortments, and that it is impossible to obtain affirmative answers when the past assortments are revenue-ordered. These findings, coupled with concise numerical experiments, reveal that considering the identification question can be essential for finding high-quality assortments from ranking-based choice models in high-stakes assortment planning problems.

Key words: assortment planning; robust optimization; nonparametric choice modeling.

History: This version: December 9, 2021.

“First, do no harm” - Hippocratic Oath

1. Introduction

The ranking-based choice model is one of the most fundamental and influential discrete choice models in revenue management. It is used by firms in a variety of industries (such as e-commerce and brick-and-mortar retail) to predict the demand for the firm’s products as a function of the subset of products that the firm offers to their customers. The popularity of the ranking-based choice model can be attributed to its generality: it can represent any random utility maximization model, and thus encompasses many other popular discrete choice models such as the multinomial logit model (Block and Marschak 1959, Farias et al. 2013). The ranking-based choice model posits that
customers who visit the firm have preferences that are represented by randomly-chosen rankings, and, based on the subset of products offered by the firm, each customer will purchase the product that is most preferred according to their personal ranking.

Unfortunately, estimating a ranking-based choice model from a firm’s historical sales data is notoriously challenging in practice. The key issue is that the ranking-based choice model is comprised of around \( n! \) parameters, where \( n \) is the number of product alternatives that a firm can elect to offer to their customers. Because the number of past assortments \( M \) that the firm has previously offered to their customers usually satisfies \( M \ll n! \), many selections of these parameters can yield a ranking-based choice model that is consistent (has low or zero prediction error) with the historical sales data generated by the firm’s past assortments. This leaves firms with the challenge of selecting \textit{which} ranking-based choice model, out of all of those that are consistent with their historical sales data, to use when making operational planning decisions.

The challenge of estimation in ranking-based choice models is particularly acute in the context of \textit{assortment planning}. Here, the typical goal of a firm is to identify a new subset of products (referred to as an \textit{assortment}) to offer to their customers in order to increase the firm’s expected revenue. Because the true relationship between assortments and expected revenue is unknown, firms will typically interpret an estimated ranking-based choice model as “ground truth” and subsequently solve an optimization problem to find an assortment which maximizes the “predicted” expected revenue (Aouad et al. 2018, Bertsimas and Mišić 2019, Honhon et al. 2012, van Ryzin and Vulcano 2015, 2017, Feldman et al. 2019, Aouad et al. 2021, Désir et al. 2021a). This widely-used technique for identifying a new assortment is referred to in the revenue management literature as \textit{estimate-then-optimize}. The estimate-then-optimize technique can be attractive from a computational standpoint, due to its decoupling of the combinatorial problems related to estimation and optimization. But the assortment which is optimal under one selection of a ranking-based choice model that is consistent with the historical sales data may be highly \textit{suboptimal} under another ranking-based choice model that is also consistent with the historical sales data. This raises concerns about whether estimate-then-optimize can be trusted to identify a new assortment for the firm with an expected revenue that, at the very least, is no less than the firm’s highest expected revenue from their past assortments.

The real-world consequences of implementing a low-quality assortment can be significant to firms. For example, when the Super Fresh grocery chain stopped carrying many of their low-selling dry grocery items, “their customers took their business elsewhere, and the retailer entered bankruptcy” (Fisher and Vaidyanathan 2012). Similarly, when Walmart rolled out significant changes to their product offerings in 2008, the new assortments resulted in a steep decline in sales and damaged
relationships between Walmart and their suppliers (Neff 2009). In fact, these negative outcomes were entirely avoidable, as Walmart eventually reinstated their previous assortments (Fisher and Vaidyanathan 2012). As these examples illustrate, it can be imperative in high-stakes assortment planning problems to have guarantees that assortments produced by algorithms will “first, do no harm” to the firm, following the code of ethics that is traditionally taught to medical students in the United States and United Kingdom (Smith 2005).

With the above motivation, we consider the following question: Is it possible to identify an assortment with an expected revenue that is strictly greater than the expected revenues of the firm’s past assortments under all of the ranking-based choice models that are consistent with the historical sales data generated by the firm’s past assortments? The answer to this question can have considerable practical value to firms. Indeed, an affirmative answer to this question implies that one can find a new assortment that can be trusted to improve the firm’s expected revenue in a way that is not exclusive to just one of the many ranking-based choice models that are consistent with the firm’s historical sales data. A negative answer to this question is also practically useful, as a negative answer warns firms that implementing an assortment found by estimate-then-optimize can result in lower expected revenues than maintaining the status quo. In the rest of this work, we will refer to the question that is italicized at the beginning of this paragraph as the identification question for ranking-based choice models.

Despite the practical value of the identification question, only limited progress has been made on answering it until now. To the best of our knowledge, the only prior work that is closely related to the identification question is that of Farias, Jagabathula, and Shah (2013, henceforth abbreviated as FJS13). FJS13 presents algorithms for computing the worst-case expected revenue of a fixed assortment under all ranking-based choice models that are consistent with the historical sales data. As we will see later on in §2, the problem of finding an assortment that maximizes this worst-case expected revenue turns out to be closely related to answering the identification question. However, no algorithms to date have been developed in the literature for solving such robust assortment optimization problems (Rusmevichientong and Topaloglu 2012, p.867), Jagabathula (2014, p.8)), and “it is not clear how one may formulate the problem of optimizing the worst-case revenue as an efficiently solvable mathematical optimization formulation.” (Mišić 2016, p.118). Nonetheless, we remark that algorithms and attractive structural results have been developed in recent years for other types of robust assortment optimization problems (Rusmevichientong and Topaloglu 2012, Bertsimas and Mišić 2017, Désir et al. 2021b, Wang et al. 2020). In fact, as a byproduct of answering the identification question, our work resolves the aforementioned gap in the literature by providing the first polynomial-time algorithms for solving robust assortment optimization problems under the data-driven uncertainty set proposed by FJS13.
In this work, we provide the first answers to the identification question by establishing a powerful structural result for a class of robust assortment optimization problems (Theorem 1 in §3). Stated succinctly, our theorem establishes the first characterization of the structure of optimal assortments for robust assortment optimization problems under the data-driven uncertainty set proposed by FJS13. Surprisingly, our theorem reveals that the optimal assortments for this class of robust assortment optimization problems have a simple and interpretable structure that is similar to the structure of the widely-studied class of revenue-ordered assortments. This simple structure is important because it allows us to drastically reduce the number of candidate assortments that need to be checked in order to answer the identification question. Our proof of Theorem 1 is based on an elementary (yet intricate) analysis of reachability conditions for vertices in a data-driven class of directed acyclic graphs.

Using our structural result, we proceed to establish the first answers to the identification question. We begin by developing an impossibility result for the case in which the collection of previously-offered assortments is equal to the widely-studied class of revenue-ordered assortments (Theorem 2 in §4.1). Specifically, when the historical sales data is generated from revenue-ordered assortments, our structural result reveals that it is impossible to have an affirmative answer to the identification question, regardless of the prices of the products and the observations of the historical sales data. To examine the practical importance of such an impossibility result, we perform numerical experiments in which we assess the performance of assortments obtained using the estimate-then-optimize technique when the historical sales data is randomly generated from revenue-ordered assortments. The results of our experiments are striking: in more than 98% of the problem instances in which the estimate-then-optimize technique recommended a new assortment, the worst-case decline in expected revenue from implementing the new assortment (relative to the expected revenue from the best past assortment) exceeded the best-case increase in expected revenue. The difference in magnitude between the worst-case and best-case change in expected revenue from implementing the new assortment found from estimate-then-optimize is also significant; the average best-case improvement of the new assortment over the best previously-offered assortment is 6.56%, while the average worst-case improvement of the new assortment over the best previously-offered assortment is -21.71%. Our results thus demonstrate that estimate-then-optimize can cause significant declines in a firm’s expected revenue, even in settings in which the firm has previously implemented the celebrated and widely-recommended class of revenue-ordered assortments.

In view of this impossibility result, we next establish that affirmative answers can indeed exist to the identification question. To establish this, we use our structural result to develop the first strongly polynomial-time algorithm for answering the identification question when the firm has
offered two past assortments (Theorem 3 in §4.2). Our algorithm obtains answers to the identification question by reducing the robust assortment optimization problem to a sequence of minimum-cost network flow problems, which can be readily solved using off-the-shelf optimization software, and our algorithm has a total running time of $O(n^5 \log(nr_n))$, where $r_n$ is the integral price of the most expensive product. Using our algorithm in numerical experiments on randomly generated problem instances, we prove that affirmative answers to the identification question can exist even in the seemingly limited case of two previous assortments. We refine the experimental findings to identify a simple example with four products in which the identification question has an affirmative answer. At the same time, we show that our algorithm is practically tractable, as the algorithm can answer the identification question in less than 30 seconds for problem instances with $n = 100$ products. These findings establish, for the first time, that there can exist tractable algorithms for answering the identification question which yield practical and actionable insights, even when a limited number of assortments have been offered in the past.

We conclude by investigating whether it can be computationally tractable to pursue answers to the identification question in general settings with more than two past assortments. Our main results here are positive; specifically, we establish the computational tractability of the identification question by developing an algorithm for answering it that runs in weakly polynomial time for any fixed number of past assortments (Theorem 4 in §4.3). These contributions thus provide evidence that it can be possible to develop practical algorithms for answering the identification question for real-world problem instances, where the composition of products in the previously-offered assortments do not exhibit any convenient structure and where there may be no ranking-based choice models that have zero prediction error on the historical sales data. As a byproduct of answering the identification question, we also obtain the first polynomial-time algorithm for solving robust assortment optimization problems under the data-driven uncertainty set proposed by FJS13 for any fixed number of past assortments.

Our paper is organized as follows. In §2, we present our problem setting and formally state the identification question. In §3, we develop our key structural result which characterizes the optimal solutions for robust assortment optimization problems under ranking-based choice models. In §4, we discuss three applications of our structural result to provide the first answers to the identification question. In §5, we offer concluding thoughts and directions for future research. For brevity, all lengthy proofs of intermediary results are found in the appendices.

**Notation and Terminology.** We use $\mathbb{R}$ to denote the real numbers, $\mathbb{R}_+$ to denote the nonnegative real numbers, and $y^\top x$ to denote the inner product of two vectors. We use the phrase ‘collection’ to refer to a set of sets. We let the set of all probability distributions which are supported on a
finite set $\mathcal{A}$ be denoted by $\Delta_{\mathcal{A}} \triangleq \{ \lambda : \sum_{a \in \mathcal{A}} \lambda_a = 1, \lambda_a \geq 0 \forall a \in \mathcal{A} \}$. We assume throughout that a norm $\| \cdot \|$ is either the $\ell_1$-norm or $\ell_\infty$-norm, and so it follows that optimization problems of the form $\min_{x,y} \{ c^T x + d^T y \mid Ax + By \leq b, \| y \| \leq \eta \}$ can be referred to as linear optimization problems. We let $\mathbb{I}\{ \cdot \}$ denote the indicator function, which equals one if $\cdot$ is true and equals zero otherwise.

**Code Availability.** The code for conducting the numerical experiments in this paper is freely available and can be accessed at https://github.com/brad-sturt/IdentificationQuestion.

## 2. Problem Setting and the Identification Question

We adopt the perspective of a firm that must select a subset of products to offer to their customers. Let the universe of products available to the firm be denoted by $\mathcal{N} \triangleq \{ 1, \ldots, n \}$, where the no-purchase option is denoted by index 0 and $\mathcal{N}_0 \triangleq \mathcal{N} \cup \{ 0 \}$. The revenue generated by selling one unit of product $i \in \mathcal{N}$ is represented by $r_i > 0$, and the revenue associated with the no-purchase option is $r_0 = 0$. An assortment is defined as any subset of products $S \subseteq \mathcal{N}_0$ that includes the no-purchase option, $0 \in S$, and we let $\mathcal{S} \triangleq \{ S \subseteq \mathcal{N}_0 : 0 \in S \}$ denote the collection of all assortments.

We study a problem setting in which the underlying relationship between assortment and customer demand is unknown, and our only information on this relationship comes from historical sales data generated by the firm’s previously-offered assortments. Let the previously-offered assortments be denoted by $\mathcal{M} \triangleq \{ S_1, \ldots, S_M \} \subseteq \mathcal{S}$, and let the indices of these past assortments be denoted by $\mathcal{M} \triangleq \{ 1, \ldots, M \}$. Unless stated otherwise, we will make no assumptions on the mechanism by which the firm selected the assortments to offer in the past. That is, the firm could have chosen the previously-offered assortments by drawing products randomly; alternatively, the previously-offered assortments could have been chosen using managerial intuition or some other systematic approach.

We assume that the firm offered each past assortment $S_m \in \mathcal{M}$ to their customers for a sufficient duration to obtain an accurate estimate of the purchase frequencies, *i.e.*, the fraction of customers $v_{m,i} \in [0, 1]$ that purchase product $i \in S_m$ when offered assortment $S_m$. This historical sales data is assumed to be normalized such that $\sum_{i \in \mathcal{N}_0} v_{m,i} = 1$, and the purchase frequencies for products that are not in an assortment are defined equal to zero, that is, $v_{m,i} = 0$ for all $i \notin S_m$ and $m \in \mathcal{M}$.

A discrete choice model is a function that predicts purchase frequencies for the firm based on the assortment that the firm offers to their customers. A *ranking-based choice model* is a type of choice model which is parameterized by a probability distribution $\lambda$ over the set of all distinct rankings of the products, where a ranking refers to a one-to-one mapping of the form $\sigma : \{ 0, \ldots, n \} \to \{ 0, \ldots, n \}$. Specifically, a ranking $\sigma$ encodes a preference for product $i$ over product $j$ if and only if $\sigma(i) < \sigma(j)$. Let the set of all distinct rankings over the products be denoted by $\Sigma$, and we readily observe that the number of distinct rankings in this set satisfies $|\Sigma| = (n+1)!$. Given a probability distribution
over rankings \( \lambda \in \Delta_\Sigma \) and an assortment \( S \in \mathcal{S} \), the prediction made by the ranking-based choice model for the purchase frequency of each product \( i \in \mathcal{N}_0 \) is given by

\[
\mathcal{D}_i^\lambda(S) \triangleq \sum_{\sigma \in \Sigma} I\left\{ i = \arg \min_{j \in S} \sigma(j) \right\} \lambda_\sigma.
\]

It is straightforward to see from the above definition that a ranking-based choice model always satisfies the equality \( \mathcal{D}_i^\lambda(S) = 0 \) for all products \( i \) that are not in the assortment \( S \). The predicted expected revenue for a firm that offers assortment \( S \) under the ranking-based choice model with parameter \( \lambda \) is given by

\[
\mathcal{R}_i^\lambda(S) \triangleq \sum_{i \in \mathcal{N}_0} r_i \mathcal{D}_i^\lambda(S) = r^T \mathcal{D}^\lambda(S).
\]

We say that a ranking-based choice model is consistent with the historical sales data generated by the firm’s previously-offered assortments if the difference between the predicted purchase frequency \( \mathcal{D}_i^\lambda(S_m) \) and the historical sales data \( v_{m,i} \) is small for each of the products \( i \in S_m \) that were offered in each of the previously-offered assortments \( m \in \mathcal{M} \). We define the set of all ranking-based choice models that are consistent with the historical sales data as

\[
\mathcal{U} \triangleq \left\{ \lambda \in \Delta_\Sigma : \text{there exists a vector } \epsilon \text{ such that } \|\epsilon\| \leq \eta \text{ and } \mathcal{D}_i^\lambda(S_m) - v_{m,i} = \epsilon_{m,i} \text{ for all } i \in S_m \text{ and } m \in \mathcal{M} \right\},
\]

where the radius \( \eta \geq 0 \) of the set \( \mathcal{U} \) is a parameter that is selected by the firm.

To develop an understanding for the above set of ranking-based choice models \( \mathcal{U} \), let us consider the case in which the radius \( \eta \) of the above set is equal to zero. In that case, we observe that the above set contains exactly the probability distributions for which the corresponding ranking-based choice models have perfect accuracy on the historical sales data generated by the previously-offered assortments. In other words, if \( \eta = 0 \), then the set \( \mathcal{U} \) is comprised of all of the probability distributions \( \lambda \in \Delta_\Sigma \) that satisfy \( \mathcal{D}_i^\lambda(S_m) = v_{m,i} \) for each of the products \( i \in S_m \) that were offered in each of the previously-offered assortments \( m \in \mathcal{M} \). From a theoretical perspective, it is known that the set \( \mathcal{U} \) with \( \eta = 0 \) is guaranteed to be nonempty if the firm’s customers’ behavior is captured by a random utility maximization model and if the historical sales data has been observed without noise; see Block and Marschak (1959). From a practical perspective, it can be reasonable to expect that the set \( \mathcal{U} \) will be nonempty with \( \eta = 0 \) in problem instances in which the number of previously-offered assortments \( M \) is much smaller than the number of parameters in the ranking-based choice model, \( |\Sigma| = (n + 1)! \). Nonetheless, if there are no ranking-based choice models that have perfect accuracy on the historical sales data, then the radius \( \eta \) can always be made sufficiently positive to ensure that the set of ranking-based choice models \( \mathcal{U} \) is nonempty. For simplicity, we make
the standing assumption throughout our paper that the set of ranking-based choice models \( U \) is nonempty for the firm’s selection of the radius \( \eta \).

The goal of the present work is to make progress on answering the *identification question* for ranking-based choice models, which is stated formally as follows:

**The Identification Question**

Given a collection of previously-offered assortments \( \mathcal{M} \), historical sales data \( v_1, \ldots, v_M \), and prices \( r_1, \ldots, r_n \), is it possible to identify an assortment with an expected revenue that is strictly greater than the expected revenues of the firm’s previously-offered assortments under all of the ranking-based choice models that are consistent with the historical sales data generated by the firm’s previously-offered assortments? That is, does there exist an assortment \( S \in \mathcal{S} \) that satisfies

\[
\mathcal{R}^{\lambda}(S) > \max_{m \in \mathcal{M}} r^\top v_m \text{ for all } \lambda \in \mathcal{U}.
\]

In this work, we provide the first answers to the identification question by viewing the identification question through the lens of *robust optimization*. Specifically, we consider the following robust assortment optimization problem, which seeks an assortment that maximizes the predicted expected revenue under the worst-case ranking-based choice model that is consistent with the historical sales data generated by the firm’s previously-offered assortments:

\[
\max_{S \in \mathcal{S}} \min_{\lambda \in \mathcal{U}} \mathcal{R}^{\lambda}(S). \tag{RO}
\]

Here, our interest in the robust assortment optimization problem (RO) is motivated by its relevance to the identification question. Specifically, we readily observe that there exists an affirmative answer to the identification question if and only if the optimal objective value of the robust assortment optimization problem (RO) is strictly greater than the highest expected revenue from the firm’s previously-offered assortments, \( \max_{m \in \mathcal{M}} r^\top v_m \). In the following section, we will develop a powerful structural result for the robust assortment optimization problem (RO) which will enable us to develop closed-form solutions and general algorithms for answering the identification question.

### 3. Characterization of Optimal Assortments for (RO)

In this section, we establish the first characterization of the structure of optimal assortments for the robust assortment optimization problem (RO). In particular, we will show that there are optimal assortments for (RO) with a simple structure that is closely related to the structure of revenue-ordered assortments. Recall the following definition of the collection of revenue-ordered assortments:

\[
\hat{\mathcal{S}} \triangleq \{ S \in \mathcal{S} : \text{ if } i^* \in S \text{ and } r_{i^*} < r_i, \text{ then } i \in S \}.
\]
A fundamental result in the theory of assortment optimization is that revenue-ordered assortments are optimal under the multinomial logit choice model (Talluri and Van Ryzin 2004, Gallego et al. 2004, Rusmevichientong et al. 2014). Revenue-ordered assortments also have attractive approximation guarantees for assortment optimization problems under mixture-of-logits and ranking-based choice models with known parameters (Rusmevichientong et al. 2014, Aouad et al. 2018, Berbeglia and Joret 2020). Due to their simplicity and strong theoretical and empirical performance, revenue-ordered assortments are widely recommended in the revenue management literature and used in industry.

In view of the above background, we proceed to develop our main result regarding the structure of optimal assortments for the robust assortment optimization problem (RO). To this end, we first define the following set of previously-offered assortments for each product $i \in \mathcal{N}_0$:

$$M_i \triangleq \{ m \in \mathcal{M} : i \in S_m \}.$$ 

In words, the above set contains all of the previously-offered assortments in which the firm offered product $i$ to their customers. In particular, we observe from this definition that the statement $M_i \subseteq M_j$ holds if, for all of the previously-offered assortments in which the firm offered product $i$, the firm also offered product $j$. We now introduce the following new collection of assortments:

$$\hat{S} \triangleq \{ S \in \mathcal{S} : \text{if } i^* \in S, r_{i^*} < r_{i}, \text{ and } M_{i^*} \subseteq M_i, \text{ then } i \in S \}.$$ 

The above collection has a natural interpretation as the collection of all assortments which, speaking informally, can be viewed as revenue-ordered relative to the firm’s previously-offered assortments. Indeed, consider two products $i^*, i \in \mathcal{N}_0$ for which the revenue $r_{i^*}$ from the first product $i^*$ is strictly less than the revenue $r_i$ from the second product $i$. Then every assortment $S \in \hat{S}$ which offers the first product must also offer the second product unless there is historical sales data from a previously-offered assortment in which the first product $i^*$ was offered and the second product $i$ was not offered. Said another way, the collection $\hat{S}$ is comprised of all of the assortments which are revenue-ordered except on pairs of products in which the demand for the lower-revenue product has previously been observed independently of the demand for the higher-revenue product.

Our main result is the following:

**Theorem 1.** There exists an assortment $S \in \hat{S}$ that is optimal for (RO).

Our proof of Theorem 1 is contained in the remainder of the present section in §3.1-§3.3. As an immediate consequence of Theorem 1, we will be able to develop exact algorithms for solving the robust assortment optimization problem (RO) that consist of optimizing over only the assortments in the collection $\hat{S}$. In §4, we develop and analyze the tractability of such exact algorithms and use our algorithms to develop the first answers to the identification question.
3.1. Preliminary Steps

We begin our proof of Theorem 1 by discussing linear optimization-based techniques for computing the worst-case expected revenue $\min_{\lambda, \epsilon} R^\lambda (S)$ corresponding to any assortment $S \in \mathcal{S}$. Following the notation from §2, we readily observe that the worst-case expected revenue for the fixed assortment $S \in \mathcal{S}$ is equal to the optimal objective value of the following linear optimization problem:

$$
\min_{\lambda, \epsilon} \sum_{\sigma \in \Sigma} \sum_{i \in S} r_i \left\{ i = \arg \min_{j \in S} \sigma(j) \right\} \lambda_{\sigma}
$$

subject to

$$
\sum_{\sigma \in \Sigma} \left\{ i = \arg \min_{j \in S_m} \sigma(j) \right\} \lambda_{\sigma} - \epsilon_{m,i} = v_{m,i} \quad \forall m \in \mathcal{M} \text{ and } i \in S_m
$$

$$
\sum_{\sigma \in \Sigma} \lambda_{\sigma} = 1
$$

$$
\|\epsilon\| \leq \eta
$$

$$
\lambda_{\sigma} \geq 0 \quad \forall \sigma \in \Sigma.
$$

(WC-S)

As a preliminary step that will facilitate our developments in the subsequent subsections, the rest of §3.1 follows reformulation techniques from Jagabathula and Rusmevichientong (2019, §2) to reduce (WC-S) to a linear optimization problem with a smaller number of decision variables.

To construct our compact reformulation of the linear optimization problem (WC-S), we introduce the following additional notation. For each assortment $S \in \mathcal{S}$ and each product in the assortment $i \in S$, let the set of rankings that prefer product $i$ to all other products in the assortment $S$ be defined as follows.

**Definition 1.** $D_i(S) \triangleq \{ \sigma \in \Sigma : i = \arg \min_{j \in S} \sigma(j) \}$.

Given the previously-offered assortments $S_1, \ldots, S_M$, we also define the following set of tuples of products.

**Definition 2.** $\mathcal{L} \triangleq \{ (i_1, \ldots, i_M) \in S_1 \times \cdots \times S_M : \cap_{m \in \mathcal{M}} D_{i_m}(S_m) \neq \emptyset \}$.

To develop intuition for Definition 2, let us reflect on the relationship between the set of tuples of products $\mathcal{L}$ and the set of all distinct rankings $\Sigma$. Firstly, it follows immediately from Definition 2 that each tuple of products $(i_1, \ldots, i_M) \in \mathcal{L}$ has at least one ranking $\sigma \in \Sigma$ that satisfies $\sigma \in \cap_{m \in \mathcal{M}} D_{i_m}(S_m)$. Secondly, we show in the following lemma that each ranking $\sigma \in \Sigma$ satisfies $\sigma \in \cap_{m \in \mathcal{M}} D_{i_m}(S_m)$ for exactly one tuple of products $(i_1, \ldots, i_M) \in \mathcal{L}$.

**Lemma 1.** For each $\sigma \in \Sigma$, there exists a unique $(i_1, \ldots, i_M) \in \mathcal{L}$ such that $\sigma \in \cap_{m \in \mathcal{M}} D_{i_m}(S_m)$.

The third and final definition of §3.1, which is presented below as Definition 3, will play a significant role in our developments in the rest of §3. Specifically, the following definition introduces a
quantity \( \rho_{i_1 \ldots i_M}(S) \) for each assortment \( S \in \mathcal{S} \) and each tuple of products \( (i_1, \ldots, i_M) \in \mathcal{L} \). The quantity \( \rho_{i_1 \ldots i_M}(S) \) can be understood as the minimum revenue among the products in the assortment \( S \in \mathcal{S} \) that can be the most preferred product in \( S \) under a ranking that corresponds to the tuple of products \( (i_1, \ldots, i_M) \in \mathcal{L} \). It will become clear momentarily that the quantity \( \rho_{i_1 \ldots i_M}(S) \) arises naturally when constructing our compact reformulation of the linear optimization problem (WC-S).

**Definition 3.** \( \rho_{i_1 \ldots i_M}(S) \triangleq \min_{i \in S : \cap_{m \in M} D_{im}(S_m) \cap D_1(S) \neq \emptyset} r_i \).

Equipped with the above definitions, we are now ready to show that the linear optimization problem (WC-S) for computing the worst-case expected revenue of any fixed assortment \( S \in \mathcal{S} \) can be reformulated as a linear optimization problem with a reduced number of decision variables. This compact reformulation of (WC-S) is presented as (WC'-S) in the following Proposition 1.

**Proposition 1.** (WC-S) is equivalent to the following linear optimization problem:

\[
\begin{align*}
\text{minimize} & \quad \sum_{(i_1, \ldots, i_M) \in \mathcal{L}} \rho_{i_1 \ldots i_M}(S) \lambda_{i_1 \ldots i_M} \\
\text{subject to} & \quad \sum_{(i_1, \ldots, i_M) \in \mathcal{L} : \cap_{m \in M} D_{im}(S_m) \cap D_1(S) \neq \emptyset} \lambda_{i_1 \ldots i_M} - \epsilon_{m,i} = v_{m,i} \quad \forall m \in \mathcal{M}, i \in S_m \\
& \quad \sum_{(i_1, \ldots, i_M) \in \mathcal{L}} \lambda_{i_1 \ldots i_M} = 1 \\
& \quad \|\epsilon\| \leq \eta \\
& \quad \lambda_{i_1 \ldots i_M} \geq 0 \quad \forall (i_1, \ldots, i_M) \in \mathcal{L}.
\end{align*}
\]

(WC'-S)

Let us offer two remarks about the linear optimization problem (WC'-S). In particular, the following two remarks articulate the key properties of (WC'-S) that will be important to our developments in the subsequent subsections.

**Remark 1.** The assortment \( S \) does not appear in any of the constraints of (WC'-S).

**Remark 2.** (WC'-S) has \( |\mathcal{L}| = \mathcal{O}(n^M) \) decision variables and \( \mathcal{O}(nM) \) constraints.

We note that the asymptotic upper bound in Remark 2 on the number of decision variables in the linear optimization problem (WC'-S) follows immediately from the fact that the set of tuples of products \( \mathcal{L} \) is a subset of \( S_1 \times \cdots \times S_M \); see Definition 2.

### 3.2. A Graphical Interpretation of Definition 3

Equipped with Proposition 1, we now describe our overarching strategy for our proof of Theorem 1. In a nutshell, our proof of Theorem 1 will follow an exchange argument. For every arbitrary assortment \( S \in \mathcal{S} \), we will show that we can construct an assortment \( S' \in \hat{\mathcal{S}} \) that satisfies...
\( \rho_{i_1\ldots i_M}(S) \leq \rho_{i_1\ldots i_M}(S') \) for all tuples of products \((i_1, \ldots, i_M) \in \mathcal{L}\). By showing this, it will follow readily from Proposition 1 and Remark 1 that the worst-case expected revenue for the new assortment \(\min_{\lambda \in \mathcal{U}} R^\lambda(S')\) will never be less than the worst-case expected revenue for the original assortment \(\min_{\lambda \in \mathcal{U}} R^\lambda(S)\). This will prove that there always exists an assortment \(S' \in \hat{S}\) that is optimal for the robust assortment optimization problem (RO).

In view of our overarching strategy for the proof of Theorem 1, we now proceed to analyze the behavior of the functions \(S \mapsto \rho_{i_1\ldots i_M}(S)\). In particular, we will show in the rest of §3.2 that \(\rho_{i_1\ldots i_M}(S)\) can be computed by analyzing the reachability of vertices in a directed acyclic graph. In the subsequent §3.3, we will use this graphical interpretation of Definition 3 to show for every arbitrary assortment \(S \in \mathcal{S}\) that we can construct an assortment \(S' \in \hat{S}\) that satisfies \(\rho_{i_1\ldots i_M}(S) \leq \rho_{i_1\ldots i_M}(S')\) for all tuples of products \((i_1, \ldots, i_M) \in \mathcal{L}\), thereby completing the proof of Theorem 1.

To develop our alternative representation of \(\rho_{i_1\ldots i_M}(S)\), we begin by showing that the set of tuples of products \(\mathcal{L}\) can be interpreted as a set of directed acyclic graphs. Indeed, consider any selection of products from each of the previously-offered assortments, \((i_1, \ldots, i_M) \in S_1 \times \cdots \times S_M\).

From this tuple of products, we will construct a directed graph, denoted by \(G_{i_1\ldots i_M}\), in which the set of vertices in the graph is equal to \(\mathcal{N}_0\), and the graph has a directed edge \((i, i_m)\) from vertex \(i\) to vertex \(i_m\) for each previously-offered assortment \(m \in \mathcal{M}\) and each product \(i \in S_m \setminus \{i_m\}\). In Figure 1, we present visualizations of the directed graphs generated by this construction procedure. In the first intermediary result of this subsection, presented below as Lemma 2, we show that the tuple of products \((i_1, \ldots, i_M)\) is an element of \(\mathcal{L}\) if and only if the directed graph \(G_{i_1\ldots i_M}\) is acyclic.

**Lemma 2.** \((i_1, \ldots, i_M) \in \mathcal{L}\) if and only if \(G_{i_1\ldots i_M}\) is acyclic.

We next introduce the definition of reachability for vertices in the directed graph \(G_{i_1\ldots i_M}\). In particular, the following definition is standard in the study of directed graphs, and we will make extensive use of Definition 4 throughout the remainder of §3.

**Definition 4.** Let \((i_1, \ldots, i_M) \in S_1 \times \cdots \times S_M\) and \(i, j \in \mathcal{N}_0\). We say that vertex \(j\) is reachable from vertex \(i\), denoted by \(j \prec_{i_1\ldots i_M} i\), if there exists a directed path in \(G_{i_1\ldots i_M}\) from \(i\) to \(j\).

We adopt the convention throughout this paper that a directed path must contain at least one directed edge. Hence, we observe that if the directed graph \(G_{i_1\ldots i_M}\) is acyclic, then it must be the case that a vertex may never be reachable from itself, that is, \(i \not\prec_{i_1\ldots i_M} i\) for all \(i \in \mathcal{N}_0\).

We now use Definition 4 to develop several intermediary results regarding the structure of the directed acyclic graph \(G_{i_1\ldots i_M}\) for each tuple of products \((i_1, \ldots, i_M) \in \mathcal{L}\). We begin with a simple intermediary result, denoted below by Lemma 3, in which we characterize the vertices in the directed acyclic graph \(G_{i_1\ldots i_M}\) that can be reachable from other vertices.
Figure 1 Visualizations of directed graphs $G_{i_1 \cdots i_M}$ corresponding to tuples of products $(i_1, \ldots, i_M) \in L$.

(a) $(i_1, i_2, i_3) = (0, 0, 0)$  
(b) $(i_1, i_2, i_3) = (1, 1, 0)$  
(c) $(i_1, i_2, i_3) = (1, 1, 2)$  
(d) $(i_1, i_2, i_3) = (2, 0, 2)$  
(e) $(i_1, i_2, i_3) = (2, 1, 2)$

Note. Each of the five figures presents a visualization of the directed graph $G_{i_1 \cdots i_M}$ corresponding to a tuple of products $(i_1, i_2, i_3) \in L$ in the case where there are $M = 3$ previously-offered assortments of the form $S_1 = \{0, 1, 2\}$, $S_2 = \{0, 1\}$, and $S_3 = \{0, 2\}$. We observe that there exists an incoming edge to vertex $i$ if and only if there exists a previously-offered assortment $m \in \{1, 2, 3\}$ that satisfies $i = i_m$.

**Lemma 3.** Let $(i_1, \ldots, i_M) \in L$ and $i, j \in N_0$. If $j \prec_{i_1 \cdots i_M} i$, then there exists a previously-offered assortment $m \in M$ that satisfies $j = i_m$.

**Proof.** Consider any tuple of products $(i_1, \ldots, i_M) \in L$. We recall from our construction of $G_{i_1 \cdots i_M}$ that $i_1, \ldots, i_M$ are the only vertices in $G_{i_1 \cdots i_M}$ that have incoming edges. Since Lemma 2 implies that $G_{i_1 \cdots i_M}$ is acyclic, we conclude that $i_1, \ldots, i_M$ are the only vertices in $G_{i_1 \cdots i_M}$ that can be reachable from other vertices. □

In our next intermediary result, denoted by Lemma 4, we relate the reachability of vertices in a directed acyclic graph $G_{i_1 \cdots i_M}$ to the set of rankings that correspond to $(i_1, \ldots, i_M) \in L$.

**Lemma 4.** Let $(i_1, \ldots, i_M) \in L$ and $i, j \in N_0$. Then, there exists a ranking $\sigma \in \cap_{m \in M} D_{i_m}(S_m)$ that satisfies $\sigma(i) < \sigma(j)$ if and only if $j \not\prec_{i_1 \cdots i_M} i$.

Intuitively, Lemma 4 shows that the reachability of vertices in a directed acyclic graph $G_{i_1 \cdots i_M}$ provides an encoding of the rankings that correspond to a tuple of products $(i_1, \ldots, i_M) \in L$. Said another way, Lemma 4 implies that if vertex $i$ has a directed path to vertex $j$ in a directed acyclic graph $G_{i_1 \cdots i_M}$, then product $j$ is always preferred to product $i$ under all rankings that correspond to the tuple of products $(i_1, \ldots, i_M) \in L$. In our final intermediary result in §3.2, denoted below
by Lemma 5, we develop a generalization of Lemma 4 that relates the reachability of vertices in a directed acyclic graph $G_{i_1 \cdots i_M}$ to the most preferred products in an assortment $S \in \mathcal{S}$.

**Lemma 5.** Let $(i_1, \ldots, i_M) \in \mathcal{L}$, $S \in \mathcal{S}$, and $i \in S$. Then, there exists a ranking $\sigma \in \cap_{m \in \mathcal{M}} D_{i_m}(S_m)$ that satisfies $i = \arg \min_{j \in S} \sigma(j)$ if and only if $j \nprec_{i_1 \cdots i_M} i$ for all $j \in S$.

Lemma 5 establishes that a product $i \in S$ is the most preferred product from assortment $S \in \mathcal{S}$ under a ranking that corresponds to the tuple of products $(i_1, \ldots, i_M) \in \mathcal{L}$ if and only if there does not exist a directed path in the directed acyclic graph $G_{i_1 \cdots i_M}$ from vertex $i$ to any vertex $j$ that satisfies $j \in S$. In other words, Lemma 5 implies that the set $\cap_{m \in \mathcal{M}} D_{i_m}(S_m) \cap D_i(S)$ is nonempty if and only if there is no vertex $j \in S$ which is reachable from vertex $i$.

In view of the above, we are now ready to develop our graphical interpretation of $\rho_{i_1 \cdots i_M}(S)$. This interpretation of $\rho_{i_1 \cdots i_M}(S)$, which is presented below in Proposition 2, will be instrumental to our analysis in the subsequent §3.3, where we will use this interpretation to analyze the behavior of the functions $S \mapsto \rho_{i_1 \cdots i_M}(S)$ for each tuple of products $(i_1, \ldots, i_M) \in \mathcal{L}$. Our graphical representation of $\rho_{i_1 \cdots i_M}(S)$ requires the following definition of the set $\mathcal{I}_{i_1 \cdots i_M}(S)$, which can be interpreted as the set of all vertices $i \in N_0$ in the directed graph $G_{i_1 \cdots i_M}$ that do not have a directed path to any of the vertices $i_1, \ldots, i_M$ that are elements of the assortment $S$.

**Definition 5.** $\mathcal{I}_{i_1 \cdots i_M}(S) \triangleq \{i \in N_0 : \text{ for all } m \in \mathcal{M}, \text{ if } i_m \in S, \text{ then } i_m \nprec_{i_1 \cdots i_M} i\}$.

To make sense of Definition 5, we recall from Lemma 3 that a vertex $j$ in a directed acyclic graph $G_{i_1 \cdots i_M}$ can be reachable from another vertex only if $j = i_m$ for some previously-offered assortment $m \in \mathcal{M}$. Therefore, it follows immediately from Lemma 5 that $S \cap \mathcal{I}_{i_1 \cdots i_M}(S)$ is the set of products $i$ for which the set of rankings $\cap_{m \in \mathcal{M}} D_{i_m}(S_m) \cap D_i(S)$ is nonempty. Combining this with Definition 3, we have concluded the proof of the following Proposition 2, which establishes our graphical interpretation of $\rho_{i_1 \cdots i_M}(S)$.

**Proposition 2.** For all $S \in \mathcal{S}$ and $(i_1, \ldots, i_M) \in \mathcal{L}$, $\rho_{i_1 \cdots i_M}(S) = \min_{i \in S \cap \mathcal{I}_{i_1 \cdots i_M}(S)} \tau_i$.

### 3.3. Proof of Theorem 1

Equipped with Propositions 1 and 2, we are now ready to present our proof of Theorem 1. In view of our overarching strategy outlined in the beginning of §3.2, the main remaining step in the proof of Theorem 1 is showing for every assortment $S \in \mathcal{S}$ that we can construct an assortment $S' \in \hat{\mathcal{S}}$ that satisfies $\rho_{i_1 \cdots i_M}(S) \leq \rho_{i_1 \cdots i_M}(S')$ for each $(i_1, \ldots, i_M) \in \mathcal{L}$. To show this, we begin by developing an intermediary result, denoted below by Lemma 6, that will allow us to compare the values of $\rho_{i_1 \cdots i_M}(S)$ and $\rho_{i_1 \cdots i_M}(S \cup \{i\})$ for every assortment $S \in \mathcal{S}$ and every product $i$ which is not in the assortment.
Lemma 6. For all $S \in \mathcal{S}$, $(i_1, \ldots, i_M) \in \mathcal{L}$, and $i \notin S$,

$$
\rho_{i_1 \ldots i_M}(S \cup \{i\}) = \begin{cases} 
\rho_{i_1 \ldots i_M}(S), & \text{if } i \notin \mathcal{I}_{i_1 \ldots i_M}(S), \\
\min \left\{ \min_{j \in S \cap \mathcal{I}_{i_1 \ldots i_M}(S)} \{j' \in \mathcal{N}_i : i \preceq j' \} r_j, r_i \right\}, & \text{if } i \in \mathcal{I}_{i_1 \ldots i_M}(S).
\end{cases}
$$

Using the above intermediary result, we show in the following Lemma 7 and Proposition 3 that for each assortment $S \in \mathcal{S}$, we can construct an assortment $S' \in \hat{\mathcal{S}}$ that satisfies $ho_{i_1 \ldots i_M}(S) \leq \rho_{i_1 \ldots i_M}(S')$ for all tuples of products $(i_1, \ldots, i_M) \in \mathcal{L}$.

Lemma 7. Let $S \in \mathcal{S}$ and $i \notin S$. If there exists $i^* \in S$ which satisfies $r_{i^*} < r_i$ and $\mathcal{M}_{i^*} \subseteq \mathcal{M}_i$, then $\rho_{i_1 \ldots i_M}(S) \leq \rho_{i_1 \ldots i_M}(S \cup \{i\})$ for each $(i_1, \ldots, i_M) \in \mathcal{L}$.

Proposition 3. For each assortment $S \in \mathcal{S}$, there exists an assortment $S' \in \hat{\mathcal{S}}$ which satisfies the inequality $\rho_{i_1 \ldots i_M}(S) \leq \rho_{i_1 \ldots i_M}(S')$ for each $(i_1, \ldots, i_M) \in \mathcal{L}$.

We now complete our proof of Theorem 1 by combining Proposition 1 with Proposition 3.

Proof of Theorem 1. It follows immediately from Proposition 1 that

$$
(RO) = \max_{S \in \mathcal{S}} \left\{ \min_{\lambda, \epsilon} \sum_{(i_1, \ldots, i_M) \in \mathcal{L}} \rho_{i_1 \ldots i_M}(S) \lambda_{i_1 \ldots i_M} \right\}
$$

subject to

$$
\sum_{(i_1, \ldots, i_M) \in \mathcal{L} : i_m = i} \lambda_{i_1 \ldots i_M} - \epsilon_{m,i} = v_{m,i} \quad \forall m \in \mathcal{M}, i \in \mathcal{S}_m
$$

$$
\sum_{(i_1, \ldots, i_M) \in \mathcal{L}} \lambda_{i_1 \ldots i_M} = 1
$$

$$
\|\epsilon\| \leq \eta
$$

$$
\lambda_{i_1 \ldots i_M} \geq 0 \quad \forall (i_1, \ldots, i_M) \in \mathcal{L}
$$

where the assortment $S \in \mathcal{S}$ in the maximization problem appears only in the objective of the inner minimization problem. Since any feasible solution $\lambda$ to the inner minimization problem is nonnegative, Theorem 1 follows immediately from Proposition 3. \qed

4. Answers to the Identification Question

In this section, we use Theorem 1 to develop the first answers to the identification question, both for specific collections of previously-offered assortments (§4.1 and §4.2) as well as algorithms for answering the identification question in general classes of problems (§4.2 and §4.3). We use these findings, together with concise numerical experiments, to argue that considering the identification question can be essential for finding high-quality assortments from ranking-based choice models in high-stakes assortment planning problems.
4.1. Impossibility Result for Revenue-Ordered Assortments

For our first answer to the identification question, we return to the widely-studied class of revenue-ordered assortments that was discussed in the beginning of §3. Due to their simplicity and desirable theoretical guarantees, a large body of literature has advocated to firms for offering revenue-ordered assortments across numerous application domains. Equipped with Theorem 1 we now show, surprisingly, that a firm which has offered the revenue-ordered assortments has, in a rigorous sense, operated under a worst-possible behavior from the perspective of obtaining affirmative answers to the identification question. For simplicity, we assume in the following Theorem 2 and Corollary 1 that the revenues of the products are distinct and satisfy \( r_1 < \cdots < r_n \).

**Theorem 2.** If \( \mathcal{M} = \mathcal{S} \) and \( \eta = 0 \), then \( \max_{S \in \mathcal{S}} \min_{\lambda \in \mathcal{U}} R(S) = \max_{m \in \mathcal{M}} r^\top v_m \).

*Proof.* Let \( \mathcal{M} = \mathcal{S} \), and let the previously-offered assortments be indexed by \( \mathcal{M} = \{ \bar{S}_1, \ldots, \bar{S}_n \} \), whereby the \( i \)-th previously-offered assortment is \( \bar{S}_i = \{ 0, i, i+1, \ldots, n-1, n \} \).

Equipped with the above notation, we first prove that the equality \( \bar{S} = \bar{\mathcal{S}} \) holds. Indeed, choose any arbitrary assortment \( S \in \mathcal{S} \), and let \( i^* = \arg\min_{j \in \mathcal{S}} r_j \) denote the product in the chosen assortment that has the smallest nonzero revenue. It readily follows from the facts that \( \mathcal{M} = \mathcal{S} \) and \( r_1 < \cdots < r_n \) that the equalities \( M_{i^*} = \{ m \in \mathcal{M} : m \leq i^* \} \) and \( M_i = \{ m \in \mathcal{M} : m \leq i \} \) hold for each \( i \in \{ i^* + 1, \ldots, n \} \). Therefore, for each \( i \in \{ i^* + 1, \ldots, n \} \), it follows from the definition of the collection \( \mathcal{S}_i \), from the fact that \( r_{i^*} < r_i \), and from the fact that \( M_{i^*} \subseteq M_i \) that \( i \in S \). We have thus shown that \( S = \{ 0, i^*, i^* + 1, \ldots, n-1, n \} = \bar{S}_{i^*} \), which implies that \( S \in \mathcal{S} \). Since the assortment \( S \in \bar{S} \) was chosen arbitrarily, we have shown that \( \bar{S} \subseteq \mathcal{S} \). The other direction of the proof that \( \mathcal{S} = \bar{S} \) follows from the fact that the inclusion \( \mathcal{M} \subseteq \mathcal{S} \) always holds\(^1\) and from the fact that \( \mathcal{M} = \mathcal{S} \). Our proof that \( \bar{S} = \bar{\mathcal{S}} \) is thus complete.

Using the above result, we have

\[
\max_{S \in \mathcal{S}} \min_{\lambda \in \mathcal{U}} R(S) = \max_{S \in \mathcal{S}} \min_{\lambda \in \mathcal{U}} R(S) = \max_{S \in \mathcal{S}} \min_{\lambda \in \mathcal{U}} R(S) = \max_{S \in \mathcal{S}} \min_{\lambda \in \mathcal{U}} R(S) = \max_{m \in \mathcal{M}} r^\top v_m,
\]

where the first equality follows from Theorem 1, the second equality holds because \( \bar{S} = \mathcal{S} \), the third equality holds because \( \mathcal{M} = \mathcal{S} \), and the final equality follows from the construction of the set of ranking-based choice models \( \mathcal{U} \) (see §2) and from the fact that \( \eta = 0 \). \( \square \)

**Corollary 1.** If \( \mathcal{M} = \mathcal{S} \) and \( \eta = 0 \), then for each assortment \( S \in \mathcal{S} \), there exists a ranking-based choice model consistent with the historical sales data \( \lambda \in \mathcal{U} \) that satisfies \( R(S) \leq \max_{m \in \mathcal{M}} r^\top v_m \).

\(^1\)To see why the inclusion \( \mathcal{M} \subseteq \mathcal{S} \) always holds, consider any previously-offered assortment \( S \in \mathcal{M} \). For each product \( i^* \in S \), suppose that there exists another product \( i \) which satisfies \( r_{i^*} < r_i \) and \( M_{i^*} \subseteq M_i \). Since \( S \in M_{i^*} \subseteq M_i \), we conclude that \( i \in S \) must hold, which proves that \( S \in \mathcal{S} \).
Proof. The proof of Corollary 1 follows immediately from Theorem 2. □

Stated in words, the above Theorem 2 and Corollary 1 establish that we can never have affirmative answers to the identification question when the firm has historical sales data that is generated by the revenue-ordered assortments. That is, it is impossible using the given historical sales data to identify a new assortment with a predicted expected revenue that strictly outperforms the expected revenues of the firm’s past assortments under all of the ranking-based choice models that are consistent with the firm’s historical sales data.

To understand the practical importance of a negative answer to the identification question, let us consider the estimate-then-optimize technique discussed in §1. In other words, suppose that one estimates a ranking-based choice model from the historical sales data generated by the revenue-ordered assortments and then recommends that the firm implement a new assortment which maximizes the predicted expected revenue under the estimated ranking-based choice model. For this setting, Theorem 2 and Corollary 1 guarantee that this estimate-then-optimize technique will never offer fidelity to the firm, in the sense that there will always exist a ranking-based choice model which is consistent with the historical sales data for which the expected revenue for the new assortment will be less than or equal to the expected revenue of the best previously-offered assortment. Moreover, as we will further see through the following concise yet insightful numerical experiment, the expected revenue from the assortment recommended by the estimate-then-optimize technique can be strictly less than the expected revenue of the best previously-offered assortment.

To perform our numerical experiment, we begin by constructing randomly-generated problem instances. In each problem instance, the revenues for the products are drawn from the distribution \( r_1, \ldots, r_n \sim \text{Uniform}[0, 1] \), and a base choice for the parameters \( \lambda^* \) of a ranking-based choice model is drawn uniformly over the \((n+1)\)!-dimensional probability simplex.\(^2\) Using this base choice for the parameters, we generate historical sales data of the form \( v_1, \ldots, v_n \), where each \( v_m \) is the historical sales data generated by the revenue-ordered assortment \( \{0, m, m+1, \ldots, n-1, n\} \) under the ranking-based choice model with the base parameter \( \lambda^* \).\(^3\) After we compute the historical sales data, we forget the base parameters \( \lambda^* \) of the ranking-based choice model and apply the estimate-then-optimize technique to obtain a new assortment. Specifically, we first estimate the parameters \( \hat{\lambda} \) of a ranking-based choice model using the historical sales data; since many selections of the parameters may be consistent with the historical sales data, we choose our estimate \( \hat{\lambda} \) as the optimal

\(^2\) If the rankings in \( \Sigma \) are indexed by \( \{\sigma_1, \ldots, \sigma_{(n+1)!}\} \), then uniform sampling over the probability simplex is obtained by drawing \( u_1, \ldots, u_{(n+1)!} \sim \text{Uniform}[0, 1] \) and setting \( \lambda_{\sigma_k} \leftarrow \log(u_k) / \sum_{k'=1}^{(n+1)!} \log(u_{k'}) \) for each \( k = 1, \ldots, (n+1)! \).

\(^3\) We sort the products in ascending order by revenue before performing our analysis, which ensures that \( \{0, m, m+1, \ldots, n\} \) for each \( m \in \{1, \ldots, n\} \) is a revenue-ordered assortment.
solution to the linear optimization problem \( \min_{\lambda \in \mathcal{U}} c^T \lambda \), where the cost vector \( c \) is drawn uniformly over \([0, 1]^{(n+1)!}\).\(^4\) We then obtain a new assortment \( S' \) as any optimal solution to the combinatorial optimization problem \( \max_{S \in S} \mathcal{R}^\lambda(S) \) which maximizes the predicted expected revenue under the estimated ranking-based choice model.\(^6\) Finally, we evaluate the new assortment obtained using estimate-then-optimize by computing the worst-case expected revenue of the new assortment under all ranking-based choice models that are consistent with the historical sales data, \( \min_{\lambda \in \mathcal{U}} \mathcal{R}^\lambda(S') \), the best-case expected revenue of the new assortment under all ranking-based choice models that are consistent with the historical sales data, \( \max_{\lambda \in \mathcal{U}} \mathcal{R}^\lambda(S') \), and the expected revenue of the best previously-offered assortment, \( \max_{m \in \mathcal{M}} r^T v_m \).

In Figures 2 and 3, we present the results of these numerical experiments for the case of \( n = 4 \) products over 1000 randomly-generated problem instances. In Figure 2, we compare the expected revenue under the best previously-offered assortment, \( \max_{m \in \mathcal{M}} r^T v_m \), to the expected revenue of the new assortments obtained using estimate-then-optimize under the worst-case ranking-based choice model that is consistent with the historical sales data, \( \min_{\lambda \in \mathcal{U}} \mathcal{R}^\lambda(S') \). We observe that the results in Figure 2 are consistent with the impossibility result from Theorem 2; indeed, the worst-case expected revenues of the new assortments obtained using estimate-then-optimize never exceed the expected revenues of the best previously-offered assortments. Furthermore, we observe for many of the problem instances that there are ranking-based choice models \( \lambda \in \mathcal{U} \) that are consistent with the historical sales data for which the resulting expected revenue of the new assortment \( \mathcal{R}^\lambda(S') \) is strictly less than the expected revenue under the best previously-offered assortment, \( \max_{m \in \mathcal{M}} r^T v_m \).

To assess whether the above findings are overly conservative from a practical standpoint, we turn to a detailed analysis of the 137 problem instances for which the expected revenue of the best previously-offered assortment, \( \max_{m \in \mathcal{M}} r^T v_m \), is strictly greater than the worst-case expected revenue \( \min_{\lambda \in \mathcal{U}} \mathcal{R}^\lambda(S') \) for the new assortment \( S' \) obtained using estimate-then-optimize. In Figure 3 we present, for each of these 137 problem instances, a visualization of the range of relative

\(^4\) Alternative approaches for estimating the parameters of a ranking-based choice model from historical sales data are provided in Mišić (2016), van Ryzin and Vulcano (2015, 2017), Désir et al. (2021a). Our approach of estimating the parameters as \( \hat{\lambda} \in \arg \min_{\lambda \in \mathcal{U}} c^T \lambda \) for a randomly-chosen cost vector \( c \) is motivated by our desire to decouple any potential biases associated with any particular estimation procedure from an empirical assessment of the estimate-then-optimize technique. In particular, our approach is viewed as a simple way of randomly sampling the parameters from the set of all parameters of ranking-based choice models which are consistent with the historical sales data.

\(^5\) An obvious downside of our simple estimation procedure for the parameters of the ranking-based choice model is that it requires solving a linear optimization problem with \( O(n!) \) decision variables, and, thus, our simple estimation procedure does not scale efficiently to problem instances with many products. Nonetheless, this estimation procedure is sufficiently fast for the purposes of this numerical study, where the aim is simply to assess the performance of assortments obtained by the estimate-then-optimize technique over revenue-ordered assortments. In particular, we believe it is a reasonable assumption that similar findings from our numerical experiment with \( n = 4 \) (see Figures 2 and 3) would be found in experiments with larger values of \( n \).

\(^6\) We solve this optimization problem using the mixed-integer linear optimization formulation given by Bertsimas and Mišić (2019, §3.2), which is implemented using the Julia programming language with JuMP and solved using Gurobi.
Figure 2  Performance of new assortments obtained using the estimate-then-optimize technique when \( \mathcal{M} = \tilde{\mathcal{S}} \).

Note. Each point corresponds to a randomly-generated problem instance. The \( x \)-axis shows the expected revenue of the best previously-offered assortment, \( \max_{m \in \mathcal{M}} r^tv_m \). The \( y \)-axis shows the worst-case expected revenue \( \min_{\lambda \in \mathcal{U}} R^\lambda(S') \) for the new assortment \( S' \) obtained using the estimate-then-optimize technique.

The results in Figure 3 reveal a striking asymmetry between the downside and upside of implementing a new assortment found by estimate-then-optimize. In all but two of the 137 instances, the percentage improvements of the predicted expected revenue of the new assortment, \( R^\lambda(S') \), over the expected revenue of the firm’s best previously-offered assortment, \( \max_{m \in \mathcal{M}} r^tv_m \), which are possible to obtain under a ranking-based choice model which is consistent with the historical sales data, \( \lambda \in \mathcal{U} \). Stated more precisely, each of the \( x \) values in Figure 3 corresponds to one of these 137 problem instances, and the corresponding interval of \( y \)-values formed by the red and blue bars is the interval \( [\Pi_\lambda : \lambda \in \mathcal{U}] \), where \( \Pi_\lambda \triangleq 100\% \times (R^\lambda(S') - \max_{m \in \mathcal{M}} r^tv_m) / (\max_{m \in \mathcal{M}} r^tv_m) \) is the relative percentage improvement of the predicted expected revenue of the new assortment \( S' \) over the expected revenue of the firm’s best previously-offered assortment for a given ranking-based choice model \( \lambda \in \mathcal{U} \). Hence, the red bars are the negative values of \( \Pi_\lambda \) that can be attained under the ranking-based choice models \( \lambda \in \mathcal{U} \), and the dotted line shows the reflection of the endpoints of red bars over the horizontal line at zero. We do not assign any likelihood to the values in each interval \( [\Pi_\lambda : \lambda \in \mathcal{U}] \), as there is no information available for inferring which of the ranking-based choice models are more likely to be the ‘truth’ among the ranking-based choice models \( \lambda \in \mathcal{U} \) that are consistent with the historical sales data.
Figure 3  Relative improvement in expected revenue of new assortments obtained using the estimate-then-optimize technique when $\mathcal{M} = \mathcal{S}$.

Note. Each of the $x$-values corresponds to one of the 137 randomly-generated problem instances from Figure 2 for which the expected revenue of the best previously-offered assortment, $\max_{m \in \mathcal{M}} r^\top v_m$, was strictly greater than the worst-case expected revenue $\min_{\lambda \in \mathcal{U}} \mathcal{R}^\lambda(S')$ for the new assortment $S'$ obtained using the estimate-then-optimize technique. The corresponding interval of $y$-values formed by the red and blue bars is the interval $[\Pi_{\lambda} : \lambda \in \mathcal{U}]$, where $\Pi_{\lambda} \triangleq 100\% \times (\mathcal{R}^\lambda(S') - \max_{m \in \mathcal{M}} r^\top v_m) / (\max_{m \in \mathcal{M}} r^\top v_m)$ is the relative percentage improvement of the expected revenue of the new assortment obtained using estimate-then-optimize over the expected revenue of the firm’s best previously-offered assortment for a given ranking-based choice model $\lambda \in \mathcal{U}$. For clarity, the problem instances are sorted along the $x$-axis by the endpoints of the red bars, and the dotted line shows the reflection of the endpoints of red bars over the horizontal line at zero.

worst-case decline in expected revenue from implementing the new assortment exceeded in magnitude the best-case increase in expected revenue. The difference in magnitude between the downside and upside is also found to be significant: the average best-case improvement of the new assortment over the best previously-offered assortment (i.e., the average of the blue endpoints) is 6.56%, while the average worst-case improvement of the new assortment over the best previously-offered assortment (i.e., the average of the red endpoints) is -21.71%. These numerical findings demonstrate that the downside risks to a firm from implementing a new assortment found by estimate-then-optimize can significantly exceed the potential upsides.

In conclusion, our theoretical and numerical analysis in this subsection lead to three main takeaways. The first takeaway is that there exist collections of previously-offered assortments in which the identification question can never have an affirmative answer. This is practically useful because
if such a negative answer can be established for a problem instance (either through impossibility results like Theorem 2 or via general algorithms like those presented in §4.2 and §4.3), then a firm can be encouraged to pursue additional small-scale experiments before committing to a large-scale implementation of a new assortment. The second takeaway is that commonly-used techniques like estimate-then-optimize can lead to a strictly worse expected revenue than those of the previously-offered assortments. In fact, the numerical results in Figures 2 and 3 show that this decline in expected revenue can be significant and outweigh the potential upside for implementing the new assortment. The third takeaway is that all of the aforementioned issues arise when the previously-offered assortments are comprised of one of the most celebrated and widely-used classes of assortments from the literature, namely, the revenue-ordered assortments. All together, these takeaways raise concerns about whether the estimate-then-optimize technique with ranking-based choice models should be trusted in high-stakes assortment planning problems.

4.2. Affirmative Answers for Two Assortments

In view of the impossibility result from the previous subsection, we next turn to using Theorem 1 to assess whether the identification question can ever be answered affirmatively. In the subsection, we establish that the answer to this question is yes, and in particular, we show that such an affirmative answer can be obtained even when the firm has only offered two past assortments.

In order to establish these results, we begin by leveraging Theorem 1 to develop the first algorithm for solving the robust assortment optimization problem (RO) with running time that is polynomial in the number of products $n$. The development of such an algorithm is important because it will allow us to establish the existence of affirmative answers to the identification question through numerical experiments. Our development in this subsection of an algorithm for the case of $M = 2$ is also important because it marks the first step towards the development of efficient general algorithms for answering the identification question in settings with large numbers of products and relatively few past assortments. Problem instances in which there are small numbers of previously-offered assortments and large numbers of products can arise in high-stakes applications in which changing to low-quality assortments can have significant negative consequences, and so the firm has made relatively few changes to their assortments thus far. We will show that the algorithms from this subsection extend to general settings with several past assortments in the following §4.3.

Our polynomial-time algorithm for answering the identification question when $M = 2$ is presented in §4.2.1, and numerical experiments using this algorithm are found in §4.2.2.
4.2.1. A Strongly Polynomial-Time Algorithm for Two Assortments. At a high level, our algorithm for answering the identification question in the case of $M = 2$ consists of reducing the robust assortment optimization problem (RO) to solving a sequence of minimum-cost network flow problems. As discussed at the end of §2, an algorithm for solving the robust assortment optimization problem (RO) can be immediately used to answer the identification question, since there is an affirmative answer to the identification question if and only if the optimal objective value of the robust assortment optimization problem (RO) is strictly greater than $\max_{m \in M} r^T v_m$. Stated formally, the main contribution of §4.2.1 is the following:

**Theorem 3.** If $M = 2$ and $\eta = 0$, then (RO) can be solved in $O(n^5 \log(nr_n))$ computation time.

In the above theorem and throughout the rest of §4.2.1, we assume that the revenues $r_1, \ldots, r_n$ are distinct, the products are sorted in ascending order by their revenue, and the revenues are represented as nonnegative integers. We also assume without any loss of generality that $n \in S_1 \cap S_2$ and that $S_1 \cup S_2 = N_0$.

The proof of Theorem 3 is found at the end of §4.2.1.

To prove Theorem 3, and to develop an algorithm with the desired computation time, we begin by establishing three intermediary results. The proofs of these intermediary results, denoted by Lemmas 8-10, are relatively straightforward and can be found in Appendix D. In our first intermediary result, denoted by Lemma 8, we develop a closed-form representation of the collection of assortments $\hat{S}$. This representation will be useful in the proof of Theorem 3 because it provides an efficient procedure for iterating over the assortments in $\hat{S}$. Moreover, the following Lemma 8 is useful because it immediately implies that the number of assortments in the collection $\hat{S}$ scales quadratically in the number of products $n$.

**Lemma 8.** If $M = 2$, then

$$\hat{S} = \left\{ S \in S : \begin{array}{l}
\text{there exists } i_1 \in (S_1 \setminus S_2) \cup \{n\} \text{ and } i_2 \in (S_2 \setminus S_1) \cup \{n\} \text{ such that } \\
S = (S_1 \cap S_2) \cup \{j \in S_1 \setminus S_2 : j \geq i_1\} \cup \{j \in S_2 \setminus S_1 : j \geq i_2\} \end{array} \right\}. \quad (1)$$

Stated concretely, Lemma 8 shows that the assortments in the collection $\hat{S}$ can be parameterized by the pairs of products from the sets $(S_1 \setminus S_2) \cup \{n\}$ and $(S_2 \setminus S_1) \cup \{n\}$. Hence, the number of assortments in the collection $\hat{S}$ is at most $(|S_1 \setminus S_2| + 1) \times (|S_2 \setminus S_1| + 1) = O(n^2)$.

---

7 To see why the assumption that $n \in S_1 \cap S_2$ can be made without loss of generality, suppose for the sake of developing intuition that the product $n$ is not contained in $S_1 \cap S_2$. In that case, we can create a fictitious product with index $n + 1$ that is associated with any arbitrary revenue in the range $r_{n+1} = (r_n, \infty)$, and we can augment the historical sales data to be $v_1' \triangleq (v_1, 0) \in \mathbb{R}_+^{n+2}$ and $v_2' \triangleq (v_2, 0) \in \mathbb{R}_+^{n+2}$. It is straightforward to see that any feasible solution to (WC'-S) with the augmented historical sales data will satisfy $\lambda_{n+1,i_2} = \lambda_{i_1,n+1} = 0$ for all $i_1 \in S_1$ and $i_2 \in S_2$ that satisfy $(n+1, i_2) \in \mathcal{L}$ and $(i_1, n+1) \in \mathcal{L}$. This implies that the optimal objective value of (WC'-S) will be unchanged using the augmented historical sales data for all assortments $\hat{S} \in \hat{S}$. Hence, we have shown that we can assume without loss of generality that the previously-offered assortments satisfy $n \in S_1 \cap S_2$. The assumption that $S_1 \cup S_2 = N_0$ can be made without loss of generality due to similar reasoning.
In our second intermediary result, denoted by Lemma 9, we develop a closed-form representation of the set of tuples of products $L$. The following representation of $L$ will be useful in the proof of Theorem 3 because it will allow us to show that the worst-case expected revenue $\min_{\lambda \in U} R^\lambda(S)$ for each assortment $S \in S$ can be computed by solving a minimum-cost network flow problem.

**Lemma 9.** If $M = 2$, then $L = ((S_1 \setminus S_2) \times S_2) \cup (S_1 \times (S_2 \setminus S_1)) \cup \{(i, i) : i \in S_1 \cap S_2\}$.

In our third and final intermediary result, denoted by Lemma 10, we show that the worst-case expected revenue $\min_{\lambda \in U} R^\lambda(S)$ for each assortment $S \in S$ can be computed by solving a minimum-cost network flow problem.

**Lemma 10.** If $M = 2$ and $\eta = 0$, then the following equality holds for all assortments $S \in S$:

$$\min_{\lambda \in U} R^\lambda(S) = \sum_{i \in S_1 \cap S_2} \rho_{ii}(S)v_{1,1} +$$

$$\min_{\lambda} \begin{bmatrix} \sum_{i_1 \in S_1 \setminus S_2} \sum_{i_2 \in S_1 \cap S_2} \rho_{i_1i_2}(S)\lambda_{i_1i_2} \\ + \sum_{i_1 \in S_1 \setminus S_2} \sum_{i_2 \in S_2 \setminus S_1} \rho_{i_1i_2}(S)\lambda_{i_1i_2} \\ + \sum_{i_1 \in S_1 \cap S_2} \sum_{i_2 \in S_2 \setminus S_1} (\rho_{i_1i_2}(S) - \rho_{i_1i_1}(S))\lambda_{i_1i_2} \end{bmatrix}$$

subject to

$$\sum_{i_2 \in S_2} \lambda_{i_1i_2} = v_{1,1} \quad \forall i_1 \in S_1 \setminus S_2$$

$$\sum_{i_1 \in S_1} \lambda_{i_1i_2} = v_{2,1} \quad \forall i_2 \in S_2 \setminus S_1$$

$$\sum_{i_1 \in S_1 \setminus S_2} \lambda_{i_1i_2} = \sum_{i_2 \in S_2 \setminus S_1} \lambda_{i_2i_2} = v_{2,1} - v_{1,1} \quad \forall i \in S_1 \cap S_2$$

$$\lambda_{i_1i_2} \geq 0 \quad \forall (i_1, i_2) \in L.$$  

We readily observe that the linear optimization problem on line (2) is a minimum-cost network flow problem, where each decision variable $\lambda_{i_1i_2}$ corresponds to the flow on a directed edge from vertex $i_1$ to vertex $i_2$ (Ahuja et al. 1988, p. 296). In particular, we observe that the minimum-cost network flow problem on line (2) takes place on a complete tripartite directed acyclic graph with $|S_1 \setminus S_2| + |S_2 \setminus S_1| + \n_1 \cap S_2| = n + 1$ vertices and $|S_1 \setminus S_2| \times |S_2| + |S_2 \setminus S_1| \times |S_1 \cap S_2| = O(n^2)$ directed edges. In Figure 4, we present a visualization of the network corresponding to the minimum-cost network flow problem from line (2).

Using the above three intermediary results, we conclude our proof of Theorem 3 by presenting an algorithm for solving the robust assortment optimization problem (RO) and establishing its running time.
Figure 4 Visualization of minimum-cost network flow problem from line (2).

Note. The figure shows a visualization of the minimum-cost network flow problem corresponding to the linear optimization problem on line (2) for the case where the previously-offered assortments are $S_1 = \{0, 1, 2, 5\}$ and $S_2 = \{0, 3, 4, 5\}$. We see that there is a vertex in the graph for each product $i \in N_0 \equiv \{0, 1, 2, 3, 4, 5\}$. The graph is a complete tripartite directed graph, where the three partitions of vertices are denoted by the dotted ellipses and correspond to $S_1 \setminus S_2 = \{1, 2\}$, $S_1 \cap S_2 = \{0, 5\}$, and $S_2 \setminus S_1 = \{3, 4\}$. The flow demands at each of the vertices and the flow cost for each of the directed edges can be found on line (2).

Proof of Theorem 3. We first describe our algorithm for solving the robust assortment optimization problem (RO), which follows a brute-force strategy. Namely, our algorithm iterates over each of the assortments $S \in \mathcal{S}$, and, for each such assortment, the algorithm computes the corresponding worst-case expected revenue $\min_{\lambda \in U} R^\lambda(S)$. The algorithm concludes by returning the maximum value of $\min_{\lambda \in U} R^\lambda(S)$ across the assortments $S \in \mathcal{S}$. The correctness of this algorithm for solving the robust assortment optimization problem (RO) follows immediately from Theorem 1.

We now analyze the running time of our algorithm by using our three intermediary results. We assume that the two assortments $S_1$ and $S_2$ are given as sorted arrays. Under this assumption, it is straightforward to see that the sets $S_1 \cap S_2$, $S_1 \setminus S_2$, and $S_2 \setminus S_1$ can be computed and stored as sorted arrays in $\mathcal{O}(n)$ computation time. We also require $\mathcal{O}(n)$ computation time to store copies of the sets $S_1, S_2, S_1 \cap S_2, S_1 \setminus S_2$, and $S_2 \setminus S_1$ in hash tables, which ensures that querying whether a given product is an element of any of these sets can be performed in $\mathcal{O}(1)$ time.

We next analyze the computation times for constructing the collection of assortments $\mathcal{S}$, constructing the set of pairs of products $L$, and computing the quantities $\rho_{i_1i_2}(S)$ for each assortment $S \in \mathcal{S}$ and each pair of products $(i_1, i_2) \in L$. Indeed, using the aforementioned data structures, it follows readily from Lemma 8 that we can construct the collection of assortments $\mathcal{S}$ in $\mathcal{O}(n^3)$.
computation time.\footnote{It follows from Lemma 8 that we can efficiently iterate over the assortments in $\hat{S}$ by iterating over the pairs of products in $(S_1 \setminus S_2) \cup \{n\}$ and $(S_2 \setminus S_1) \cup \{n\}$. Constructing the collection $\hat{S}$ thus requires iterating over the $(|S_1 \setminus S_2| + 1) \times (|S_2 \setminus S_1| + 1) = O(n^2)$ assortments, and each of the assortments is comprised of at most $O(n)$ products.} Moreover, it follows from Lemma 9 that the set of pairs of products $L$ can be computed in $O(n^2)$ time. Finally, we analyze the computation times for computing the quantities $\varphi_{s_{i_1} s_{i_2}}(S)$ for each assortment $S \in \hat{S}$ and each pair of products $(i_1, i_2) \in L$. Indeed, we recall from Lemma 8 that $|\hat{S}| = O(n^2)$, and we recall from Lemma 9 that $|L| = O(n^2)$. Therefore, there are $|\hat{S}| \times |L| = O(n^4)$ different ways of choosing an assortment $S \in \hat{S}$ and a pair of products $(i_1, i_2) \in L$. For each assortment $S \in \mathcal{S}$ and pair of products $(i_1, i_2) \in \mathcal{L}$, it follows readily from Definition 3 and Lemmas 8 and 9 that

$$
\varphi_{s_{i_1} s_{i_2}}(S) = \begin{cases} 
\rho_{i_1} & \text{if } i_1, i_2 \in S_1 \cap S_2, i_1 = i_2, \text{ and } i_1 \in S, \\
0 & \text{if } i_1, i_2 \in S_1 \cap S_2, i_1 = i_2, \text{ and } i_1 \notin S, \\
\rho_{i_2} & \min \{ \rho_{i_1}, \min_{j \in S \cap S_1 \setminus S_2} r_j \}, \text{ if } i_1 \in S_1 \cap S_2, i_2 \in S_2 \setminus S_1, \text{ and } i_2 \notin S, \text{ and } i_1 \in S, \\
0 & \min \{ \rho_{i_1}, \min_{j \in S \cap S_2 \setminus S_1} r_j \}, \text{ if } i_1 \in S_1 \cap S_2, i_2 \in S_2 \setminus S_1, i_1 \notin S, \text{ and } i_1 \notin S, \\
f_{i_1, i_2} & \min \{ f_{i_1}, \min_{j \in S \cap S_1 \setminus S_2} r_j \}, \text{ if } i_1 \in S_1 \setminus S_2, i_2 \in S_2 \setminus S_1, i_1 \notin S, \text{ and } i_2 \notin S, \\
f_{i_1, i_2} & \min \{ f_{i_1}, \min_{j \in S \cap S_2 \setminus S_1} r_j \}, \text{ if } i_1 \in S_1 \setminus S_2, i_2 \in S_2 \setminus S_1, i_1 \notin S, \text{ and } i_2 \notin S.
\end{cases}
$$

We observe that the quantities $\min_{j \in S \cap S_1 \setminus S_2} r_j$ and $\min_{j \in S \cap S_2 \setminus S_1} r_j$ appear in many of the above cases, and we see that these quantities can be precomputed for each of the assortments $S \in \hat{S}$ in a total of $|\hat{S}| \times O(n) = O(n^3)$ computation time. Given that we have precomputed these quantities, and given the fact that our data structures allow us to query whether any product is an element of the sets $S_1 \setminus S_2$, $S_2 \setminus S_1$, and $S_1 \cap S_2$ in $O(1)$ time, we conclude that all of the $\varphi_{s_{i_1} s_{i_2}}(S)$ can be computed in a total of $O(n^4 + n^3) = O(n^4)$ time. In summary, we have established that constructing the collection of assortments $\hat{S}$, constructing the set of pairs of products $L$, and computing the quantities $\varphi_{s_{i_1} s_{i_2}}(S)$ for each assortment $S \in \hat{S}$ and each pair of products $(i_1, i_2) \in L$ can be performed in a total of $O(n^4)$ computation time.

We conclude our proof of Theorem 3 by establishing the total computation time of our brute-force algorithm using the information computed above. In each iteration of our algorithm, we select an assortment $S \in \hat{S}$ and compute the worst-case expected revenue $\min_{x \in \mathcal{L}} E^\lambda(S)$. As shown in Lemma 10, we can compute $\min_{x \in \mathcal{L}} E^\lambda(S)$ by solving a minimum-cost network flow problem over
a graph with \( n + 1 \) vertices and \( O(n^2) \) edges. Using the minimum-cost network flow algorithm of Orlin (1997) and Tarjan (1997), we observe that Problem (2) can be solved in \( O(n^3 \log(nr_n)) \) computation time.\(^9\) Since the worst-case expected revenue \( \min_{\lambda \in \mathcal{U}} R^\lambda(S) \) must be computed for each assortment \( S \in \hat{\mathcal{S}} \), and since it follows readily from Lemma 8 that \( |\hat{\mathcal{S}}| = O(n^2) \), we conclude that our algorithm requires a total of \( O(n^5 \log(nr_n)) \) computation time. \( \square \)

4.2.2. Numerical Experiments for Two Assortments. We will now perform numerical experiments to show, using our polynomial-time algorithm from §4.2.1, that there can be affirmative answers to the identification question in the case of \( M = 2 \).

To perform our numerical experiments, we begin by constructing randomly-generated problem instances in a manner that is similar to that taken in §4.1. In each randomly-generated problem instance, the revenues are drawn from the distribution \( r_1, \ldots, r_n \sim \text{Uniform}[0,1] \), and we sort the products such that \( r_1 < \cdots < r_n \). The two past assortments \( S_1, S_2 \in \mathcal{S} \) are also constructed randomly, whereby the two assortments satisfy \( \{0, n\} \subseteq S_1 \cap S_2 \) and, for each of the remaining products \( j \in \{1, \ldots, n-1\} \), we randomly assign the product to the assortments with distribution given by \( P(j \in S_1 \cap S_2) = 1/3 \), \( P(j \in S_1 \setminus S_2) = 1/3 \), and \( P(j \in S_2 \setminus S_1) = 1/3 \). In order to generate historical sales data for these two assortments, we generate a base choice for the parameters \( \lambda^* \) of the ranking-based choice model. Because we will be performing numerical experiments on problem instances with larger values of \( n \) than were considered in §4.1, it will not be viable from a computational tractability standpoint to generate base parameters \( \lambda^* \) that have nonzero values for each of the \( (n+1)! \) parameters of a ranking-based choice model. To get around this, we restrict the numerical experiments to generating base parameters \( \lambda^* \) which are sparse. Specifically, we generate the base parameters in each problem instance by first randomly selecting a subset of rankings \( \Sigma' \subseteq \Sigma \) of length \( |\Sigma'| = K \);\(^10\) we then assign \( \lambda^*_\sigma \leftarrow 0 \) for each ranking \( \sigma \notin \Sigma' \), and we choose the remaining parameters \( \{\lambda_\sigma : \sigma \in \Sigma'\} \) by drawing uniformly over the \( K \)-dimensional probability simplex. Using this base choice for the parameters, we generate historical sales data of the form \( v_1 \) and \( v_2 \) corresponding to the two assortments \( S_1 \) and \( S_2 \) under the ranking-based choice model with the base parameters \( \lambda^* \). After we compute the historical sales data, we ignore the base parameters \( \lambda^* \) of the ranking-based choice model as well as the choice of \( K \), and we apply the algorithm from §4.2.1 to obtain a new assortment, denoted by \( S' \).

---

\(^9\) The algorithm of Orlin (1997) and Tarjan (1997) computes the minimum-cost network flow on a directed graph in \( O((VE \log V) \min \{\log(VC), E \log V\}) \) running time, where \( V \) is the number of vertices, \( E \) is the number of directed edges, and \( C \) is the maximum absolute value of any edge cost. The algorithm requires that \( C \) is integral; for more details, see Tarjan (1997, §3). In our case, Problem (2) is a minimum-cost network flow problem in a directed graph where \( V = n \), \( E = O(n^2) \), and \( C = \max_{S \in \hat{\mathcal{S}}, (i,j) \in E} p_{i,j}(S) = r_n \).

\(^10\) We use rejection sampling to ensure that each of the \( \binom{n+1}{k} \) subsets of rankings is selected with equal probability.
Figure 5  Performance of new assortments obtained using algorithm from §4.2.1 when $M = 2$, $K = 10$, and $n = 10$.

Note. Each point corresponds to a randomly-generated problem instance. The $x$-axis shows the expected revenue of the best previously-offered assortment, $\max_{m \in \mathcal{M}} r^T v_m$. The $y$-axis shows the worst-case expected revenue $\min_{\lambda \in \mathcal{U}} R^{\lambda}(S')$ for the new assortment $S'$ obtained by using the algorithm from §4.2.1.

In Figures 5 and 6, we present the results of the numerical experiments conducted as described above. In Figure 5, we present the results of these numerical experiments in the case of $n = 10$ products over 1000 randomly-generated problem instances and for $K = 10$. This figure compares the expected revenue under the best previously-offered assortment, $\max_{m \in \mathcal{M}} r^T v_m$, to the predicted expected revenue of the new assortment under the worst-case ranking-based choice model that is consistent with the historical sales data, $\min_{\lambda \in \mathcal{U}} R^{\lambda}(S')$. These results establish that there can be affirmative answers to the identification question in the case of $M = 2$; indeed, we observe from Figure 5 that there are problem instances for which the predicted expected revenue of the new assortment obtained by our algorithm from §4.2.1 is strictly greater than the expected revenue of the best previously-offered assortment under all ranking-based choice model that are consistent with the historical sales data. In Figure 6, we show the average computation times for our algorithm from §4.2.1 on problem instances in which the number of products is varied across $n \in \{10, 12, 14, \ldots, 100\}$ and $K = 1000$. Here, we use the larger $K = 1000$ to ensure that the sparsity of the randomly-generated base parameters $\lambda^*$ does not introduce any biases on the resulting computation times of our algorithm. The results in Figure 6 show that the computation time of the algorithm remains
under 30 seconds even when there are one hundred products. This is viewed as promising from a practical perspective, as it shows that a general algorithm for answering the identification question for the case of $M = 2$ can scale to problem instances with realistic numbers of products.

Motivated by the numerical findings in Figure 5, we further conducted an informal analysis of the randomly-generated problem instances in which the worst-case expected revenue of the new assortment was strictly greater than the expected revenue of the best previously-offered assortment. Our analysis consisted of a visual examination of the assortments that were found by the algorithm from §4.2.1, along with an exploratory analysis in which we manually adjusted the parameters which were used to generate the problem instances to understand their impact on the resulting assortments found by the algorithm from §4.2.1. Our goal in conducting this informal analysis was to gain a qualitative understanding of the types of problem instances for which the algorithm from §4.2.1 is able to find an affirmative answer to the identification question. We were also interested in assessing the performance of the estimate-then-optimize technique in problem instances in which there are affirmative answers to the identification question.

The main takeaway from our informal analysis is a simple but insightful example of a problem instance for which the identification question has an affirmative answer. The example, which is presented in Appendix E, is insightful for three specific reasons. First, the example shows that there
are problem instances in which the assortment \( S \in \hat{S} \) that answers the identification question is not a revenue-ordered assortment. Second, the example shows that the estimate-then-optimize technique can perform poorly (that is, yield an assortment with an expected revenue which is strictly less than the best expected revenue from the previously-offered assortments) even on problem instances for which there are affirmative answers to the identification question. Third, the example shows that the aforementioned two properties can be obtained in relatively simple problem instances which are comprised of only \( n = 4 \) products and in which the revenues and historical sales data have numerical values which are relatively easy to manipulate. More broadly, this example illustrates that assortments which are associated with affirmative answers to the identification question can in general have a non-trivial structure, and discovering such assortments appears to be challenging without the aid of algorithms like those developed in §4.2.1 and in the subsequent §4.3.

4.3. Tractability of Identification Question for Fixed Number of Past Assortments

We conclude §4 by studying whether it is possible to design theoretically-efficient algorithms for answering to the identification question in general settings with several previously-offered assortments. Establishing the existence of such algorithms is important because their existence suggests that practical algorithms, like that from §4.2.1, can be developed for answering the identification question in real-world applications with relatively small numbers of previously-offered assortments and large numbers of products. Problem instances in which there are small numbers of previously-offered assortments and large numbers of products can arise in high-stakes applications in which changing to low-quality assortments can have significant negative consequences, and so the firm has made relatively few changes to their assortments up to this point.

In this subsection, our results are positive and consist of developing the first polynomial-time algorithm for answering the identification question for any fixed number of previously-offered assortments \( M \). The algorithm developed in this subsection is particularly attractive due to its generality: it does not require any assumptions on the composition of products in the previously-offered assortments. Moreover, the algorithm allows for the radius \( \eta \) in the set of ranking-based choice models to be strictly positive (see §2); hence, the algorithm can be applied in settings in which there are no ranking-based choice models that have perfect accuracy on the firm’s historical sales data. To the best of our knowledge, the general algorithm developed in this subsection is also the first for solving robust assortment optimization problems over the data-driven uncertainty set proposed by FJS13 with computation time that is polynomial in the number of products \( n \).

Our algorithm for answering the identification question in general settings follows the same high-level strategy as developed in §4.2. Specifically, our algorithm reduces the identification question to
solving the robust assortment optimization problem (RO) and checking to see if the optimal objective value of (RO) is strictly greater than the best expected revenue among the previously-offered assortments. Our algorithm solves the robust assortment optimization problem by computing the worst-case expected revenue \( \min_{\lambda \in \mathcal{U}} \mathcal{R}^\lambda(S) \) for each candidate assortment \( S \in \hat{S} \) and outputting the assortment that has the maximum worst-case expected revenue. The correctness of such a brute-force algorithm for solving the robust assortment optimization problem (RO) follows immediately from Theorem 1. In the remainder of §4.3, we prove that there exists an implementation of the aforementioned brute-force algorithm which yields the following theoretical guarantee on its computational tractability:

**Theorem 4.** (RO) can be solved in weakly \( \mathcal{O}(\text{poly}(n)) \) computation time for every fixed \( M \).

In the above theorem and throughout the rest of §4.3, we assume that the revenues \( r_1, \ldots, r_n \) are distinct. The above theorem holds for any choice of the radius \( \eta \geq 0 \) in the set of ranking-based choice models and any composition of products in the previously-offered assortments. We note that our algorithm is guaranteed to run in weakly, as opposed to strongly, polynomial time due to its reduction to solving linear optimization problems.

To prove Theorem 4, we will make use of several intermediary results (Lemma 11-14), the proofs of which can be found in Appendix F. The primary workhorse in our proof of Theorem 4 is contained in the proof of our first intermediary result, denoted by Lemma 11, in which we develop an algorithm for constructing the collection of assortments \( \hat{S} \) from the previously-offered assortments. The algorithm which achieves the specified running time of the following lemma is based on dynamic programming over a compact graphical representation of the collection of assortments \( \hat{S} \). In particular, the algorithm can be viewed as attractive due to its mild dependence on the number of assortments in the collection \( \hat{S} \).

**Lemma 11.** The collection of assortments \( \hat{S} \) can be constructed in \( \mathcal{O}(n^2(M + |\hat{S}|)) \) time.

We observe that the algorithm which achieves the specified running time in Lemma 11 is efficient when the number of assortments in the collection \( \hat{S} \) is small. In fact, we have seen evidence up to this point that the number of assortments in the collection \( \hat{S} \) can indeed be small in special cases of problem instances: namely, we recall from §4.1 that \( |\hat{S}| = \mathcal{O}(n) \) when the previously-offered assortments are the revenue-ordered assortments, and we recall from §4.2 that \( |\hat{S}| = \mathcal{O}(n^2) \) when there are two previously-offered assortments. In the next intermediary result, denoted by Lemma 12, we develop a more general result along these lines. Specifically, the following lemma establishes that the number of assortments in the collection \( \hat{S} \) can be upper bounded by a polynomial of the
number of products \( n \) for any fixed number of previously-offered assortments \( M \). While the bound in the following lemma is not the tightest possible, the bound will be sufficient for its theoretical purpose in this subsection of proving Theorem 4.

Lemma 12. \( |\hat{S}| \leq (n + 2)^2M \).

The remaining intermediary results will be used in our proof of Theorem 4 to show that the worst-case expected revenue \( \min_{\lambda \in U} R^\lambda(S) \) can be computed for each assortment \( S \in \hat{S} \) in weakly polynomial time for every fixed number of previously-offered assortments \( M \). The following intermediary results accomplish this by showing for every fixed \( M \geq 2 \) and assortment \( S \in \hat{S} \) that the worst-case expected revenue \( \min_{\lambda \in U} R^\lambda(S) \) can be computed by solving a linear optimization problem that can be constructed in polynomial time (Lemma 13-14) and has a polynomial number of decision variables and constraints (see Remark 2 from §3.1).

Lemma 13. \( |L| = O(n^2M) \), and \( L \) can be constructed in \( O(Mn^{M+1}) \) computation time.

Lemma 14. For each assortment \( S \in \hat{S} \) and each tuple of products \((i_1, \ldots, i_M) \in L\), the quantity \( \rho_{i_1 \cdots i_M}(S) \) can be computed in \( O(M^2n) \) time.

Using the above intermediary lemmas, we conclude §4.3 with our proof of Theorem 4.

Proof of Theorem 4. As described at the beginning of §4.3, we consider a brute-force algorithm for solving the robust assortment optimization problem (RO) that consists of computing the worst-case expected revenue \( \min_{\lambda \in U} R^\lambda(S) \) for each candidate assortment \( S \in \hat{S} \) and outputting the assortment that has the maximum worst-case expected revenue. The computation time of this brute-force algorithm is analyzed as follows. In Lemmas 11 and 12, we established that the collection of assortments \( \hat{S} \) can be constructed in \( \mathcal{O}(n^2(M + |\hat{S}|)) = \mathcal{O}(n^2(M + (n + 2)^2M)) = \mathcal{O}(\text{poly}(n)) \) computation time, where the last equality holds for any fixed \( M \). In Lemma 13, we established that the set of tuples of products \( L \) can be constructed in \( \mathcal{O}(Mn^{M+1}) = \mathcal{O}(\text{poly}(n)) \) computation time, where the equality holds for any fixed \( M \). Given any assortment \( S \in \hat{S} \), we established in Lemma 14 that the quantities \( \rho_{i_1 \cdots i_M}(S) \) for each \((i_1, \ldots, i_M) \in L\) can all be computed in a total of \( \mathcal{O}(|L| \times M^2n) = \mathcal{O}(n^2 \times M^2n) = \mathcal{O}(\text{poly}(n)) \) computation time, where the last equality holds for any fixed \( M \). Given the set of tuples of products \( L \) and the quantities \( \rho_{i_1 \cdots i_M}(S) \) for each \((i_1, \ldots, i_M) \in L\), we established in Remark 2 from §3.1 that we can compute the worst-case expected revenue \( \min_{\lambda \in U} R^\lambda(S) \) by constructing and solving a linear optimization problem with \( \mathcal{O}(n^M) = \mathcal{O}(\text{poly}(n)) \) decision variables and \( \mathcal{O}(nM) = \mathcal{O}(\text{poly}(n)) \) constraints for any fixed \( M \). Since linear optimization can be solved in weakly polynomial time via the ellipsoid algorithm, we conclude that each iteration of our algorithm requires \( \mathcal{O}(\text{poly}(n)) \) time for any fixed \( M \). Our proof of Theorem 4 is thus complete. \( \square \)
5. Conclusion and Future Research

In this work, we investigated a popular class of high-dimensional discrete choice models, known as ranking-based choice models, in the context of assortment planning problems. Motivated by the fact that many ranking-based choice models can be consistent with a firm’s historical sales data, we considered the identification question, which asks whether it is possible to identify a new assortment that outperforms the firm’s past assortments under all of the ranking-based choice models that are consistent with the firm’s historical sales data. By analyzing and developing algorithms for solving a class of robust assortment optimization problems, we established the existence of affirmative as well as negative answers to the identification question. Moreover, we developed polynomial-time algorithms for answering the identification question in general settings with any fixed number of past assortments. Together with concise numerical experiments, these findings revealed that considering the identification question can be essential to making good assortment decisions from ranking-based choice models in high-stakes assortment planning problems.

We believe this work opens up a number of promising directions for future research. First, our work showed for the first time that the identification question can be answered for one popular class of high-dimensional discrete choice models. Yet there are many other high-dimensional discrete choice models beyond the ranking-based choice model for which the identification question can be asked, such as models for capturing irrational customer choice (Berbeglia 2018, Chen and Mišić 2021, Jena et al. 2021). Second, our work showed that polynomial-time algorithms can be developed for finding assortments that answer the identification question for ranking-based choice models. However, it may be possible in certain settings that existing algorithms for estimating high-dimensional discrete choice models such as expectation-maximization (Talluri and Van Ryzin 2004, van Ryzin and Vulcano 2017, Şimşek and Topaloglu 2018), in combination with algorithms for finding assortments that maximize the predicted expected revenue, can lead to assortments with provable performance guarantees with respect to the identification question. Establishing such guarantees would provide new assurances to firms for using estimate-then-optimize in high-stakes assortment planning problems. Finally, we believe that the present and related work (e.g., Kallus and Zhou (2018), Sturt (2021)) provide a starting point for using robust optimization to develop efficient algorithms that are valuable for operations management problems in which good average performance is paramount. In particular, future work may extend the algorithms developed in the present paper to answer the identification question in numerous other revenue management problems, ranging from multi-product pricing to dynamic assortment planning.
References

Ahuja RK, Magnanti TL, Orlin JB (1988) Network Flows (Pearson).

Aouad A, Farias V, Levi R (2021) Assortment optimization under consider-then-choose choice models. Management Science 67(6):3368–3386.

Aouad A, Farias V, Levi R, Segev D (2018) The approximability of assortment optimization under ranking preferences. Operations Research 66(6):1661–1669.

Berbeglia G (2018) The generalized stochastic preference choice model. arXiv preprint arXiv:1803.04244.

Berbeglia G, Joret G (2020) Assortment optimisation under a general discrete choice model: A tight analysis of revenue-ordered assortments. Algorithmica 82(4):681–720.

Bertsimas D, Mišić VV (2017) Robust product line design. Operations Research 65(1):19–37.

Bertsimas D, Mišić VV (2019) Exact first-choice product line optimization. Operations Research 67(3):651–670.

Block H, Marschak J (1959) Random orderings and stochastic theories of response. Cowles Foundation Discussion Papers 66, Cowles Foundation for Research in Economics, Yale University, URL https://EconPapers.repec.org/RePEc:cwl:cwldpp:66.

Chen YC, Mišić V (2021) Decision forest: A nonparametric approach to modeling irrational choice. Management Science (forthcoming).

Désir A, Goyal V, Jagabathula S, Segev D (2021a) Mallows-smoothed distribution over rankings approach for modeling choice. Operations Research 69(4):1206–1227.

Désir A, Goyal V, Jiang B, Xie T, Zhang J (2021b) Robust assortment optimization under the markov chain model. Available at SSRN.

Farias VF, Jagabathula S, Shah D (2013) A nonparametric approach to modeling choice with limited data. Management Science 59(2):305–322.

Feldman J, Paul A, Topaloglu H (2019) Assortment optimization with small consideration sets. Operations Research 67(5):1283–1299.

Fisher M, Vaidyanathan R (2012) Which products should you stock? Harvard Business Review 90(11):108–119.

Gallego G, Iyengar G, Phillips R, Dubey A (2004) Managing flexible products on a network. Available at SSRN 3567371.

Honhon D, Jonnalagedda S, Pan XA (2012) Optimal algorithms for assortment selection under ranking-based consumer choice models. Manufacturing & Service Operations Management 14(2):279–289.

Jagabathula S (2014) Assortment optimization under general choice. Available at SSRN 2512831.

Jagabathula S, Rusnevcijentong P (2019) The limit of rationality in choice modeling: Formulation, computation, and implications. Management Science 65(5):2196–2215.
Jena SD, Lodi A, Sole C (2021) On the estimation of discrete choice models to capture irrational customer behaviors. arXiv preprint arXiv:2109.03882.

Kallus N, Zhou A (2018) Confounding-robust policy improvement. Advances in Neural Information Processing Systems 31.

Mišić VV (2016) Data, models and decisions for large-scale stochastic optimization problems. Ph.D. thesis, Massachusetts Institute of Technology.

Neff J (2009) Walmart’s shopper-friendly redesign takes toll on sales. Advertising Age 80(35):1–1,26.

Orlin JB (1997) A polynomial time primal network simplex algorithm for minimum cost flows. Mathematical Programming 78(2):109–129.

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

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

Şimşek AS, Topaloglu H (2018) An expectation-maximization algorithm to estimate the parameters of the markov chain choice model. Operations Research 66(3):748–760.

Smith CM (2005) Origin and uses of primum non nocere—above all, do no harm! The Journal of Clinical Pharmacology 45(4):371–377.

Sturt B (2021) A nonparametric algorithm for optimal stopping based on robust optimization. arXiv preprint arXiv:2103.03300.

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

Tarjan RE (1997) Dynamic trees as search trees via euler tours, applied to the network simplex algorithm. Mathematical Programming 78(2):169–177.

van Ryzin G, Vulcano G (2015) A market discovery algorithm to estimate a general class of nonparametric choice models. Management Science 61(2):281–300.

van Ryzin G, Vulcano G (2017) An expectation-maximization method to estimate a rank-based choice model of demand. Operations Research 65(2):396–407.

Wang Z, Peura H, Wiesemann W (2020) Randomized assortment optimization. Available at SSRN 3685695.
Technical Proofs and Additional Results

Appendix A: Proofs of Intermediary Technical Results from §3.1

A.1. Proof of Lemma 1

Consider any ranking \( \sigma \in \Sigma \). It follows from Definition 1 that \( \sigma \in \bigcap_{m \in M} D_{i_m}(S_m) \) if and only if \( i_m = \arg \min_{j \in S_m} \sigma(j) \) for each previously-offered assortment \( m \in M \). Hence, we have shown that there is a unique tuple of products \( (i_1, \ldots, i_M) \) \( \in S_1 \times \cdots \times S_M \) that satisfies \( \sigma \in \bigcap_{m \in M} D_{i_m}(S_m) \). Moreover, it follows from the fact that \( \sigma \in \bigcap_{m \in M} D_{i_m}(S_m) \) that \( \bigcap_{m \in M} D_{i_m}(S_m) \neq \emptyset \), and so it follows from Definition 2 that \( (i_1, \ldots, i_M) \in \mathcal{L} \). Our proof of Lemma 1 is thus complete. \( \Box \)

A.2. Proof of Proposition 1

For every assortment \( S' \in \mathcal{S} \), product in the assortment \( i \in S' \), and ranking \( \sigma \in \Sigma \), it follows immediately from Definition 1 that the equality \( \mathbb{I} \{ i = \arg \min_{j \in S'} \sigma(j) \} = 1 \) is satisfied if and only if \( \sigma \in D_{i}(S') \). Therefore, for every assortment \( S \in \mathcal{S} \), we observe that the linear optimization problem (WC-\( S \)) is equivalent to

\[
\begin{align*}
\text{minimize} & \quad \sum_{i \in S} \left( \sum_{\sigma \in D_{i}(S)} \lambda_{\sigma} \right) r_i \\
\text{subject to} & \quad \sum_{\sigma \in D_{i}(S_m)} \lambda_{\sigma} = 1 \quad \forall m \in M \text{ and } i \in S_m \\
& \quad \sum_{\sigma \in \Sigma} \lambda_{\sigma} = 1 \\
& \quad \|\epsilon\| \leq \eta \\
& \quad \lambda_{\sigma} \geq 0 \quad \forall \sigma \in \Sigma.
\end{align*}
\]

(WC-\( S \)-1)

Moreover, we recall from Lemma 1 that there is a unique tuple of products \( (i_1, \ldots, i_M) \in \mathcal{L} \) corresponding to each ranking \( \sigma \in \Sigma \) that satisfies \( \sigma \in \bigcap_{m \in M} D_{i_m}(S_m) \). Therefore, it follows immediately from Lemma 1 that the linear optimization problem (WC-\( S \)-1) is equivalent to

\[
\begin{align*}
\text{minimize} & \quad \sum_{(i_1, \ldots, i_M) \in \mathcal{L}} \sum_{i \in S} r_i \left( \sum_{\sigma \in \bigcap_{m \in M} D_{i_m}(S_m) \cap D_{i}(S)} \lambda_{\sigma} \right) \\
\text{subject to} & \quad \sum_{(i_1, \ldots, i_M) \in \mathcal{L}: i_m = i} \left( \sum_{\sigma \in \bigcap_{m \in M} D_{i_m}(S_m) \cap D_{i}(S)} \lambda_{\sigma} \right) - \epsilon_{m,i} = v_{m,i} \quad \forall m \in M \text{ and } i \in S_m \\
& \quad \sum_{(i_1, \ldots, i_M) \in \mathcal{L}} \left( \sum_{\sigma \in \bigcap_{m \in M} D_{i_m}(S_m) \cap D_{i}(S)} \lambda_{\sigma} \right) = 1 \\
& \quad \|\epsilon\| \leq \eta \\
& \quad \lambda_{\sigma} \geq 0 \quad \forall \sigma \in \Sigma.
\end{align*}
\]

(WC-\( S \)-2)
We now simplify the linear optimization problem (WC-S-2) by performing a transformation on its decision variables. Specifically, we transform (WC-S-2) by creating the following new decision variables for each tuple of products \((i_1, \ldots, i_M) \in \mathcal{L}\) and each product \(i \in S\):

\[
\lambda_{i_1: \ldots: i_M} \leftarrow \sum_{\sigma \in \cap_{m \in M} D_{i_m}(S_m)} \lambda_{\sigma}; \quad \omega_{i_1: \ldots: i_M} \leftarrow \sum_{\sigma \in \cap_{m \in M} D_{i_m}(S_m) \cap D_i(S)} \lambda_{\sigma}.
\]

Let us make three observations about the new decision variables defined above. First, we observe immediately from the definition of the new decision variables \(\omega_{i_1: \ldots: i_M}\) that the equality \(\omega_{i_1: \ldots: i_M} = 0\) must hold for each tuple of products \((i_1, \ldots, i_M) \in \mathcal{L}\) and product \(i \in S\) that satisfy \(\cap_{m \in M} D_{i_m}(S_m) \cap D_i(S) = \emptyset\). Second, we observe that the nonnegativity constraints \(\lambda_{\sigma} \geq 0\) in the linear optimization problem (WC-S-2) imply that the new decision variables must also satisfy the nonnegativity constraints \(\lambda_{i_1: \ldots: i_M} \geq 0\) and \(\omega_{i_1: \ldots: i_M} \geq 0\). Third, we observe from the definitions of the new decision variables that the equality \(\sum_{i \in S} \omega_{i_1: \ldots: i_M} = \lambda_{i_1: \ldots: i_M}\) must hold for every tuple of products \((i_1, \ldots, i_M) \in \mathcal{L}\). Based on the definitions of the new decision variables and by applying the aforementioned three observations, we have shown that the linear optimization problem (WC-S-2) is equivalent to

\[
\begin{align*}
\text{minimize} & \quad \sum_{(i_1, \ldots, i_M) \in \mathcal{L}} \sum_{i \in S} r_i \omega_{i_1: \ldots: i_M} \\
\text{subject to} & \quad \omega_{i_1: \ldots: i_M} = \lambda_{i_1: \ldots: i_M}, \quad \forall (i_1, \ldots, i_M) \in \mathcal{L} \\
& \quad \sum_{(i_1, \ldots, i_M) \in \mathcal{L}, i_m = i} \lambda_{i_1: \ldots: i_M} - \epsilon_{m,i} = v_{m,i}, \quad \forall m \in M \text{ and } i \in S_m \\
& \quad \sum_{(i_1, \ldots, i_M) \in \mathcal{L}} \lambda_{i_1: \ldots: i_M} = 1 \\
& \quad \|\epsilon\| \leq \eta \\
& \quad \lambda_{i_1: \ldots: i_M} \geq 0, \quad \forall (i_1, \ldots, i_M) \in \mathcal{L} \\
& \quad \omega_{i_1: \ldots: i_M} \geq 0, \quad \forall i \in S \text{ and } (i_1, \ldots, i_M) \in \mathcal{L} \\
& \quad \omega_{i_1: \ldots: i_M} = 0, \quad \forall i \in S \text{ and } (i_1, \ldots, i_M) \in \mathcal{L} \text{ such that } \cap_{m \in M} D_{i_m}(S_m) \cap D_i(S) = \emptyset.
\end{align*}
\]

(WC-S-3)

After rearranging terms, we observe that the linear optimization problem (WC-S-3) is equivalent to

\[
\begin{align*}
\text{minimize} & \quad \sum_{(i_1, \ldots, i_M) \in \mathcal{L}} \left[ \sum_{i \in S} r_i \omega_{i_1: \ldots: i_M} \right] \\
\text{subject to} & \quad \sum_{i \in S} \omega_{i_1: \ldots: i_M} = \lambda_{i_1: \ldots: i_M} \\
& \quad \omega_{i_1: \ldots: i_M} \geq 0, \quad \forall i \in S \\
& \quad \omega_{i_1: \ldots: i_M} = 0, \quad \forall i \in S \text{ such that } \cap_{m \in M} D_{i_m}(S_m) \cap D_i(S) = \emptyset \\
& \quad \sum_{(i_1, \ldots, i_M) \in \mathcal{L}, i_m = i} \lambda_{i_1: \ldots: i_M} - \epsilon_{m,i} = v_{m,i}, \quad \forall m \in M \text{ and } i \in S_m \\
& \quad \sum_{(i_1, \ldots, i_M) \in \mathcal{L}} \lambda_{i_1: \ldots: i_M} = 1 \\
& \quad \|\epsilon\| \leq \eta \\
& \quad \lambda_{i_1: \ldots: i_M} \geq 0, \quad \forall (i_1, \ldots, i_M) \in \mathcal{L}.
\end{align*}
\]

(WC-S-4)
Finally, we readily observe from Definition 3 that the following equality holds for every \((i_1, \ldots, i_M) \in L\):

\[
\begin{align*}
\text{minimize} & \quad \sum_{i \in S} r_i \omega_{i_1, i_2, \ldots, i_M} \\
\text{subject to} & \quad \sum_{i \in S} \omega_{i, i_1, i_2, \ldots, i_M} = \lambda_{i_1, i_2, \ldots, i_M} \\
& \quad \omega_{i, i_1, i_2, \ldots, i_M} \geq 0 \quad \forall i \in S \\
& \quad \omega_{i, i_1, i_2, \ldots, i_M} = 0 \quad \forall i \in S \text{ such that } \cap_{m \in M} D_{i_m}(S_m) \cap D_i(S) = \emptyset
\end{align*}
\]

It follows from the above equality that (WC-S-4) is equivalent to the linear optimization problem (WC'-S), which concludes our proof of Proposition 1. \(\square\)

**Appendix B: Proofs of Intermediary Technical Results from §3.2**

**B.1. Proof of Lemma 2**

Consider any tuple of products \((i_1, \ldots, i_M) \in S_1 \times \cdots \times S_M\). First, we recall the fundamental result in graph theory that a directed graph is acyclic if and only if there exists a ranking \(\sigma \in \Sigma\) that satisfies \(\sigma(i) < \sigma(j)\) for every directed edge \((j, i)\) in the graph (Ahuja et al. 1988, p. 77). Second, we recall from our construction of \(G_{i_1, \ldots, i_M}\) that the directed edges in \(G_{i_1, \ldots, i_M}\) are exactly those of the form \((j, i_m)\) for each previously-offered assortment \(m \in M\) and product \(j \in S_m \setminus \{i_m\}\). Combining these two recollections, we see that the directed graph \(G_{i_1, \ldots, i_M}\) is acyclic if and only if there exists a ranking \(\sigma \in \Sigma\) that satisfies \(\sigma(i_m) < \sigma(j)\) for each \(m \in M\) and \(j \in S_m \setminus \{i_m\}\). Since

\[
[\sigma(i_m) < \sigma(j) \text{ for all } m \in M \text{ and } j \in S_m \setminus \{i_m\}] \iff \left[ i_m = \arg \min_{j \in S_m} \sigma(j) \right] \iff [\sigma \in \cap_{m \in M} D_{i_m}(S_m)],
\]

we conclude that \(G_{i_1, \ldots, i_M}\) is acyclic if and only if \((i_1, \ldots, i_M) \in L\). \(\square\)

**B.2. Proof of Lemma 4**

Let \(i, j \in N_0\) and \((i_1, \ldots, i_M) \in L\), in which case it follows from Lemma 2 that the directed graph \(G_{i_1, \ldots, i_M}\) is acyclic.

To show the first direction of the desired result, let us suppose that \(j \not\sim_{i_1, \ldots, i_M} i\). In this case, we observe that the directed edge \((j, i)\) can be added to \(G_{i_1, \ldots, i_M}\) without inducing any cycles. For notational convenience, let this augmented graph with the directed edge \((j, i)\) be denoted by \(\tilde{G}_{i_1, \ldots, i_M}\). Since the augmented graph \(\tilde{G}_{i_1, \ldots, i_M}\) is acyclic, it follows from Aluja et al. (1988, p. 77) that there exists a ranking \(\sigma \in \Sigma\) that satisfies \(\sigma(j') < \sigma(i')\) for all directed edges \((i', j')\) in the augmented graph. Moreover, since the set of directed edges in the augmented graph \(\tilde{G}_{i_1, \ldots, i_M}\) is a superset of the set of directed edges in the original graph \(G_{i_1, \ldots, i_M}\), it follows from the construction of the original graph that this ranking satisfies \(\sigma(i_m) < \sigma(i')\) for all \(m \in M\) and all \(i' \in S_m \setminus \{i_m\}\), which implies that \(\sigma \in \cap_{m \in M} D_{i_m}(S_m)\). Since the augmented graph has the directed edge \((j, i)\), we observe that this ranking satisfies \(\sigma(i) < \sigma(j)\). In summary, we have shown that if \(j \not\sim_{i_1, \ldots, i_M} i\), then there exists a ranking \(\sigma \in \cap_{m \in M} D_{i_m}(S_m)\) that satisfies \(\sigma(i) < \sigma(j)\). This concludes our proof of the first direction of Lemma 4.
To show the other direction, let us suppose that \( j \prec_{i_1 \ldots i_M} i \). In this case, consider any arbitrary ranking \( \sigma \in \Sigma \) that satisfies \( \sigma \in \cap_{m \in M} D_{i_m}(S_m) \). It follows from the fact that \( \sigma \in \cap_{m \in M} D_{i_m}(S_m) \) and from the construction of \( G_{i_1 \ldots i_M} \) that the inequality \( \sigma(j') < \sigma(i') \) holds for each directed edge \((i', j')\) in this graph. Moreover, without loss of generality, let the directed path from vertex \( i \) to vertex \( j \) be denoted by the sequence of vertices \( i, j_1, \ldots, j_{\nu}, j \) which satisfies the property that \((i, j_1), (j_1, j_2), \ldots, (j_{\nu-1}, j_{\nu}), (j_{\nu}, j)\) are directed edges in \( G_{i_1 \ldots i_M} \). It then follows from our earlier reasoning that \( \sigma(i) > \sigma(j_1) > \cdots > \sigma(j_{\nu}) > \sigma(j) \).

Since the ranking \( \sigma \in \cap_{m \in M} D_{i_m}(S_m) \) was chosen arbitrarily, our proof of the other direction is complete.

\[ \square \]

### B.3. Proof of Lemma 5

Consider any tuple of products \((i_1, \ldots, i_M) \in \mathcal{L}\), assortment \( S \in \mathcal{S}\), and product \( i \in \mathcal{S}\).

To prove the first direction of Lemma 5, suppose there exists a product \( j \in S \) that satisfies \( j \prec_{i_1 \ldots i_M} i \). In this case, it follows immediately from Lemma 3 that there exists a previously-offered assortment \( m \in \mathcal{M}\) that satisfies \( i_m \in S \) and \( i_m \prec_{i_1 \ldots i_M} i \). Therefore, for each ranking \( \sigma \in \cap_{m \in M} D_{i_m}(S_m) \), it follows from Lemma 4 that \( \sigma(i_m) < \sigma(i) \), which combined with the fact that \( i_m \in S \) implies that \( i \neq \arg \min_{j \in S} \sigma(j) \). That concludes our proof of the first direction.

To prove the other direction, suppose that \( i \neq \arg \min_{j \in S} \sigma(j) \) for all rankings \( \sigma \in \cap_{m \in M} D_{i_m}(S_m) \). In this case, it follows immediately from Definition 1 that \( \cap_{m \in M} D_{i_m}(S_m) \cap D_i(S) = \emptyset \). Moreover, it follows from the fact that \((i_1, \ldots, i_M) \in \mathcal{L}\) and from Lemma 2 that the directed graph \( G_{i_1 \ldots i_M} \) is acyclic. Therefore, we observe from the fact that \( \cap_{m \in M} D_{i_m}(S_m) \cap D_i(S) = \emptyset \) and from Lemma 2 that adding the directed edges \( \{(j, i) : j \in S \setminus \{i\}\} \) to the directed acyclic graph \( G_{i_1 \ldots i_M} \) would result in a directed cycle. Since a cycle visits every vertex at most once, there must exist a single vertex \( j \in S \setminus \{i\} \) such that adding only the directed edge \((j, i)\) to \( G_{i_1 \ldots i_M} \) would result in a directed cycle, which implies that \( j \prec_{i_1 \ldots i_M} i \). This concludes our proof of the other direction.

### Appendix C: Proofs of Intermediary Technical Results from §3.3

#### C.1. Proof of Lemma 6

Consider any assortment \( S \in \mathcal{S}\), tuple of products \((i_1, \ldots, i_M) \in \mathcal{L}\), and product \( i \notin \mathcal{S}\). We first observe that

\[
I_{i_1 \ldots i_M}(S \cup \{i\}) = \{ j \in N_0 : \text{for all } m \in \mathcal{M} \text{, if } i_m \in S \cup \{i\} \text{, then } i_m \neq i_{i_1 \ldots i_M} j \} \\
= \{ j \in N_0 : \text{for all } m \in \mathcal{M} \text{, if } i_m \in S \text{, then } i_m \neq i_{i_1 \ldots i_M} j \} \\
\cap \{ j \in N_0 : \text{if there exists } m \in \mathcal{M} \text{ such that } i = i_m \text{, then } i \neq i_{i_1 \ldots i_M} j \} \\
= I_{i_1 \ldots i_M}(S) \cap \{ j \in N_0 : \text{if there exists } m \in \mathcal{M} \text{ such that } i = i_m \text{, then } i \neq i_{i_1 \ldots i_M} j \} \\
= \begin{cases} I_{i_1 \ldots i_M}(S), & \text{if } i \notin \{i_1, \ldots, i_M\} \\
I_{i_1 \ldots i_M}(S) \cap \{ j \in N_0 : i \neq i_{i_1 \ldots i_M} j \}, & \text{if } i \in \{i_1, \ldots, i_M\} \\
I_{i_1 \ldots i_M}(S) \cap \{ j \in N_0 : i \neq i_{i_1 \ldots i_M} j \} & \end{cases} \\
= I_{i_1 \ldots i_M}(S) \cap \{ j \in N_0 : i \neq i_{i_1 \ldots i_M} j \}. \tag{EC.1}
\]
Indeed, the first and third equalities follow from Definition 5. The second and fourth equalities follow from algebra. The final equality follows from the fact that if $i \notin \{i_1, \ldots, i_M\}$, then it follows from the construction of $G_{i_1 \cdots i_M}$ that there are no incoming edges to vertex $i$, which implies that $\{j \in N_0 : i \neq i_1 \cdots i_M j\} = N_0$.

Therefore, it follows from line (EC.1) and Proposition 2 that

$$\rho_{i_1 \cdots i_M} (S \cup \{i\}) = \min_{j \in (S \cup \{i\}) \cap \mathcal{I}_{i_1 \cdots i_M} (S \cup \{i\})} r_j$$

$$= \min_{j \in (S \cup \{i\}) \cap \mathcal{I}_{i_1 \cdots i_M} (S \cup \{i\}) \cap \{j' \in N_0 : i \neq i_1 \cdots i_M j'\}} r_j$$

$$= \min \left\{ \min_{j \in S \cap \mathcal{I}_{i_1 \cdots i_M} (S) \cap \{j' \in N_0 : i \neq i_1 \cdots i_M j'\}} r_j, \min_{j \in \mathcal{I}_{i_1 \cdots i_M} (S) \cap \{j' \in N_0 : i \neq i_1 \cdots i_M j'\}} r_j, r_i \right\}, \quad \text{if } i \notin \mathcal{I}_{i_1 \cdots i_M} (S), \quad \text{(EC.2)}$$

where the first equality follows from Proposition 2, the second equality follows from line (EC.1), and the last equality follows from algebra and the fact that $i \neq i_1 \cdots i_M i$.

We conclude the proof of Lemma 6 by considering the case where $i \notin \mathcal{I}_{i_1 \cdots i_M} (S)$. For this case, suppose for the sake of developing a contradiction that there exists $j \in S \cap \mathcal{I}_{i_1 \cdots i_M} (S)$ which satisfies $i \prec_{i_1 \cdots i_M} j$. Under that supposition, it would follow from the fact that $i \notin \mathcal{I}_{i_1 \cdots i_M} (S)$ and Definition 5 that there would exist a previously-offered assortment $m \in \mathcal{M}$ that satisfies $i_m \in S$ and $i_m \prec_{i_1 \cdots i_M} i$. Thus, by the transitive property, we have $i_m \prec_{i_1 \cdots i_M} \cdot \prec_{i_1 \cdots i_M} j$, which contradicts the supposition that $j \in \mathcal{I}_{i_1 \cdots i_M} (S)$. Because we have a contradiction, we have shown that $i \neq i_1 \cdots i_M j$ for all $j \in S \cap \mathcal{I}_{i_1 \cdots i_M} (S)$, and so the desired result follows immediately from line (EC.2) and Definition 3. □

### C.2. Proof of Lemma 7

Consider any assortment $S \in \mathcal{S}$ and any product $i \notin S$. Suppose that there exists a product $i^* \in S$ which satisfies $r_{i^*} < r_i$ and $\mathcal{M}_{i^*} \subseteq \mathcal{M}_i$. For each tuple of products $(i_1, \ldots, i_M) \in \mathcal{L}$, we have two cases to consider:

- **Case 1:** Suppose that $i \notin \mathcal{I}_{i_1 \cdots i_M} (S)$. In this case, it follows immediately from Lemma 6 that

$$\rho_{i_1 \cdots i_M} (S \cup \{i\}) = \rho_{i_1 \cdots i_M} (S),$$

and so the inequality $\rho_{i_1 \cdots i_M} (S) \leq \rho_{i_1 \cdots i_M} (S \cup \{i\})$ holds when $i \notin \mathcal{I}_{i_1 \cdots i_M} (S)$.

- **Case 2:** Suppose that $i \in \mathcal{I}_{i_1 \cdots i_M} (S)$.

In this case, we begin by showing that $i_m \notin S$ for each $m \in \mathcal{M}_i$. Indeed, consider any previously-offered assortment $m \in \mathcal{M}_i$. On one hand, if $i = i_m$, then it follows immediately from the fact that $i \notin \mathcal{I}_{i_1 \cdots i_M} (S)$ that $i_m \notin S$. On the other hand, if $i \neq i_m$, then it also must be the case that $i_m \notin S$, else we would have a contradiction with the fact that $i \in \mathcal{I}_{i_1 \cdots i_M} (S)$ and the fact that there is, by our construction of $G_{i_1 \cdots i_M}$, a directed edge from vertex $i$ to vertex $i_m$. We have thus shown that $i_m \notin S$ for all $m \in \mathcal{M}_i$.

We next show that $i_m \in \mathcal{I}_{i_1 \cdots i_M} (S)$ for all $m \in \mathcal{M}_i$. Indeed, consider any arbitrary $m \in \mathcal{M}_i$. On one hand, if $i_m = i$, then the statement $i_m \in \mathcal{I}_{i_1 \cdots i_M} (S)$ follows immediately from the fact that $i \in \mathcal{I}_{i_1 \cdots i_M} (S)$.

On the other hand, if $i_m \neq i$, then it follows from the fact that $i \in \mathcal{I}_{i_1 \cdots i_M} (S)$ and the fact that there
is a directed edge from vertex \(i\) to vertex \(i_m\) that there must not be a directed path from vertex \(i_m\) to a vertex \(i_m'\) that satisfies \(i_m', \in S\) for any \(m' \in \mathcal{M}\). We have thus shown that \(i_m \in \mathcal{I}_{i_1\ldots i_M}(S)\) for all \(m \in \mathcal{M}_i\).

Using the above results, we now prove that \(i^* \in \mathcal{I}_{i_1\ldots i_M}(S)\). Indeed, we have shown in the above results that \(i_m \notin S\) and \(i_m \in \mathcal{I}_{i_1\ldots i_M}(S)\) for all \(m \in \mathcal{M}_i\). Therefore, it follows from the supposition that \(\mathcal{M}_{i^*} \subseteq \mathcal{M}_i\) that \(i_m \notin S\) and \(i_m \in \mathcal{I}_{i_1\ldots i_M}(S)\) for all \(m \in \mathcal{M}_{i^*}\). Since we have supposed that \(i^* \in S\), it follows from the fact that \(i_m \notin S\) for all \(m \in \mathcal{M}_{i^*}\) that \(i^* \neq i_m\) for all \(m \in \mathcal{M}_{i^*}\). Moreover, it follows from the construction of \(\mathcal{G}_{i_1\ldots i_M}\) that all of the outgoing edges from vertex \(i^*\) are incoming edges to vertices \(i_m\) for \(m \in \mathcal{M}_{i^*}\). Therefore, it follows from the fact that \(i_m \in \mathcal{I}_{i_1\ldots i_M}(S)\) for all \(m \in \mathcal{M}_{i^*}\) that \(i^* \in \mathcal{I}_{i_1\ldots i_M}(S)\).

We now prove the desired result for Case 2. First, we observe that

\[
\rho_{i_1\ldots i_M}(S) = \min_{j \in \mathcal{S}_{\mathcal{I}_{i_1\ldots i_M}(S)}} r_j \leq r_{i^*},
\]  

(EC.3)

where the equality follows from Proposition 2 and the inequality follows from our supposition that \(i^* \in S\) and because we have shown that \(i^* \in \mathcal{I}_{i_1\ldots i_M}(S)\). Therefore, we have that

\[
\rho_{i_1\ldots i_M}(S \cup \{i\}) = \min_{j \in \mathcal{S}_{\mathcal{I}_{i_1\ldots i_M}(S) \cup \{i\}}} \min_{j \in \mathcal{S}_{\mathcal{I}_{i_1\ldots i_M}(S) \cup \{i\}}} \min_{j \in \mathcal{S}_{\mathcal{I}_{i_1\ldots i_M}(S) \cup \{i\}}} r_j, r_{i^*} = \rho_{i_1\ldots i_M}(S),
\]

where the first equality follows from Lemma 6 and the supposition of Case 2 that \(i \in \mathcal{I}_{i_1\ldots i_M}(S)\), the inequality follows algebra, the second equality follows from Proposition 2, and the final equality holds because of our supposition that \(r_{i^*} < r_i\) and because of line (EC.3), which showed that \(\rho_{i_1\ldots i_M}(S) \leq r_{i^*}\). This concludes the proof of Case 2.

In both of the above two cases, we showed that \(\rho_{i_1\ldots i_M}(S) \leq \rho_{i_1\ldots i_M}(S \cup \{i\})\), and so our proof of Lemma 7 is complete. \(\square\)

C.3. Proof of Proposition 3

Consider any arbitrary assortment \(S \in \mathcal{S}\). For this assortment, we define a new assortment as

\[
S' \triangleq S \cup \{i \in \mathcal{N}_0 : \text{there exists } i^* \in S \text{ such that } \mathcal{M}_{i^*} \subseteq \mathcal{M}_i \text{ and } r_{i^*} < r_i\}.
\]

It follows immediately from the definition of the collection \(\hat{S}\) that this new assortment satisfies \(S' \in \hat{S}\). Moreover, let \(\{j_1, \ldots, j_\nu\} \triangleq S' \setminus S\) denote the new products that have been added into the assortment. Then we observe for each tuple of products \((i_1, \ldots, i_M) \in \mathcal{L}\) that

\[
\rho_{i_1\ldots i_M}(S') = \rho_{i_1\ldots i_M}(S) + \sum_{i=1}^{\nu} (\rho_{i_1\ldots i_M}(S \cup \{j_1, \ldots, j_i\}) - \rho_{i_1\ldots i_M}(S \cup \{j_1, \ldots, j_{i-1}\})) \geq \rho_{i_1\ldots i_M}(S).
\]

Indeed, the equality follows from algebra. The inequality follows from Lemma 7, which implies that \(\rho_{i_1\ldots i_M}(S \cup \{j_1, \ldots, j_i\}) \geq \rho_{i_1\ldots i_M}(S \cup \{j_1, \ldots, j_{i-1}\})\) for each \(i \in \{1, \ldots, \nu\}\). Since the assortment \(S \in \mathcal{S}\) was chosen arbitrarily, our proof of Proposition 3 is complete. \(\square\)
Appendix D: Proofs of Intermediary Technical Results from §4.2

D.1. Proof of Lemma 8

For notational convenience, let the collection on the right side of line (1) in Lemma 8 be denoted by $\hat{S}$. It follows immediately from the definition of $\hat{S}$ and from the fact that $r_0 < r_1 < \cdots < r_n$ that each assortment $S \in \hat{S}$ is also an element of $\hat{S}$. This proves that $\hat{S} \supseteq \hat{S}$.

To show the other direction, consider any arbitrary assortment $S \in \hat{S}$. We first show that $S_1 \cap S_2 \subseteq S$. Indeed, it follows from the fact that $S, S_1, S_2 \in S$ that $0 \in S \cap S_1 \cap S_2$. Moreover, for each product $j \in (S_1 \cap S_2) \setminus \{0\}$, we readily observe that the inequality $r_j > r_0$ and the equality $M_j = M_0$ both hold. Therefore, it follows from the definition of $\hat{S}$ and the fact that $S \in \hat{S}$ that each product $j \in S \cap S_1$ is also an element of $S$. We have thus shown that $S_1 \cap S_2 \subseteq S$ for all $S \in \hat{S}$. Next, we define the following integers:

$$i_1 \triangleq \min \left\{ n, \min_{j \in S_1 \cap S_2} j \right\}, \quad i_2 \triangleq \min \left\{ n, \min_{j \in S_2 \setminus S_1} j \right\},$$

where any minimization problem over an empty feasible set is defined equal to $\infty$. It follows from the fact that $S \in \hat{S}$, from the definition of $\hat{S}$, and from the assumption of $r_0 < r_1 < \cdots < r_n$ that

$$\{j \in S_1 \setminus S_2 : j \geq i_1\} \subseteq S \quad \text{and} \quad \{j \in S_2 \setminus S_1 : j \geq i_2\} \subseteq S.$$

Hence, it follows from the assumption that $S_1 \cup S_2 = N_0$ that

$$S = (S_1 \cap S_2) \cup \{j \in S_1 \setminus S_2 : j \geq i_1\} \cup \{j \in S_2 \setminus S_1 : j \geq i_2\}.$$

We have thus shown that the assortment $S$ is an element of the collection of assortments $\hat{S}$. Since $S \in \hat{S}$ was chosen arbitrarily, we have shown that $\hat{S} \subseteq \hat{S}$, which concludes our proof of Lemma 8. $\square$

D.2. Proof of Lemma 9

It follows from Definition 2 and the fact that $M = 2$ that

$$L = \{(i_1, i_2) : D_{i_1}(S_1) \cap D_{i_2}(S_2) \neq \emptyset\}.$$

The rest of the proof of Lemma 9 is split into three cases.

In the first case, consider any arbitrary pair of products $(i_1, i_2) \in (S_1 \setminus S_2) \times S_2$. For this pair of products, consider any ranking $\sigma \in \Sigma$ which satisfies the equalities $\sigma(i_1) = 0$ and $\sigma(i_2) = 1$. It follows from the fact that $i_1 \in S_1$ that this ranking satisfies the equality $\arg \min_{j \in S_1} \sigma(j) = 1$. Moreover, it follows from the facts that $i_1 \notin S_2$ and $i_2 \in S_2$ that this ranking also satisfies the equality $\arg \min_{j \in S_2} \sigma(j) = 2$. We have thus shown that $\sigma \in D_1(S_1) \cap D_2(S_2)$, which proves that $D_1(S_1) \cap D_2(S_2) \neq \emptyset$. Since the pair of products $(i_1, i_2) \in (S_1 \setminus S_2) \times S_2$ was chosen arbitrarily, we have shown that $(S_1 \setminus S_2) \times S_2 \subseteq L$.

In the second case, consider any arbitrary pair of products $(i_1, i_2) \in S_1 \times (S_2 \setminus S_1)$. Using identical reasoning as the first case, we observe that $D_1(S_1) \cap D_2(S_2) \neq \emptyset$, which shows that $S_1 \times (S_2 \setminus S_1) \subseteq L$.

In the third case, consider any arbitrary pair of products $(i_1, i_2) \in (S_1 \cap S_2) \times (S_1 \cap S_2)$. For this pair of products, it follows from Definition 1 that the inequality $\sigma(i_1) < \sigma(j)$ must hold for all rankings $\sigma \in D_{i_1}(S_1)$
and all products \( j \in (S_1 \cap S_2) \setminus \{i_1\} \). It also follows from Definition 1 that the inequality \( \sigma(i_2) < \sigma(j) \) must hold for all rankings \( \sigma \in D_{i_1}(S_2) \) and all products \( j \in (S_1 \cap S_2) \setminus \{i_2\} \). It follows immediately from these inequalities that there exists a ranking that satisfies \( \sigma \in D_{i_1}(S_1) \cap D_{i_2}(S_2) \) if and only if \( i_1 = i_2 \), and so it follows from Definition 2 that \((i_1, i_2) \in \mathcal{L}\) if and only if \( i_1 = i_2 \).

Since the three cases we have considered are exhaustive, our proof of Lemma 9 is complete. \( \square \)

D.3. Proof of Lemma 10

Consider any arbitrary assortment \( S \in \mathcal{S} \). It follows from Proposition 1 and our assumptions of \( M = 2 \) and \( \eta = 0 \) that

\[
\min_{\lambda \in \mathcal{U}} \mathcal{R}^\lambda(S) = \left[ \begin{array}{l}
\text{minimize} \sum_{(i_1, i_2) \in \mathcal{L}} \rho_{i_1 i_2}(S) \lambda_{i_1 i_2} \\
\text{subject to} \sum_{(i_1, i_2) \in \mathcal{L} : i_1 = i} \lambda_{i_1 i_2} = v_{1,i} \quad \forall i \in S_1 \\
\sum_{(i_1, i_2) \in \mathcal{L} : i_2 = i} \lambda_{i_1 i_2} = v_{2,i} \quad \forall i \in S_2 \\
\sum_{(i_1, i_2) \in \mathcal{L}} \lambda_{i_1 i_2} = 1 \\
\lambda_{11} \geq 0 \quad \forall (i_1, i_2) \in \mathcal{L}
\end{array} \right]
\]

After applying Lemma 9 to the above optimization problem and removing the redundant constraint \( \sum_{(i_1, i_2) \in \mathcal{L}} \lambda_{i_1 i_2} = 1 \), we observe that

\[
\min_{\lambda \in \mathcal{U}} \mathcal{R}^\lambda(S) = \left[ \begin{array}{l}
\text{minimize} \sum_{i_1 \in S_1 \setminus S_2} \sum_{i_2 \in S_1 \setminus S_2} \rho_{i_1 i_2}(S) \lambda_{i_1 i_2} + \sum_{i_1 \in S_1 \setminus S_2} \sum_{i_2 \in S_2 \setminus S_1} \rho_{i_1 i_2}(S) \lambda_{i_1 i_2} \\
\sum_{i_1 \in S_1 \cap S_2} \sum_{i_2 \in S_2 \setminus S_1} \rho_{i_1 i_2}(S) \lambda_{i_1 i_2} + \sum_{i \in S_1 \setminus S_2} \rho_{ii}(S) \lambda_{ii} \\
\lambda_{ii} + \sum_{i_2 \in S_2 \setminus S_1} \lambda_{i_1 i_2} = v_{1,i} \quad \forall i \in S_1 \setminus S_2 \\
\lambda_{ii} + \sum_{i_1 \in S_1 \setminus S_2} \lambda_{i_1 i_2} = v_{2,i} \quad \forall i \in S_1 \setminus S_2 \\
\lambda_{i_1 i_2} \geq 0 \quad \forall (i_1, i_2) \in \mathcal{L}
\end{array} \right]
\]

Finally, for each \( i \in S_1 \cap S_2 \), we eliminate the decision variable \( \lambda_{ii} \) from the above optimization problem by using the equality \( \lambda_{ii} = v_{1,i} - \sum_{i_2 \in S_2 \setminus S_1} \lambda_{i_1 i_2} \). With this elimination, we obtain the desired result. \( \square \)

Appendix E: In-Depth Analysis of a Specific Problem Instance

In this section, we conduct an in-depth analysis of a specific problem instance with two previously-offered assortments. Our goal here is to show that there exist problem instances which can have affirmative answers to the identification question, the unique assortments which answer the identification question for the problem instances are not revenue-ordered assortments, and the estimate-then-optimize technique can yield new assortments for the problem instances with expected revenues that are strictly less than the best expected revenues obtained from the firm’s previously-offered assortments.
E.1. Description of Problem Instance

We begin by defining the specific problem instance that will be our focus throughout this section. The problem instance is comprised of two previously-offered assortments, which are denoted by $S_1 = \{0, 2, 3, 4\}$ and $S_2 = \{0, 1, 2, 4\}$. The revenues corresponding to the four products are $r_1 = $10, $r_2 = $20, $r_3 = $30, and $r_4 = $100. The historical sales data for the two previously-offered assortments are the purchase frequencies $(v_{1,0}, v_{1,2}, v_{1,3}, v_{1,4}) = (0.3, 0.3, 0.3, 0.1)$ and $(v_{2,0}, v_{2,1}, v_{2,2}, v_{2,4}) = (0.3, 0.3, 0.1, 0.3)$. In the remainder of this section, we will investigate the identification question for this problem instance in the case of $\eta = 0$.

E.2. Useful Facts for Problem Instance

To facilitate our analysis throughout this section, we next develop some useful facts about the problem instance defined in §E.1. Indeed, we first observe that the expected revenues associated with the two previously-offered assortments $S_1 = \{0, 2, 3, 4\}$ and $S_2 = \{0, 1, 2, 4\}$ are

$$r^Tv_1 = 10 \times 0.3 + 20 \times 0.3 + 30 \times 0.3 + 100 \times 0.1 = $25,$
$$r^Tv_2 = 10 \times 0.3 + 20 \times 0.3 + 20 \times 0.1 + 100 \times 0.3 = $35.$

Hence, the best expected revenue from a previously-offered assortment is $\max\{25, 35\} = $35.

We next show that the set of ranking-based choice models $U$ is nonempty when the radius is $\eta = 0$. To show this, we will construct a ranking-based choice model $\lambda$ that has perfect accuracy on the historical sales data. Specifically, consider the following five rankings:

$$\begin{align*}
\sigma^1(0) &= 0, & \sigma^2(1) &= 0, & \sigma^3(1) &= 0, & \sigma^4(2) &= 0, & \sigma^5(3) &= 0, \\
\sigma^1(1) &= 1, & \sigma^2(2) &= 1, & \sigma^3(4) &= 1, & \sigma^4(4) &= 1, & \sigma^5(4) &= 1, \\
\sigma^1(2) &= 2, & \sigma^2(4) &= 2, & \sigma^3(0) &= 2, & \sigma^4(0) &= 2, & \sigma^5(0) &= 2, \\
\sigma^1(3) &= 3, & \sigma^2(0) &= 3, & \sigma^3(4) &= 3, & \sigma^4(4) &= 3, & \sigma^5(4) &= 3, \\
\sigma^1(4) &= 4, & \sigma^2(4) &= 4, & \sigma^3(4) &= 4, & \sigma^4(4) &= 4, & \sigma^5(4) &= 4.
\end{align*}$$

Indeed, we recall from §2 that a ranking $\sigma \in \Sigma$ prefers product $i \in N_0$ over product $j \in N_0$ if $\sigma(i) < \sigma(j)$. For the sake of simplicity, we have chosen without loss of generality to not specify the preference ordering among products which are less preferred than the no-purchase option in each ranking. We observe that the most preferred products from the previously-offered assortments $S_1 = \{0, 2, 3, 4\}$ and $S_2 = \{0, 1, 2, 4\}$ under each of the five rankings are

$$\begin{align*}
\arg\min_{j \in S_1} \sigma^1(j) &= 0, & \arg\min_{j \in S_1} \sigma^2(j) &= 2, & \arg\min_{j \in S_1} \sigma^3(j) &= 4, & \arg\min_{j \in S_1} \sigma^4(j) &= 2, & \arg\min_{j \in S_1} \sigma^5(j) &= 3, \\
\arg\min_{j \in S_2} \sigma^1(j) &= 0, & \arg\min_{j \in S_2} \sigma^2(j) &= 1, & \arg\min_{j \in S_2} \sigma^3(j) &= 1, & \arg\min_{j \in S_2} \sigma^4(j) &= 2, & \arg\min_{j \in S_2} \sigma^5(j) &= 4.
\end{align*}$$

It follows from the above observations that the historical sales data $(v_{1,0}, v_{1,2}, v_{1,3}, v_{1,4}) = (0.3, 0.3, 0.3, 0.1)$ and $(v_{2,0}, v_{2,1}, v_{2,2}, v_{2,4}) = (0.3, 0.3, 0.1, 0.3)$ are fit perfectly by the ranking-based choice model $\hat{\lambda} \in \Delta_\Sigma$ which satisfies the following equalities:

$$\begin{align*}
\hat{\lambda}_{s,1} &= 0.3, & \hat{\lambda}_{s,2} &= 0.2, & \hat{\lambda}_{s,3} &= 0.1, & \hat{\lambda}_{s,4} &= 0.1, & \hat{\lambda}_{s,5} &= 0.3.
\end{align*}$$
## E.3. Analysis of Estimate-Then-Optimize for Problem Instance

Equipped with the useful facts from §E.2, we now consider the performance of the estimate-then-optimize technique for the problem instance defined in §E.1. Specifically, suppose we use the estimate-then-optimize technique in which our estimate of the ranking-based choice model is defined by the probability distribution $\hat{\lambda}$ defined in §E.2. In that case, the estimate-then-optimize technique will output a new assortment $S^{\hat{\lambda}}$ which is an optimal solution to the following combinatorial optimization problem:

$$\max_{S \in \mathcal{S}} \mathcal{R}^{\hat{\lambda}}(S).$$

To determine the optimal solution to the above combinatorial optimization problem, let us inspect the five rankings $\sigma^1, \ldots, \sigma^5$ which have nonzero probability under the estimated probability distribution. It is easy to see that the first of these five rankings satisfies the equality $\arg\min_{j \in \mathcal{S}} \sigma^1(j) = 0$ for all assortments $S \in \mathcal{S}$. Moreover, we observe that the assortment $\{0, 4\} \in \mathcal{S}$ satisfies the equalities $\arg\min_{j \in \{0, 4\}} \sigma^2(j) = \cdots = \arg\min_{j \in \{0, 4\}} \sigma^5(j) = 4$. From this reasoning, we conclude that $\{0, 4\}$ is an optimal assortment to the above combinatorial optimization problem, and the predicted expected revenue for this assortment is $\mathcal{R}^{\hat{\lambda}}(S^{\hat{\lambda}}) = \mathcal{R}^{\hat{\lambda}}(\{0, 4\}) = 0.7 \times \$100 = \$70$.

We will now show that the assortment $S^{\hat{\lambda}} = \{0, 4\}$ obtained from estimate-then-optimize can have an expected revenue that is strictly lower than the best expected revenue obtained from a previously-offered assortment under a ranking-based choice model that is consistent with the historical sales data. To show this, we consider the following seven rankings:

$$\sigma^6(0) = 0, \ \ \sigma^7(1) = 0, \ \ \sigma^8(2) = 0, \ \ \sigma^9(3) = 0, \ \ \sigma^{10}(4) = 0, \ \ \sigma^{11}(1) = 0, \ \ \sigma^{12}(3) = 0,$$

$$\sigma^6( ) = 1, \ \ \sigma^7(0) = 1, \ \ \sigma^8(0) = 1, \ \ \sigma^9(0) = 1, \ \ \sigma^{10}(0) = 1, \ \ \sigma^{11}(2) = 1, \ \ \sigma^{12}(4) = 1,$$

$$\sigma^6( ) = 2, \ \ \sigma^7( ) = 2, \ \ \sigma^8( ) = 2, \ \ \sigma^9( ) = 2, \ \ \sigma^{10}( ) = 2, \ \ \sigma^{11}(0) = 2, \ \ \sigma^{12}(0) = 2,$$

$$\sigma^6( ) = 3, \ \ \sigma^7( ) = 3, \ \ \sigma^8( ) = 3, \ \ \sigma^9( ) = 3, \ \ \sigma^{10}( ) = 3, \ \ \sigma^{11}( ) = 3, \ \ \sigma^{12}( ) = 3,$$

$$\sigma^6( ) = 4, \ \ \sigma^7( ) = 4, \ \ \sigma^8( ) = 4, \ \ \sigma^9( ) = 4, \ \ \sigma^{10}( ) = 4, \ \ \sigma^{11}( ) = 4, \ \ \sigma^{12}( ) = 4.$$

We observe that the most preferred products from the first previously-offered assortment $S_1 = \{0, 2, 3, 4\}$ are

$$\arg\min_{j \in S_1} \sigma^6(j) = 0, \ \ \arg\min_{j \in S_1} \sigma^7(j) = 0, \ \ \arg\min_{j \in S_1} \sigma^8(j) = 2, \ \ \arg\min_{j \in S_1} \sigma^9(j) = 3, \ \ \arg\min_{j \in S_1} \sigma^{10}(j) = 4, \ \ \arg\min_{j \in S_1} \sigma^{11}(j) = 2, \ \ \arg\min_{j \in S_1} \sigma^{12}(j) = 3,$$

the most preferred products from the second previously-offered assortment $S_2 = \{0, 1, 2, 4\}$ are

$$\arg\min_{j \in S_2} \sigma^6(j) = 0, \ \ \arg\min_{j \in S_2} \sigma^7(j) = 1, \ \ \arg\min_{j \in S_2} \sigma^8(j) = 2, \ \ \arg\min_{j \in S_2} \sigma^9(j) = 0, \ \ \arg\min_{j \in S_2} \sigma^{10}(j) = 4, \ \ \arg\min_{j \in S_2} \sigma^{11}(j) = 1, \ \ \arg\min_{j \in S_2} \sigma^{12}(j) = 4,$$

and the most preferred products from the new assortment $S^{\hat{\lambda}} = \{0, 4\}$ from estimate-then-optimize are

$$\arg\min_{j \in S^{\hat{\lambda}}} \sigma^6(j) = 0, \ \ \arg\min_{j \in S^{\hat{\lambda}}} \sigma^7(j) = 0, \ \ \arg\min_{j \in S^{\hat{\lambda}}} \sigma^8(j) = 0, \ \ \arg\min_{j \in S^{\hat{\lambda}}} \sigma^9(j) = 0, \ \ \arg\min_{j \in S^{\hat{\lambda}}} \sigma^{10}(j) = 4, \ \ \arg\min_{j \in S^{\hat{\lambda}}} \sigma^{11}(j) = 0, \ \ \arg\min_{j \in S^{\hat{\lambda}}} \sigma^{12}(j) = 4.$$
It follows from the above observations that the historical sales data \((v_{1.0}, v_{1.2}, v_{1.3}, v_{1.4}) = (0.3, 0.3, 0.3, 0.1)\) and \((v_{2.0}, v_{2.1}, v_{2.2}, v_{2.4}) = (0.3, 0.3, 0.1, 0.3)\) are fit perfectly by the ranking-based choice model \(\hat{\lambda} \in \Delta_S\) which satisfies the following equalities:

\[
\tilde{\lambda}_{a^6} = 0.2, \quad \tilde{\lambda}_{a^7} = 0.1, \quad \tilde{\lambda}_{a^8} = 0.1, \quad \tilde{\lambda}_{a^9} = 0.1, \quad \tilde{\lambda}_{a^{10}} = 0.1, \quad \tilde{\lambda}_{a^{11}} = 0.2, \quad \tilde{\lambda}_{a^{12}} = 0.2.
\]

However, we observe that the expected revenue for the assortment \(S^\lambda\) under the ranking-based choice model with parameters \(\hat{\lambda}\) is \(\mathcal{R}^\lambda(S^\lambda) = \mathcal{R}^\lambda(\{0,4\}) = 0.3 \times $100 = $30\). Since this is strictly less than the best expected revenue from a previously-offered assortment, \$35, we have thus shown that the estimate-then-optimize technique can yield an assortment for the problem instance defined in §E.1 with an expected revenue that is strictly worse than the best expected revenue obtained from a previously-offered assortment under a ranking-based choice model that is consistent with the historical sales data.

E.4. Affirmative Answer to Identification Question for Problem Instance

We next show that there are affirmative answers to the identification question for the problem instance defined in §E.1. From this point forward, our analysis will closely follow the algorithm for answering the identification question which was developed in §4.2.1. We begin by constructing the collection of assortments \(\bar{S}\) corresponding to the previously-offered assortments \(S_1 = \{0, 2, 3, 4\}\) and \(S_2 = \{0, 1, 2, 4\}\). Indeed, we observe that \(S_1 \cap S_2 = \{0, 2, 4\}, \ S_1 \setminus S_2 = \{3\}, \) and \(S_2 \setminus S_1 = \{1\}\), and so it follows from Lemma 8 that

\[
\bar{S} = \left\{ S \in S : \begin{array}{l} \text{there exists } i_1 \in \{3, 4\} \text{ and } i_2 \in \{1, 4\} \text{ such that } \\
S = \{0, 2, 4\} \cup \{j \in \{3\} : j \geq i_1\} \cup \{j \in \{1\} : j \geq i_2\} \end{array} \right\} = \left\{ \begin{array}{l} \{0, 1, 2, 3, 4\}, \\
\{0, 1, 2, 4\}, \\
\{0, 2, 3, 4\}, \\
\{0, 2, 4\} \end{array} \right\}.
\]

Under the assumption that \(\eta = 0\) from §E.1, it follows from our implementation of the algorithm from §4.2.1 that

\[
\min_{\lambda \in U} \mathcal{R}^\lambda(\{0,1,2,3,4\}) = \$14, \\
\min_{\lambda \in U} \mathcal{R}^\lambda(\{0,1,2,4\}) = \$35, \\
\min_{\lambda \in U} \mathcal{R}^\lambda(\{0,2,3,4\}) = \$25, \\
\min_{\lambda \in U} \mathcal{R}^\lambda(\{0,2,4\}) = \$36.
\]

Hence, we observe that there is an affirmative answer to the identification question associated with the assortment \(\{0, 2, 4\}\), since this assortment satisfies \(\mathcal{R}^\lambda(\{0,2,4\}) \geq \$36 > \$35\) for all ranking-based choice models \(\lambda \in U\). In particular, we note that \(\{0, 2, 4\}\) is not a revenue-ordered assortment. We remark that the correctness of the above equalities \(\min_{\lambda \in U} \mathcal{R}^\lambda(\{0,1,2,4\}) = \$35\) and \(\min_{\lambda \in U} \mathcal{R}^\lambda(\{0,2,3,4\}) = \$25\) for the two previously-offered assortments follows immediately from the fact that \(\eta = 0\), which implies that \(\mathcal{R}^\lambda(S_m) = v_m\) for all \(m \in \{1, 2\}\) and all \(\lambda \in U\).

We conclude this section by showing that \(\{0, 2, 4\}\) is the unique assortment that answers the identification question for our specific problem instance. To show this, we present below an exhaustive list of the worst-case
expected revenues \( \min_{\lambda \in U} \mathcal{R}^\lambda (S) \) for all assortments \( S \in S \) which satisfy \( 4 \in S \). We compute the following list by using our implementation of the algorithm from §4.2.1. The following list shows that \( \{0, 2, 4\} \) is the unique assortment which leads to an affirmative answer to the identification question for our specific problem instance.

\[
\begin{align*}
\min_{\lambda \in U} \mathcal{R}^\lambda (\{0, 4\}) &= $30, \\
\min_{\lambda \in U} \mathcal{R}^\lambda (\{0, 1, 4\}) &= $33, \\
\min_{\lambda \in U} \mathcal{R}^\lambda (\{0, 2, 4\}) &= $36, \\
\min_{\lambda \in U} \mathcal{R}^\lambda (\{0, 3, 4\}) &= $19, \\
\min_{\lambda \in U} \mathcal{R}^\lambda (\{0, 1, 2, 4\}) &= $35, \\
\min_{\lambda \in U} \mathcal{R}^\lambda (\{0, 1, 3, 4\}) &= $12, \\
\min_{\lambda \in U} \mathcal{R}^\lambda (\{0, 2, 3, 4\}) &= $25, \\
\min_{\lambda \in U} \mathcal{R}^\lambda (\{0, 1, 2, 3, 4\}) &= $14.
\end{align*}
\]

Appendix F: Proofs of Intermediary Technical Results from §4.3

F.1. Proof of Lemma 11

As the first step in our proof of Lemma 11, we develop an algorithm for constructing a directed graph, denoted by \( G \equiv (\mathcal{V}, \mathcal{E}) \), in which the set of vertices in the directed graph is defined as \( \mathcal{V} \equiv N \) and the set of directed edges is defined as \( \mathcal{E} \equiv \{(i^*, i) \in N \times N : r_{i^*} < r_i \text{ and } M_{i^*} \subseteq M_i\} \). This directed graph has a natural correspondence with the collection of assortments \( \hat{S} \), as we readily observe that an assortment satisfies \( S \in \hat{S} \) if and only if \([0 \in S]\) and \([\text{we have that } i \in S \text{ whenever there exists a product } i^* \in S \text{ and a directed edge } (i^*, i) \in \mathcal{E}\]\. It is easy to see that this directed graph is acyclic and has the property that a vertex \( j \in \mathcal{V} \) is reachable from a vertex \( i \in \mathcal{V} \) if and only if there is a directed edge \((i, j) \in \mathcal{E}\) from vertex \( i \) to vertex \( j \). A directed acyclic graph which has the aforementioned property for each pair of vertices is referred to as a transitive closure (Aluja et al. 1988, p.90).

Our algorithm for constructing the directed acyclic graph \( G \) that is a transitive closure is presented in Algorithm 1. In the algorithm, we first iterate over each product \( i \in N \) and construct the corresponding set \( M_i \) of previously-offered assortments which offered that product. It is easy to see that each of the sets \( M_i \) can be constructed in \( O(M) \) computation time by checking whether the product satisfies \( i \in S_m \) for each previously-offered assortment \( m \in M_i \); hence, we observe that all of the sets \( M_0, \ldots, M_n \) can be constructed in a total of \( O(Mn) \) computation time. Assume that we store the sets \( M_0, \ldots, M_n \) as unsorted arrays as well as hash tables. Given these data structures for \( M_0, \ldots, M_n \), we then iterate over each pair of products \((i, i^*) \in N \times N \) and check in \( O(M) \) computation time whether \( r_{i^*} < r_i \) and \( M_{i^*} \subseteq M_i \). Since there are \( O(n^2) \) pairs of products in \( N \times N \), we conclude that the set of directed edges \( \mathcal{E} \) can be constructed in a total of \( O(n^2 M) \) computation time. Combining all of the steps, and since it takes \( O(n) \) computation time to construct the set of vertices \( \mathcal{V} \), we have shown that the total computation time for Algorithm 1 is \( O(n + Mn + n^2 M) = O(n^2 M) \).
Construct-$\mathcal{G}(\mathcal{M}, r)$

Inputs:
- The collection of previously-offered assortments, $\mathcal{M} \equiv \{S_1, \ldots, S_M\}$.
- The revenues of the products, $r \equiv (r_0, r_1, \ldots, r_n)$.

Output:
- $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} \equiv N_0$ and $\mathcal{E} \equiv \{(i^*, i) \in N_0 \times N_0 : r_{i^*} < r_i$ and $M_{i^*} \subseteq M_i\}$.

Procedure:
1. Initialize the vertex set $\mathcal{V} \leftarrow \emptyset$ and edge set $\mathcal{E} \leftarrow \emptyset$.
2. For each product $i \in N_0$:
   (a) Update $\mathcal{V} \leftarrow \mathcal{V} \cup \{i\}$.
   (b) Construct the set $M_i$ of previously-offered assortments which offered product $i$.
3. For each pair of products $(i^*, i) \in N_0 \times N_0$:
   (a) If $r_{i^*} < r_i$ and $M_{i^*} \subseteq M_i$:
      i. Update $\mathcal{E} \leftarrow \mathcal{E} \cup \{(i^*, i)\}$.
4. Output $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$ and terminate.

Algorithm 1: A procedure for constructing the directed acyclic graph $\mathcal{G}$.

We next describe our algorithm for constructing the collection of assortments $\hat{S}$ from the directed graph $\mathcal{G}$. This algorithm is denoted by $\text{Construct-}\hat{S}(\mathcal{M}, r)$ and is found in Algorithm 2. In this algorithm, we first use Algorithm 1 to construct the directed acyclic graph $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$ which is a transitive closure, and then we invoke a recursive subroutine denoted by $\text{RecursiveStep}(\mathcal{G})$ in Algorithm 3. The goal of the recursive subroutine is to take as an input a generic directed acyclic graph $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$ which is a transitive closure, and for that graph, output the collection of subsets of vertices $\mathcal{A} \equiv \{S \subseteq \mathcal{V} : \text{if } i \in S \text{ and } (i, j) \in \mathcal{E}, \text{ then } j \in S\}$. Algorithm 2 concludes by removing the subsets of vertices from $\mathcal{A}$ which do not include the no-purchase option 0, and then outputs the remaining subsets of vertices from $\mathcal{A}$. The correctness of Algorithm 2 follows immediately from our earlier observation that an assortment satisfies $S \in \hat{S}$ if and only if $[0 \in S]$ and $[\text{we have that } i \in S \text{ whenever there exists a product } i^* \in S \text{ and a directed edge } (i^*, i) \in \mathcal{E}]$.

At a high level, the recursive subroutine in Algorithm 3 is comprised of two cases. The base case of the subroutine is when the graph has no vertices, in which case it is clear that $\mathcal{A} = \{\emptyset\}$. If we are not in the base case, then the aim of the recursive subroutine is to construct the collections $\{S \in \mathcal{A} : i \notin S\}$ and $\{S \in \mathcal{A} : i \in S\}$ for a chosen vertex $i \in \mathcal{V}$ and output the union of these two collections. The construction of the collection $\{S \in \mathcal{A} : i \notin S\}$ takes place on lines (2b)-(2c) of Algorithm 3, and the construction of the collection $\{S \in \mathcal{A} : i \in S\}$ takes place on lines (2d)-(2f) of Algorithm 3.

Up to this point, we have established that Algorithm 1 is correct (that is, it delivers the desired output for any valid input), and we have established that Algorithm 2 is correct under the assumption that Algorithm 3 is correct. Therefore, it remains for us to prove that the recursive subroutine in Algorithm 3 is correct. To prove the correctness of the recursive subroutine, we will make use of four intermediary claims, which are denoted below by Claims EC.1-EC.4. The purpose of the first two intermediary claims, denoted by Claims EC.1 and EC.2, is to show that the graphs $\mathcal{G}' \equiv (\mathcal{V}', \mathcal{E}')$ and $\mathcal{G}'' \equiv (\mathcal{V}'', \mathcal{E}'')$ constructed on lines (2b)
**Construct-$\hat{S}(\mathcal{M}, r)$**

**Inputs:**
- The collection of previously-offered assortments, $\mathcal{M} \equiv \{S_1, \ldots, S_M\}$.
- The revenues of the products, $r \equiv (r_0, r_1, \ldots, r_n)$.

**Output:**
- The collection of assortments $\hat{S}$ corresponding to the collection of previously-offered assortments $\mathcal{M}$ and the revenues $r$.

**Procedure:**
1. Construct the directed acyclic graph $\mathcal{G} \leftarrow$ **Construct-$\mathcal{G}(\mathcal{M}, r)$**.
2. Compute the collection of assortments $\hat{S} \leftarrow$ **RecursiveStep($\mathcal{G}$)**.
3. For each $S \in \hat{S}$:
   (a) If $0 \not\in S$:
      i. Update $\hat{S} \leftarrow \hat{S} \setminus \{S\}$.
4. Output $\hat{S}$ and terminate.

Algorithm 2: A procedure for constructing $\hat{S}$.

**RecursiveStep($\mathcal{G}$)**

**Inputs:**
- A directed acyclic graph $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$ which is a transitive closure.

**Output:**
- The collection $\mathcal{A} \equiv \{S \subseteq \mathcal{V}: \text{if } i \in S \text{ and } (i, j) \in \mathcal{E}, \text{then } j \in S\}$.

**Procedure:**
1. If $\mathcal{V} = \emptyset$:
   (a) Output the collection $\mathcal{A} \equiv \{\emptyset\}$ and terminate.
2. Otherwise:
   (a) Choose any vertex $i \in \mathcal{V}$.
   (b) Create a copy of $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$ in which the vertices $\{i\} \cup \{\ell: (\ell, i) \in \mathcal{E}\}$ and the incoming and outgoing edges of these vertices are removed. Denote this new graph by $\mathcal{G}' \equiv (\mathcal{V}', \mathcal{E}')$.
   (c) Compute the collection $\mathcal{A}' \leftarrow$ **RecursiveStep($\mathcal{G}'$)**.
   (d) Create a copy of $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$ in which the vertices $\{i\} \cup \{\ell: (i, \ell) \in \mathcal{E}\}$ and the incoming and outgoing edges of these vertices are removed. Denote this new graph by $\mathcal{G}'' \equiv (\mathcal{V}'', \mathcal{E}'')$.
   (e) Compute the collection $\mathcal{A}'' \leftarrow$ **RecursiveStep($\mathcal{G}''$)**.
   (f) Compute the collection $\mathcal{A}'''' \leftarrow \{S \cup \{i\} \cup \{\ell: (i, \ell) \in \mathcal{E}\}: S \in \mathcal{A}''\}$.
   (g) Output the collection $\mathcal{A} \equiv \mathcal{A}' \cup \mathcal{A}''''$ and terminate.

Algorithm 3: A recursive subroutine which is used in Algorithm 2.
and (2d) of Algorithm 3 are directed acyclic graphs that are transitive closures, which implies that $G'$ and $G''$ are valid inputs on lines (2c) and (2e) of Algorithm 3. The purpose of the second two intermediary claims, denoted by Claims EC.3 and EC.4, is to show that the union of the two collections $A'$ and $A''$ constructed on lines (2c) and (2f) of Algorithm 3 provides the desired output on line (2g) of Algorithm 3.

**Claim EC.1.** Let $G \equiv (V, E)$ be a directed acyclic graph that is a transitive closure. For notational convenience, let $E_i$ denote the set of incoming and outgoing edges from each vertex $i \in V$. Then for each vertex $i \in V$, we have that $\tilde{G} \equiv (V \setminus \{i\}, E \setminus E_i)$ is a directed acyclic graph that is a transitive closure.

**Proof of Claim EC.1.** Let $G \equiv (V, E)$ be a directed acyclic graph that is a transitive closure, and let $i \in V$ be any chosen vertex. It is clear that removing a vertex and its associated incoming and outgoing edges from a directed acyclic graph will not induce any cycles. Therefore, it follows from the fact that the original graph $G \equiv (V, E)$ is a directed acyclic graph that the new graph $\tilde{G} \equiv (V \setminus \{i\}, E \setminus E_i)$ is also a directed acyclic graph. Moreover, consider any two arbitrary vertices $j, k \in V \setminus \{i\}$ which satisfy the property that vertex $k$ is reachable from vertex $j$ in the new graph $\tilde{G} \equiv (V \setminus \{i\}, E \setminus E_i)$. Then it follows immediately from the construction of the new graph that vertex $k$ is reachable from vertex $j$ in the original graph $G \equiv (V, E)$. Since the original graph $G \equiv (V, E)$ is a transitive closure, there must exist a directed edge from vertex $j$ to vertex $k$ in the original graph. Since neither $j$ nor $k$ are equal to $i$, we have thus shown that there is a directed edge $(j, k) \in E \setminus E_i$. Since the two vertices $j, k \in V \setminus \{i\}$ were chosen arbitrarily, we have shown that the new graph is also a transitive closure. Our proof of Claim EC.1 is thus complete. □

**Claim EC.2.** Let $G \equiv (V, E)$ be a directed acyclic graph that is a transitive closure. For notational convenience, let $E_i$ denote the set of incoming and outgoing edges from each vertex $i \in V$. Then for each subset of vertices $B \subseteq V$, we have that $\tilde{G} \equiv (V \setminus B, E \setminus (\cup_{i \in B} E_i))$ is a directed acyclic graph that is a transitive closure.

**Proof of Claim EC.2.** Let $G \equiv (V, E)$ be a directed acyclic graph that is a transitive closure, let $B \subseteq V$ be a subset of vertices, and define the directed graph $\tilde{G} \equiv (V \setminus B, E \setminus (\cup_{i \in B} E_i))$. The rest of the proof follows a straightforward induction argument. Indeed, let the vertices of $B$ be indexed by $B \equiv \{i^B_1, \ldots, i^B_{|B|}\}$. For each $j \in \{0, 1, \ldots, |B|\}$, we define the following directed graph:

$$G^B_j \equiv (V \setminus \{i^B_1, \ldots, i^B_j\}, E \setminus \{(k, \ell) \in E : k \in \{i^B_1, \ldots, i^B_j\} \text{ or } \ell \in \{i^B_1, \ldots, i^B_j\}\}).$$

We readily observe from the above definition that $G^B_0 = G$ and that $G^B_{|B|} = \tilde{G}$. In the remainder, we will prove by induction that each $G^B_0, \ldots, G^B_{|B|}$ is a directed acyclic graph that is a transitive closure. Our induction proof proceeds as follows. In the base case, it follows from the equality $G^B_0 = G$ that $G^B_0$ is a directed acyclic graph that is a transitive closure. Next, assume by induction that $G^B_0, \ldots, G^B_{j-1}$ are directed acyclic graphs that are transitive closures for any $j \in \{1, \ldots, |B|\}$. Then we readily observe from the definition of the directed graph $G^B_j \equiv (V^B_j, E^B_j)$ that the following equalities hold:

$$V^B_j = V^B_{j-1} \setminus \{i^B_j\}; \quad E^B_j = E^B_{j-1} \setminus \{(k, \ell) \in E^B_{j-1} : k = i^B_j \text{ or } \ell = i^B_j\}.$$

Using the above equalities, it follows immediately from Claim EC.1 and the induction hypothesis that $G^B_j$ is a directed acyclic graph that is a transitive closure. This concludes our induction proof, and since the equality $G^B_{|B|} = \tilde{G}$ holds, our proof of Claim EC.2 is complete. □
Claim EC.3. Let $\mathcal{G} \equiv (V, E)$ be a directed acyclic graph that is a transitive closure. For notational convenience, let $\mathcal{E}_i$ denote the set of incoming and outgoing edges from each vertex $i \in V$. For each vertex $i \in V$,

$$
\{S \subseteq V \setminus \{i\} : \text{if } k \in S \text{ and } (k, j) \in \mathcal{E}, \text{ then } j \in S\}
$$

$$
= \{S \subseteq V \setminus \{(i) \cup \{\ell : (\ell, i) \in \mathcal{E}\}} : \text{if } k \in S \text{ and } (k, j) \in \mathcal{E} \setminus \bigg( \mathcal{E}_i \cup \bigcup_{\ell : (\ell, i) \in \mathcal{E}} \mathcal{E}_\ell \bigg) \text{, then } j \in S\}. \tag{EC.4}
$$

Proof of Claim EC.3. Let $\mathcal{G} \equiv (V, E)$ be a directed acyclic graph that is a transitive closure, and let $i \in V$ be any chosen vertex from this graph. We first observe that

$$
\{S \subseteq V \setminus \{i\} : \text{if } k \in S \text{ and } (k, j) \in \mathcal{E}, \text{ then } j \in S\} \tag{EC.5}
$$

$$
= \{S \subseteq V \setminus \{(i) \cup \{\ell : (\ell, i) \in \mathcal{E}\}} : \text{if } k \in S \text{ and } (k, j) \in \mathcal{E}, \text{ then } j \in S\}, \tag{EC.6}
$$

where the above equality follows from the fact that any subset of vertices $S$ from the collection on line (EC.5) must not contain any vertices in the graph that have an outgoing edge to vertex $i$. For notational convenience, let the collection of subsets of vertices on line (EC.4) be denoted by $\hat{A}$, and let the collection of subsets of vertices on line (EC.6) be denoted by $\hat{A}'$. In other words, let

$$
\hat{A} \triangleq \{S \subseteq V \setminus \{i\} \cup \{\ell : (\ell, i) \in \mathcal{E}\}) : \text{if } k \in S \text{ and } (k, j) \in \mathcal{E}, \text{ then } j \in S\}
$$

$$
\hat{A}' \triangleq \{S \subseteq V \setminus \{(i) \cup \{\ell : (\ell, i) \in \mathcal{E}\}} : \text{if } k \in S \text{ and } (k, j) \in \mathcal{E} \setminus \bigg( \mathcal{E}_i \cup \bigcup_{\ell : (\ell, i) \in \mathcal{E}} \mathcal{E}_\ell \bigg), \text{ then } j \in S\}. \tag{EC.7}
$$

In the remainder of the proof, we will show that $\hat{A} = \hat{A}'$.

To show the first direction, consider any arbitrary subset of vertices $S \in \hat{A}'$. For this subset of vertices, consider any two arbitrary vertices $k, j \in V \setminus \{(i) \cup \{\ell : (\ell, i) \in \mathcal{E}\})$ which satisfy the conditions $[k \in S] \text{ and } [(k, j) \in \mathcal{E} \setminus (\mathcal{E}_i \cup \bigcup_{\ell : (\ell, i) \in \mathcal{E}} \mathcal{E}_\ell)]$. In this case, it follows immediately from the fact that $\mathcal{E} \setminus (\mathcal{E}_i \cup \bigcup_{\ell : (\ell, i) \in \mathcal{E}} \mathcal{E}_\ell) \subseteq \mathcal{E}$ that the condition $[(k, j) \in \mathcal{E}]$ holds. Therefore, it follows immediately from the definition of $\hat{A}'$ that $j \in S$. Since the two vertices $k, j \in V \setminus \{(i) \cup \{\ell : (\ell, i) \in \mathcal{E}\})$ which satisfy the conditions $[k \in S] \text{ and } [(k, j) \in \mathcal{E} \setminus (\mathcal{E}_i \cup \bigcup_{\ell : (\ell, i) \in \mathcal{E}} \mathcal{E}_\ell)]$ were chosen arbitrarily, we have shown that $S \in \hat{A}$, and since the subset of vertices $S \in \hat{A}$ was chosen arbitrarily, we have shown that $\hat{A}' \subseteq \hat{A}$. Our proof of the first direction is thus complete.

To show the other direction, consider any arbitrary subset of vertices $S \in \hat{A}$. For this subset of vertices, consider any two arbitrary vertices $k, j \in V \setminus \{(i) \cup \{\ell : (\ell, i) \in \mathcal{E}\})$ which satisfy the conditions $[k \in S] \text{ and } [(k, j) \in \mathcal{E}]$. We now suppose, for the sake of developing a contradiction, that the directed edge from vertex $k$ to vertex $j$ satisfies $(k, j) \in \mathcal{E} \cup \bigcup_{\ell : (\ell, i) \in \mathcal{E}} \mathcal{E}_\ell$. Under this supposition, we have two cases to consider. In the first case, the directed edge satisfies $(k, j) \in \mathcal{E}_i$. However, that would imply that either $j = i$ or $k = i$, and those equalities contradict the facts that $k \in S$ and $S \subseteq V \setminus \{(i) \cup \{\ell : (\ell, i) \in \mathcal{E}\})$. In the second case, the directed edge satisfies $(k, j) \in \bigcup_{\ell : (\ell, i) \in \mathcal{E}} \mathcal{E}_\ell$, which implies that there are directed edges $(k, j) \in \mathcal{E}$ and $(j, i) \in \mathcal{E}$ in the graph $\mathcal{G}$. Since the graph is a transitive closure, the existence of directed edges $(k, j) \in \mathcal{E}$ and $(j, i) \in \mathcal{E}$ implies that there must exist a directed edge $(k, i) \in \mathcal{E}$. However, the existence of a directed edge $(k, i) \in \mathcal{E}$ contradicts the facts that $k \in S$ and $S \subseteq V \setminus \{(i) \cup \{\ell : (\ell, i) \in \mathcal{E}\})$. In all cases, we have proved by contradiction that the vertices $k$ and $j$ must satisfy the condition $[(k, j) \in \mathcal{E} \setminus (\mathcal{E}_i \cup \bigcup_{\ell : (\ell, i) \in \mathcal{E}} \mathcal{E}_\ell)]$. Since the
two vertices $k, j \in V \setminus \{i\}$ which satisfy the conditions $[k \in S]$ and $[(k, j) \in E]$ were chosen arbitrarily, we have shown that $S \in \tilde{A}$; and since the subset of vertices $S \in \tilde{A}$ was chosen arbitrarily, we have shown that $\tilde{A} \subseteq \tilde{A}$. Our proof of the second direction is thus complete.

We have thus shown that $\tilde{A} = \tilde{A}'$, which concludes our proof of Claim EC.3. □

**Claim EC.4.** Let $G \equiv (V, E)$ be a directed acyclic graph that is a transitive closure. For notational convenience, let $E_i$ denote the set of incoming and outgoing edges from each vertex $i \in V$. For each vertex $i \in V$,

$$
\{S \subseteq V : [i \in S] \text{ and } [k \in S \text{ and } (k, j) \in E], \text{ then } j \in S]\}
$$

$$
\{S' \cup \{i\} \cup \{\ell : (i, \ell) \in E]\} : \begin{cases} S' \subseteq V \setminus \{i\} \cup \{\ell : (i, \ell) \in E\} : \\
\text{if } k \in S \text{ and } (k, j) \in E \setminus \bigcup_{\ell : (i, \ell) \in E} E_i \text{, then } j \in S \end{cases}
$$

(EC.7)

**Proof of Claim EC.4.** Let $G \equiv (V, E)$ be a directed acyclic graph that is a transitive closure, and let $i \in V$ be any chosen vertex from this graph. Then we observe that

$$
\{S \subseteq V : [i \in S] \text{ and } [k \in S \text{ and } (k, j) \in E], \text{ then } j \in S]\}
$$

(EC.8)

$$
= \{S \subseteq V : [i] \cup \{\ell : (i, \ell) \in E\} \subseteq S\} \text{ and } [k \in S \text{ and } (k, j) \in E], \text{ then } j \in S]\}
$$

(EC.9)

$$
= \left\{ S \subseteq V : [i] \cup \{\ell : (i, \ell) \in E\} \subseteq S \right\} \text{ and } \begin{cases} \text{if } k \in S \text{ and } (k, j) \in E \setminus \bigcup_{\ell : (i, \ell) \in E} E_i \text{, then } j \in S \end{cases}
$$

(EC.10)

Indeed, the equality of the collections of subsets of vertices on lines (EC.8) and (EC.9) follows from the fact that any subset of vertices $S$ in the collection on line (EC.8) satisfies $i \in S$, and so it must contain all of the vertices in the graph which have an incoming edge from vertex $i$. For notational convenience, let the collections of subsets of vertices on lines (EC.9) and (EC.10) be denoted by $\tilde{A}'$ and $\tilde{A}$, respectively. To show that line (EC.9) is equal to line (EC.10), we first observe that the inclusion $\tilde{A}' \subseteq \tilde{A}$ follows immediately from the definitions of the collections $\tilde{A}'$ and $\tilde{A}$. To show the other direction, consider any subset of vertices $S \in \tilde{A}$ and consider any vertex $k \in S$. If there exists a directed edge $(k, j) \in E$ which satisfies $(k, j) \in E \cup \bigcup_{\ell : (i, \ell) \in E} E_i$, then we observe that the vertex $j$ must satisfy $j \in \{i\} \cup \{\ell : (i, \ell) \in E\}$. Since it follows from $S \in \tilde{A}$ that the inclusion $\{i\} \cup \{\ell : (i, \ell) \in E\} \subseteq S$ holds, we conclude that $S \in \tilde{A}'$. We have thus shown that line (EC.9) is equal to line (EC.10). Since the equivalence of lines (EC.10) and (EC.7) follows readily from algebra, our proof of Claim EC.4 is complete. □

Using the above intermediary claims, we now prove the correctness of the recursive subroutine in Algorithm 3. Indeed, it is clear that the recursive subroutine yields the correct output in the base case where $V = \emptyset$. Next, let us assume by induction that the recursive subroutine yields the correct output for all valid input graphs with up to $p - 1$ vertices, and consider any valid input $G \equiv (V, E)$ for which the number of vertices is $|V| = p$. Let $i \in V$ denote the vertex from this graph which is chosen in line (2a) of Algorithm 3, where the existence of such a vertex follows from the fact that we are in the case where $V \neq \emptyset$. It follows immediately from Claim EC.2 that the graphs $G' \equiv (V', E')$ and $G'' \equiv (V'', E'')$ constructed on lines (2b) and
(2d) of Algorithm 3 are directed acyclic graphs which are transitive closures, which implies that these graphs are valid inputs to Algorithm 3 in lines (2c) and (2e). Therefore, it follows from the induction hypothesis and lines (2c), (2e), and (2f) of Algorithm 3 that

$$A' = \{ S \subseteq V \setminus \{ i \} \cup \{ \ell : (\ell, i) \in E \} : \text{if } k \in S \text{ and } (k, j) \in E \setminus \bigcup_{\ell : (\ell, i) \in E} E_\ell \text{, then } j \in S \}$$

$$A'' = \{ S \subseteq V \setminus \{ i \} \cup \{ \ell : (\ell, i) \in E \} : \text{if } k \in S \text{ and } (k, j) \in E \setminus \bigcup_{\ell : (\ell, i) \in E} E_\ell \text{, then } j \in S \}$$

$$A''' = \{ S \cup \{ i \} \cup \{ \ell : (i, \ell) \in E \} : S \in A'' \} ,$$

where the induction hypothesis can be applied because \( G' \equiv (V', E') \) and \( G'' \equiv (V'', E'') \) are valid inputs to Algorithm 3 and because \( |V'| \leq p - 1 \) and \( |V''| \leq p - 1 \). Therefore, it follows from Claims EC.3 and EC.4 that

$$A' = \{ S \subseteq V \setminus \{ i \} : \text{if } k \in S \text{ and } (k, j) \in E \text{, then } j \in S \}$$

$$A''' = \{ S \subseteq V : \text{if } i \in S \text{ and } (k, j) \in E \text{, then } j \in S \} ,$$

which proves that the output of Algorithm 3 is

$$A' \cup A''' = \{ S \subseteq V : \text{if } k \in S \text{ and } (k, j) \in E \text{, then } j \in S \} .$$

This completes our proof of the correctness of Algorithm 3.

To conclude our proof of Lemma 11, we analyze the computation time of Algorithm 2. Indeed, we recall that the computation time required for line (1) in Algorithm 2 is \( O(n^2 M) \). In what follows, we assume that all directed graphs are stored as adjacency lists. Under this assumption, our analysis of the computation time for line (2) in Algorithm 3 is split into the following two intermediary claims, denoted by Claim EC.5 and EC.6. In our first intermediary claim, presented below as Claim EC.5, we establish the computation time required for lines (2b), (2d), (2f), and (2g) of Algorithm 3.

**Claim EC.5.** If \( G \equiv (V, E) \) is a directed acyclic graph that is a transitive closure with \( |V| \geq 1 \), then lines (2b), (2d), (2f), and (2g) of Algorithm 3 can be performed in \( O(|V| \times \text{RecursiveStep}(G)) \) time.

**Proof.** We observe that the set of vertices \( \{ i \} \cup \{ \ell : (\ell, i) \in E \} \) in line (2b) of Algorithm 3 can be queried and stored as a hash table in \( O(|\{ i \} \cup \{ \ell : (\ell, i) \in E \}|) = O(|V|) \) computation time. Therefore, the directed graph \( G' \equiv (V', E') \) in line (2b) of Algorithm 3 can be constructed from scratch in \( O(|V| + |E|) \) computation time. By identical reasoning, we observe that the directed graph \( G'' \equiv (V'', E'') \) in line (2d) of Algorithm 3 can be constructed in \( O(|V| + |E|) \) computation time. It is easy to see for each \( S \in A' \) that \( S \cup \{ i \} \cup \{ \ell : (i, \ell) \in E \} \) is the union of three disjoint sets, which implies that the collection on line (2f) of Algorithm 3 can be constructed in a total of \( O \left( \sum_{S \in A'} |S| + |\{ \ell : (\ell, i) \in E \}| \right) = O \left( \sum_{S \in A''} |S| \right) \) computation time. It is similarly easy to see that \( A \equiv A' \cup A''' \) is the union of two disjoint collections, which implies that line (2g) of Algorithm 3 can be performed in \( O \left( \sum_{S \in A'} |S| + \sum_{S \in A''} |S| \right) = O \left( \sum_{S \in A} |S| \right) \) computation time. All combined, we have shown that the computation time required for lines (2b), (2d), (2f), and (2g) of Algorithm 3 is

\[
\underbrace{O(|V| + |E|)}_{(2b)} + \underbrace{O(|V| + |E|)}_{(2d)} + \underbrace{O \left( \sum_{S \in A''} |S| \right)}_{(2f)} + \underbrace{O \left( \sum_{S \in A} |S| \right)}_{(2g)} = O \left( |V| + |E| + \sum_{S \in A} |S| \right) .
\]
Using the fact that graphs always satisfy the inequality $|\mathcal{E}| \leq |\mathcal{V}|^2$ and the fact that $|S| \leq |\mathcal{V}|$ for all $S \in \mathcal{A}$, the above computation time simplifies to $\mathcal{O}(|\mathcal{V}|^2 + |\mathcal{V}| \times |\mathcal{A}|)$. Since $\mathcal{A} \equiv \text{RecursiveStep}(\mathcal{G})$ is the output of Algorithm 3, and since it is easy to see that $|\mathcal{A}| \geq |\mathcal{V}|$, our proof of Claim EC.5 is complete. □

In our second intermediary claim, presented below as Claim EC.6, we use Claim EC.5 to establish the computation time for Algorithm 3 for any valid input.

**Claim EC.6.** If $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$ is a directed acyclic graph that is a transitive closure with $|\mathcal{V}| \geq 1$, then the computation time for Algorithm 3 is $\mathcal{O}(|\mathcal{V}|^2 \times |\text{RecursiveStep}(\mathcal{G})|)$.

**Proof.** Let $T(\mathcal{G})$ denote the computation time required to run Algorithm 3 for any valid input $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$. It follows from Claim EC.5 that $T(\mathcal{G})$ can be represented by an asymptotic recurrence of the form

$$T(\mathcal{G}) = \begin{cases} |\mathcal{V}| \times |\text{RecursiveStep}(\mathcal{G})| + T(\mathcal{G}') + T(\mathcal{G}''), & \text{if } \mathcal{V} \neq \emptyset, \\ 1, & \text{if } \mathcal{V} = \emptyset, \end{cases} \quad \text{(EC.11)}$$

where $\mathcal{G}'$ and $\mathcal{G}''$ are the subgraphs constructed in Algorithm 3 on lines (2b) and (2d). We will now prove that the above recurrence satisfies the following inequality:

$$T(\mathcal{G}) \leq (|\mathcal{V}| + 1)^2 \times |\text{RecursiveStep}(\mathcal{G})|.$$ 

Indeed, the above inequality clearly holds in the base case. Now assume by induction that the above inequality holds for all valid inputs graphs to Algorithm 3 with up to $p - 1$ vertices, and consider any valid input $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$ to Algorithm 3 for which the number of vertices is $|\mathcal{V}| = p$. For this input, we have

$$T(\mathcal{G}) = |\mathcal{V}| \times |\text{RecursiveStep}(\mathcal{G})| + T(\mathcal{G}') + T(\mathcal{G}'')$$

$$\leq |\mathcal{V}| \times |\text{RecursiveStep}(\mathcal{G})| + (|\mathcal{V}'| + 1)^2 \times |\text{RecursiveStep}(\mathcal{G}')| + (|\mathcal{V}''| + 1)^2 \times |\text{RecursiveStep}(\mathcal{G}'')|$$

$$\leq |\mathcal{V}| \times |\text{RecursiveStep}(\mathcal{G})| + |\mathcal{V}|^2 \times |\text{RecursiveStep}(\mathcal{G}')| + |\mathcal{V}|^2 \times |\text{RecursiveStep}(\mathcal{G}'')|$$

$$= |\mathcal{V}| \times |\text{RecursiveStep}(\mathcal{G})| + |\mathcal{V}|^2 \times |\text{RecursiveStep}(\mathcal{G}')| + |\mathcal{V}|^2 \times |\text{RecursiveStep}(\mathcal{G}'')|$$

$$\leq (|\mathcal{V}| + 1)^2 \times |\text{RecursiveStep}(\mathcal{G})|,$$

where the first line follows from (EC.11) and from the fact that $\mathcal{V} \neq \emptyset$, the second line follows from the induction hypothesis, the third line follows from the facts that $|\mathcal{V}'| < |\mathcal{V}|$ and $|\mathcal{V}''| < |\mathcal{V}|$, the fourth line follows from the fact that $|\text{RecursiveStep}(\mathcal{G})| = |\text{RecursiveStep}(\mathcal{G}')| + |\text{RecursiveStep}(\mathcal{G}'')|$, and the final line follows from algebra. This concludes our proof of Claim EC.6. □

Our analysis of the computation time for line (2) of Algorithm 2 follows readily from Claim EC.6. Indeed, we observe that the graph $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$ that was constructed on line (1) of Algorithm 2 satisfies $|\mathcal{V}'| = n + 1$ and $|\mathcal{E}'| \leq (n + 1)^2$. Therefore, it follows from Claim EC.6 that line (2) of Algorithm 2 requires $\mathcal{O}((n + 1)^2 \times |\text{RecursiveStep}(\mathcal{G})|) = \mathcal{O}(n^2 |\hat{\mathcal{S}}|)$ computation time. Since the loop on line (3) of Algorithm 2 can be performed in $\mathcal{O}(\sum_{S \in \mathcal{G}} |S|) = \mathcal{O}(n |\hat{\mathcal{S}}|)$ computation time, we have shown that the total computation time for Algorithm 2 is

$$\mathcal{O}(n^2 M) + \mathcal{O}(n^2 |\hat{\mathcal{S}}|) + \mathcal{O}(n |\hat{\mathcal{S}}|) = \mathcal{O} \left( n^2 (M + |\hat{\mathcal{S}}|) \right).$$

This concludes our proof of Lemma 11. □
F.2. Proof of Lemma 12

We readily observe that the collection of assortments \( \hat{S} \) is a subset of

\[
\hat{S} \triangleq \{ S \in \mathcal{S} : \text{if } i^* \in S, \ r_{i^*} < r_i, \text{ and } M_{i^*} = M_i, \text{ then } i \in S \},
\]

where we recall from the beginning of §3 that \( M_i \) is defined as the subset of the previously-offered assortments \( \mathcal{M} \) \( \equiv \{1, \ldots, M\} \) that offered product \( i \). For each subset of previously-offered assortments \( \mathcal{C} \subseteq \mathcal{M} \), let the products which are offered only in the assortments in \( \mathcal{C} \) be denoted by

\[
\mathcal{N}(\mathcal{C}) \triangleq \{ i \in \mathcal{N}_0 : M_i = \mathcal{C} \}.
\]

Equipped with the above definitions, we observe that \( \{ \mathcal{N}(\mathcal{C}) : \mathcal{C} \subseteq \mathcal{M} \} \) is the collection of subsets of products which always appear together in the previously-offered assortments, and we readily observe that \( |\{ \mathcal{N}(\mathcal{C}) : \mathcal{C} \subseteq \mathcal{M} \}| = 2^M \). Therefore,

\[
|\hat{S}| \leq |\tilde{S}| \leq \prod_{\mathcal{C} \subseteq \mathcal{M}} (|\mathcal{N}(\mathcal{C})| + 1) \leq (n + 2)^M. \tag{EC.12}
\]

Indeed, the first inequality on line (EC.12) holds because the collection of assortments \( \hat{S} \) is a subset of the collection of assortments \( \tilde{S} \). To see why the second inequality on line (EC.12) holds, consider any arbitrary subset of previously-offered assortments \( \mathcal{C} \subseteq \mathcal{M} \), and let the products in \( \mathcal{N}(\mathcal{C}) \) be indexed in ascending order by revenue; that is, let the products that comprise \( \mathcal{N}(\mathcal{C}) \) be denoted by \( i_1^C, \ldots, i_{|\mathcal{N}(\mathcal{C})|}^C \), where \( r_{i_1^C} < \cdots < r_{i_{|\mathcal{N}(\mathcal{C})|}^C} \). Then, we observe from the definition of the collection \( \tilde{S} \) that every assortment \( S \in \tilde{S} \) must satisfy the condition \( \{i_0^C, \ldots, i_{|\mathcal{N}(\mathcal{C})|}^C\} \subseteq S \) and \( i_j^C \notin S \) for some \( j \in \{0, \ldots, \mathcal{N}(\mathcal{C})\} \). Since \( \mathcal{C} \subseteq \mathcal{M} \) was chosen arbitrarily, we have shown that

\[
\tilde{S} \subseteq \bigcup_{\mathcal{C} \subseteq \mathcal{M}} \mathcal{F}^C : \mathcal{F}^C \in \{ \emptyset, \{i_1^C\}, \{i_1^C, i_2^C\}, \ldots, \{i_1^C, \ldots, i_{|\mathcal{N}(\mathcal{C})|}^C\} \},
\]

which proves that the second inequality on line (EC.12) holds. The third inequality on line (EC.12) follows from the fact that \( |\{ \mathcal{N}(\mathcal{C}) : \mathcal{C} \subseteq \mathcal{M} \}| = 2^M \) and from the fact that \( |\mathcal{N}(\mathcal{C})| \leq n + 1 \) for every \( \mathcal{C} \subseteq \mathcal{M} \). We have thus proven that \( |\tilde{S}| \) is at most \( (n + 2)^M \), which concludes our proof of Lemma 12. \( \square \)

F.3. Proof of Lemma 13

To construct the set of tuples of products \( \mathcal{L} \), we iterate over each tuple of products \( (i_1, \ldots, i_M) \in S_1 \times \cdots \times S_M \). For each such tuple of products, we can construct the corresponding directed graph \( \mathcal{G}_{i_1 \cdots i_M} \) that is described in the beginning of §3.2. According to Lemma 2 in §3.2, the tuple of products \( (i_1, \ldots, i_M) \) satisfies \( (i_1, \ldots, i_M) \in \mathcal{L} \) if and only if the directed graph \( \mathcal{G}_{i_1 \cdots i_M} \) is acyclic. We can check whether a directed graph is acyclic by using the well-known topological sorting algorithm (Ahuja et al. 1988, p.79). Therefore, our algorithm for constructing the set of tuples of products \( \mathcal{L} \) is to iterate over each tuple of products \( (i_1, \ldots, i_M) \in S_1 \times \cdots \times S_M \) and, for each such tuple of products, to check whether the tuple of products satisfies \( (i_1, \ldots, i_M) \in \mathcal{L} \) by performing the topological sorting algorithm on the directed graph \( \mathcal{G}_{i_1 \cdots i_M} \).

The computation time of the above algorithm for constructing \( \mathcal{L} \) can be analyzed as follows. The number of iterations in the algorithm is equal to \( |S_1 \times \cdots \times S_M| = \mathcal{O}(n^M) \), where we observe that it is trivial from a
computation time analysis to enumerate and iterate over the tuples of products in $S_1 \times \cdots \times S_M$. For each iteration, we must construct a directed graph $G_{i_1, \ldots, i_M}$ that has $n+1 = O(n)$ vertices and $\sum_{m \in M} |S_m| + 1 = O(Mn)$ directed edges. Constructing this directed graph requires examining each product in each of the assortments $S_1, \ldots, S_M$, and so we can construct $G_{i_1, \ldots, i_M}$ in $O(Mn)$ computation time. The computation time for the topological sort algorithm on a directed graph is equal to the number of vertices plus the number of directed edges in the directed graph, and so we can check whether $G_{i_1, \ldots, i_M}$ is acyclic in $O(n + Mn) = O(Mn)$ computation time. Therefore, we have shown that our algorithm for constructing the set of tuples of products $\mathcal{L}$ requires $O(n^M \times (Mn + Mn)) = O(Mn^{M+1})$ computation time. We also observe from the inequality $|\mathcal{L}| \leq |S_1 \times \cdots \times S_M|$ that $|\mathcal{L}| = O(n^M)$, which concludes our proof of Lemma 13. □

F.4. Proof of Lemma 14

Consider any assortment $S \in \hat{S}$ and any tuple of products $(i_1, \ldots, i_M) \in \mathcal{L}$. To motivate our algorithm for computing $\rho_{i_1, \ldots, i_M}(S)$, we begin by recalling the directed graph $G_{i_1, \ldots, i_M}$ corresponding to the tuple of products $(i_1, \ldots, i_M)$ that is described in the beginning of §3.2. According to Lemma 2 in §3.2, it follows from the fact that $(i_1, \ldots, i_M) \in \mathcal{L}$ that $G_{i_1, \ldots, i_M}$ is a directed acyclic graph. Moreover, we recall that

$$\rho_{i_1, \ldots, i_M}(S) = \min_{i \in S : i \nLeftarrow_{i_1, \ldots, i_M} S} r_i = \min_{i \in S : i \nLeftarrow_{i_1, \ldots, i_M} S} \text{ for all } i_m \in S,$$

where the first equality follows from Proposition 2 in §3.2 and the second equality follows from Definition 5 in §3.2. We recall from Definition 4 in §3.2 that the notation $i \nLeftarrow_{i_1, \ldots, i_M} j$ means that there is no directed path from vertex $j$ to vertex $i$ in the graph $G_{i_1, \ldots, i_M}$. Stated in words, line (EC.13) shows that $\rho_{i_1, \ldots, i_M}(S)$ is equal to the minimum revenue $r_i$ among all of the vertices $i$ in the directed acyclic graph $G_{i_1, \ldots, i_M}$ which do not have a directed path to a vertex $i_m$ for any previously-offered assortment $m \in \mathcal{M}$ which satisfies $i_m \in S$.

Based on the above observations, we arrive at the following straightforward algorithm (presented as Algorithm 4) for computing $\rho_{i_1, \ldots, i_M}(S)$. The algorithm begins on line (1) of Algorithm 4 by constructing the directed acyclic graph $G_{i_1, \ldots, i_M}$. In particular, it follows from identical reasoning as in the proof of Lemma 13 that $G_{i_1, \ldots, i_M}$ can be constructed in $O(Mn)$ computation time and that this graph is comprised of $O(n)$ vertices and $O(Mn)$ directed edges. In line (2) of Algorithm 4, we iterate over each of the previously offered assortments $m \in \mathcal{M}$ which satisfy $i_m \in S$. For each such previously-offered assortment $m$, we mark all of the vertices in the graph which have a directed path to vertex $i_m$. Since the graph has $O(Mn)$ directed edges, and assuming henceforth that $G_{i_1, \ldots, i_M}$ is stored in memory as an adjacency list, it is easy to see using a standard graph traversal algorithm like depth-first search that the loop in line (2) of Algorithm 4 can be performed in a total of $O(M \times Mn) = O(M^2n)$ computation time. Finally, line (3) of Algorithm 4 iterates over all of the vertices in at most $O(Mn)$ computation time and outputs the minimum $r_i$ among all of the vertices $i$ which are unmarked. The correctness of Algorithm 4 follows immediately from our earlier reasoning on line (EC.13), and the total computation time required for Algorithm 4 is

$$O(Mn) + O(M^2n) + O(n) = O(M^2n).$$

This concludes our proof of Lemma 14. □
\textbf{Construct-}p(S, (i_1, \ldots, i_M), \mathcal{M}, r)

\textbf{Inputs:}
- An assortment, \( S \in \mathcal{S} \).
- A tuple of products, \((i_1, \ldots, i_M) \in \mathcal{L}\).
- The collection of previously-offered assortments, \( \mathcal{M} \equiv \{S_1, \ldots, S_M\} \).
- The revenues of the products, \( r \equiv (r_0, r_1, \ldots, r_n) \).

\textbf{Output:}
- The quantity \( \rho_{i_1 \ldots i_M}(S) \).

\textbf{Procedure:}
1. Construct the directed acyclic graph \( G_{i_1 \ldots i_M} \).
2. For each previously-offered assortment \( m \in \mathcal{M} \):
   (a) If \( i_m \in S \):
      i. Mark each unmarked vertex in \( G_{i_1 \ldots i_M} \) that has a directed path to vertex \( i_m \).
3. Output the minimum \( r_i \) among all unmarked vertices \( i \) in \( G_{i_1 \ldots i_M} \) and terminate.

Algorithm 4: A procedure for computing \( \rho_{i_1 \ldots i_M}(S) \).