Problem definition: Motivated by several practical selling scenarios that require previous purchases to unlock future options, we consider a multi-stage assortment optimization problem, where the seller makes sequential assortment decisions with commitment, and the customer makes sequential choices to maximize her expected utility. Methodology/results: We start with the two-stage problem and formulate it as a dynamic combinatorial optimization problem. We show that this problem is polynomial-time solvable when the customer is fully myopic or fully forward-looking. In particular, when the customer is fully forward-looking, the optimal policy entails that the assortment in each stage is revenue-ordered and a product with higher revenue always leads to a wider range of future options. Moreover, we find that the optimal assortment in the first stage must be smaller than the optimal assortment when there were no second stage and the optimal assortment in the second stage must be larger than the optimal assortment when there were no first stage. When the customer is partially forward-looking, we show that the problem is NP-hard in general. In this case, we establish the polynomial-time solvability under certain conditions. In addition, we propose a 2-approximation algorithm in the general setting. We further extend the above results to the multi-stage problem with an arbitrary number of stages, for which we derive generalized structural properties and efficient algorithms. Managerial implications: Firms can benefit from our study and improve their sequential assortment strategies when their interaction with each customer consists of multiple stages.

Key words: Multi-stage choice model, assortment optimization, revenue management.
the seller and the customer will happen. Much research has been done to tackle such single-stage problems.

However, in many practical selling scenarios, the interaction between a seller and an incoming customer consists of multiple stages, with multiple (possibly correlated) transactions taking place in a period of time. In such multi-stage selling scenarios, the seller usually has commitment power over multiple stages, and is able to use the availability of some future purchase options to incentivize his customers to choose some particular purchase options in earlier stages. Meanwhile, under such strategies, customers are often forward-looking. Usually, as a customer spends more money in initial stages, she will get access to more attractive purchasing opportunities in the future (e.g., exclusive deals, special offers, upgrade opportunities); however, if she spends less money in initial stages, then she may face limited purchase options in the future. The customer will weigh the benefits of these future options in her purchasing decisions (note that at the time when the customer makes a decision, the exact values of the benefits of the future options are uncertain, and the customer’s current decision is influenced by her expectation of the future). Such scenarios are widespread in practice. We provide a few examples as below.

- **Airline tickets and ancillaries.** Airline ancillary revenue (i.e., revenue from non-ticket sources such as baggage, upgrades, seat selection, change/cancellation fees, boarding priority, onboard services, etc.) has experienced a sharp 485% increase over nine years (from $22.6 billion in 2010 to $109.5 billion in 2019; see IdeaWorksCompany 2019), clearly exhibiting the trend that airlines are seeking to understand the complete customer purchasing cycle and take advantage of not only the booking stage — but also the post-booking stage. A key challenge is then how to design and combine the offerings in each stage to achieve better differentiation and increase revenue. A common design principle in airline practice is to let customers who bought more expensive tickets have more ancillary purchase options. For example, according to United Airlines, American Airlines and Delta Air Lines’ policies, customers who bought “basic economy” fares are generally ineligible for upgrades and changing/canceling flights, and would face strict restrictions regarding paid baggage and boarding priority (Clark.com 2018). On the other hand, customers who bought more expensive tickets generally have more flexible options like upgrading with miles and same-day flight changes (American Airlines 2018). Note that many of these ancillaries should be viewed as second-stage options or assortments rather than immediate benefits, since the exact values of those options (e.g., paid upgrades, flexible changes/cancellation, or cheaper additional baggage) are often uncertain at the time when customers purchase their tickets, and customers would make finalized ancillary decisions after they realize the values of those options (in a future stage). Therefore, one

---

1 In this paper, we use “he” to denote a seller, and use “she” to denote a customer.
can view that airlines are facing a multi-stage selling scenario, with customers making sequential choices over multiple stages.

- **Sports season ticket selling.** Many professional sports leagues have both a regular season and a playoff each year, and the tickets are sold separately. While the playoffs tickets are arguably more valuable and more difficult to obtain, it is common practice for sports teams to use guaranteed access to exclusive presales of playoffs tickets (with better seats and cheaper prices) to attract their fans to purchase regular season memberships. For example, the majority of NFL Playoff tickets are sold to the home team’s season ticket holders (Seatgeek 2021), and almost all NBA teams highlight the benefits of being their season ticket members by promising their members the access to presales of playoffs tickets before the general public (NBA 2022a,b). Here, the selling strategy is to link different levels of regular season ticket memberships (in the first stage) to different assortments of playoffs tickets (in the second stage). The payoff tickets should be viewed as second-stage options since they will only be available after the regular season and their exact values are uncertain at the time when fans purchase regular season memberships.

Despite the prevalence of such selling scenarios, to the best of our knowledge, there is no existing research studying the seller’s optimal strategy (in terms of a multi-stage assortment policy) in such scenarios. More generally, we can interpret this practical challenge to a general assortment optimization problem as follows:

- The seller commits to a multi-stage assortment policy at the beginning, specifying how the customer’s choices in previous stages will determine the assortments in subsequent stages.
- In each stage, the seller offers an assortment of products based on his policy and the customer’s choice history, and the customer makes a rational choice to maximize her (discounted) expected long-term utility.
- The objective of the seller is to determine the optimal assortment policy that obtains the largest expected revenue.

The above problem can be viewed as a multi-stage extension of the traditional assortment optimization problem. We refer to this problem as the *multi-stage assortment optimization* problem. Clearly, this problem cannot be tackled by the traditional assortment optimization framework, as its multi-stage nature makes it much more complicated (see also Section 2 for a comparison with related work). In this paper, we attempt to address this challenge and answer the following research questions:

1. How should we model the customer’s choice behavior in such a multi-stage selling scenario?
2. Given the customer’s choice behavior, what is the seller’s optimal assortment policy? In other words, how can we efficiently solve the multi-stage assortment optimization problem?
3. Are there any interesting structural properties in the seller’s optimal assortment policy?
In particular, in the main part of our paper, we focus on the two-stage assortment optimization problem. Compared with the general multi-stage problem, we believe that this two-stage problem illustrates clearer and more intuitive insights, while retaining most of the important features of the multi-stage problem. Indeed, as we will show in Appendix A, most of our results established for the two-stage problem naturally extend to the multi-stage problem, with key insights and algorithmic ideas remaining almost unchanged.

**Overview of Results.** To begin our study, we first build a multi-stage choice model for the customer in the multi-stage interaction. By incorporating an expected future utility term into the customer choice model, we derive the multi-stage choice model based on the random-utility-maximization principle (specifically, we adopt the multinomial logit (MNL) model in this paper). In particular, in the multi-stage choice model, there is a discount factor $\gamma$ ($0 \leq \gamma \leq 1$) that describes the customer’s degree of being forward-looking, i.e., how she discounts her future utility. As we will show, this parameter plays an important role in determining the structure of the optimal solution as well as the complexity of the problem.

Based on the multi-stage choice model, we formulate the associated two-stage assortment optimization problem, and study its properties in detail. We discuss the two-stage problem in three cases: the fully-myopic-customer ($\gamma = 0$) case, the fully-forward-looking-customer ($\gamma = 1$) case, and the partially-forward-looking-customer ($0 < \gamma < 1$) case. We obtain the following results:

- When the customer is fully myopic ($\gamma = 0$), we show that the two-stage assortment optimization problem is equivalent to two separate single-stage problems, thus can be easily solved in $O(n)$ time, and the optimal solution follows the structure of the optimal solution to the classical single-stage problem, with the optimal assortment in each stage being revenue-ordered\(^2\).

- When the customer is fully forward-looking ($\gamma = 1$), we show that the problem is polynomial-time solvable, and the optimal policy should satisfy the following properties: (1) the first-stage assortment is revenue-ordered, (2) the second-stage assortment is revenue-ordered, and (3) a product with higher revenue in the first stage should always lead to a wider range of offering in the second stage. These properties are intuitive and are also consistent with practice. Also, interestingly, we find that compared with the optimal assortment in each stage alone, in the two-stage problem, the seller should offer fewer products in the first stage, and offer more products in the second stage. This is because the seller is using “more options in the second stage” as an incentive to encourage the customer to buy more expensive products in the first stage. Based on these findings, we propose an efficient algorithm that solves the problem in $O(n^2 \log(n))$ time.

\(^2\) A revenue-ordered assortment refers to an assortment that consist of a certain number of products with the highest revenue; see Definition 1.

Electronic copy available at: https://ssrn.com/abstract=3243742
• When the customer is partially forward-looking (0 < γ < 1), we show that the optimal first-stage assortment is still revenue-ordered; moreover, as long as the customer makes a purchase in the first stage, the optimal second-stage assortment is also revenue-ordered. However, if the customer buys nothing in the first stage, then the second-stage assortment may not necessarily be revenue-ordered. We find that compared with the γ = 1 case, this subtle difference brings significant challenge for solving the problem: in fact, the problem is generally NP-hard for the 0 < γ < 1 case. Still, we find that the problem is polynomial-time solvable under certain structural conditions, in which case we give precise characterizations of the optimal policy; when such conditions do not hold, we provide a fast and simple 2-approximation algorithm (i.e., an algorithm whose expected revenue is always at least half as large as the optimal one) for the general setting.

After understanding the structure of the optimal policy, we study the performance of a simpler class of policies — static policies (defined in Section 5). These policies are common in practice and intuitively appealing. By giving the necessary and sufficient conditions for static policies to be optimal when γ = 0 and γ = 1 and giving the necessary conditions for static policies to be optimal when 0 < γ < 1, we show that static policies are generally sub-optimal for the problem. We also show that as long as γ > 0, static policies fail to provide a constant-factor approximation for the problem.

Moreover, we extend our results to two more general settings. The first generalization allows the set of products that the seller uses to determine his assortment in each stage to depend on the customer’s purchasing history. This extension enables our model to tackle a wider range of problems in practice. The second setting considers the multi-stage assortment optimization problem with an arbitrarily given number of stages (in particular, we focus on the important case where the customer is fully forward-looking). For both of the two generalizations, we give precise characterizations of the optimal policy, and derive efficient algorithms. These extended results indicate that the insights and methodologies that we develop for the two-stage model are quite general.

Outline. The rest of the paper is organized as follows. In Section 2, we review the relevant literature. In Section 3, we derive our multi-stage choice model. In Section 4, we formulate the two-stage assortment optimization problem, and study its properties in three cases: the fully-myopic-customer case (Section 4.1), the fully-forward-looking-customer case (Section 4.2), and the partially-forward-looking-customer case (Section 4.3). In Section 5, we study the performance of static policies in this problem. In Section 6, we consider a generalized model that allows choice-dependent product sets. In Appendix A, we extend our results to the multi-stage assortment problem with an arbitrarily given number of stages. All proofs are deferred to the appendix.
2. Literature Review

In this section, we review related literature to our work. Broadly speaking, our work is related to three streams of research: assortment optimization, bundling, and dynamic discrete choice models. In the following, we will mainly review literature in these three areas that are related to our work.

**Assortment optimization.** Assortment optimization refers to the problem of determining a subset of products (taken from a larger set of products) to offer to customers in order to maximize expected revenue. It is a central topic in revenue management, with wide applications in practice; see Kök et al. (2009) for a comprehensive survey and Kök and Fisher (2007) and Fisher and Vaidyanathan (2014) for examples of successful real-world implementations. A key step in formulating an assortment optimization problem is to identify a choice model that captures the choice behavior of customers. In their seminal work, Talluri and van Ryzin (2004) first consider the assortment optimization problem under the MNL model. They show that revenue-ordered assortments are optimal for the problem. This property is intuitively appealing, and enables the seller to find an optimal assortment in linear time. Since then, assortment optimization under the MNL model has been an active research area. Rusmevichientong et al. (2010) study this problem with a cardinality constraint on the offered assortment. They find that although the revenue-ordered assortments are no longer optimal in this case, the problem can still be reformulated as a linear program and can be solved in quadratic time. Davis et al. (2013) consider an extended problem where the constraints on the offered assortment can be captured by a totally unimodular constraint matrix. They show that the problem can be reformulated as a linear program, thus can be solved in polynomial time. Along with the above line of research, there is also research studying assortment optimization under other choice models. In particular, Davis et al. (2014) and Li and Rusmevichientong (2014) study the assortment optimization problem under the two-level nested logit model. They show that this problem is polynomial-time solvable and present efficient algorithms for this problem. Li et al. (2015) further study the assortment optimization problem under the multi-level nested logit model, for which they provide an efficient algorithm.

In this paper, we propose a novel multi-stage choice model beyond traditional single-stage choice models, and study the associated assortment optimization problem. We note that there are several recent papers on assortment optimization which also propose new choice models that involve multiple stages; see Flores et al. (2019) for a two-stage choice model that extends the MNL model, Liu et al. (2019) for a multi-stage choice model motivated by healthcare appointment booking, and Fata et al. (2019), Liu et al. (2020), Gao et al. (2020) for several more recent models that further extend the previous ones. However, all these papers make a common assumption — each customer buys at most one product throughout the entire process (i.e., the interaction immediately ends once a single purchase occurs). By contrast, our choice model allows the customer to sequentially buy
multiple products over multiple stages, which allows the seller to generate more revenue by incentivizing the customer to buy more. Such differences make our multi-stage assortment optimization problem fundamentally different from existing ones in the literature. As a consequence, we obtain very different results and novel insights.

**Bundling.** Our work is conceptually related to the research on bundling. Bundling refers to the strategy of simultaneously selling multiple products in particular combinations and at particular prices to increase revenue. For example, a firm may sell a bundle of products for a lower price than they would charge if the customer bought all of them separately, such that the customer would be willing to simultaneously buy the whole bundle and the firm would obtain more revenue. Bundling has been the focus of many papers in different disciplines including economics, marketing, and operations management. Two key decisions studied in the literature are the design and the price optimization of product bundles, see Hanson and Martin (1990) and Venkatesh and Mahajan (2009) for some comprehensive reviews.

Although the bundling problem and the multi-stage assortment optimization problem share a similar objective of maximizing revenue when the customer purchases multiple products, the selling scenarios captured by the two problems are fundamentally different (and in some sense complementary): the bundling problem studies a scenario where the customer simultaneously purchases multiple products, while the multi-stage assortment optimization problem studies a scenario where the customer sequentially purchases multiple products. In particular, the bundling problem is a single-stage problem, as the customer realizes all her valuations and makes her choice only for once; by contrast, our problem is a multi-stage problem, where the customer sequentially realizes her valuations and makes multiple choices. Technically, our problem is challenging because of the need to solve a dynamic optimization problem. Practically, the multi-stage nature of our model is important and inherited in many real-world applications, which needs a thorough study.

**Dynamic discrete choice models.** Finally, our work is related to a stream of research in econometric literature on dynamic discrete choice models. Dynamic discrete choice model describes a utility-maximizing customer’s long-term choice behavior in dynamic settings. A dynamic discrete choice model is based on an associated single-stage discrete choice model, and is derived via dynamic programming (for finite stages) or Markov decision process (for infinite stages). Dynamic discrete choice models have played important roles in empirical research in many fields (e.g., industrial organization, labor economics, and marketing). Seminal papers include Wolpin (1984), Miller (1984), Pakes (1986) and Rust (1987). The estimation of such models has also been studied extensively. Here we refer to three recent surveys of dynamic discrete choice models: Aguirregabiria and Mira (2010), Arcidiacono and Ellickson (2011) and Keane et al. (2011).
It is worth noting that our proposed multi-stage choice model belongs to the class of dynamic discrete choice models. This fact brings two merits to our multi-stage choice model. First, the wide empirical application of dynamic discrete choice models provides justification for our multi-stage choice model. Second, since existing literature has proposed many methods to estimate general discrete choice models, the estimation problem of our multi-stage choice model becomes easy — to estimate our multi-stage choice model, instead of creating new estimation methods on our own, we can apply some well-developed estimation techniques to achieve our goal. In particular, we can efficiently estimate the structural preference parameters with full data via the Nested Fixed Point algorithm (Rust 1987) or Hotz-Miller-type methods (Hotz and Miller 1993), and estimate the parameters with censored data via adjusted versions of the Expectation-Maximization (EM) algorithm (Arcidiacono and Miller 2011).

3. Model
We consider a seller who offers a certain number of products to the customers. In our model, customers purchase in a number of stages. In each stage, a customer can buy at most one product and the products offered to the customer in each stage can depend on the purchase decisions of the customer in previous stages. More precisely, we consider $T$ stages indexed by $t = 1, 2, \ldots, T$. In each stage $t$, the seller can choose to offer a subset of products from $N_t = \{1, 2, \ldots, n_t\}$ to the customer. Note that while we use the same index $i = 1, 2, \ldots$ to denote “product $i$” in different stages, “product $i$” can be completely different in different stages (i.e., the use of the same index $i$ is only for notational simplicity) — throughout the paper, we will always provide the necessary stage information whenever we mention “product $i$”. We use $p_i(t)$, $i = 1, \ldots, n_t$ to denote the price associated with product $i$ in stage $t$. In our model, $p_i(t)$s are exogenously given (e.g., because the seller has announced the prices for the products or they are determined by competitor’s prices for similar products). Furthermore, without loss of generality, we assume that the products in each stage are ordered such that $p_1(t) \geq p_2(t) \geq \cdots \geq p_{n_t}(t)$ for all $t = 1, 2, \ldots, T$. In each stage $t$, the seller first determines an assortment $S_t \subseteq N_t$ to offer. Then the customer can choose at most one product from $S_t$ in that stage and the seller will collect the revenue of the corresponding product. The customer can also choose a “no purchase” option denoted by 0 in each stage, which earns a revenue of $p_0(t) = 0$ for the seller.

We use a random utility model to capture customer’s utility toward the products. In particular, a customer has utility

$$u_i(t) = \mu_i(t) + \epsilon_i(t), \quad i = 0, \ldots, n_t, \quad t = 1, \ldots, T$$

One can also view $p_i(t)$ as the profit of each product in the case when profit maximization is considered.
for product \( i \) of stage \( t \), where \( \mu_i^{(t)} \)s are the deterministic part of the utility (which already takes into account the disutility due to the price). In the following, we adopt the multinomial logit (MNL) framework and assume \( \epsilon_i^{(t)} \)s are i.i.d. standard Gumbel random variables with mean zero. Furthermore, we normalize the mean utility of the no-purchase option in each stage \( t \) to be \( \mu_0^{(t)} = 0 \). We assume customers are aware of the deterministic utilities \( \mu_i^{(t)} \)s in each stage at the beginning, but the random parts of the utilities \( \epsilon_i^{(t)} \)s are only realized at stage \( t \).

In this paper, we study the assortment policies for the seller and also the corresponding purchase decision of the customer. At each stage \( t \), we define \( h_t^s \triangleq (S_1, \ldots, S_{t-1}) \) to be the offering history of the seller, and define \( h_t^c \triangleq (c_1, \ldots, c_{t-1}) \) to be the choices made by the customer. Here \( S_t \subseteq \mathcal{N}_t \) is the assortment offered by the seller at stage \( t \) and \( c_t \in S_t \cup \{0\} \) is the product chosen by the customer. For notational simplicity, we let \( h_1^s = c_1^s = \text{null} \).

With the above notations, we define the seller’s policy \( \Pi \) to be a set of mappings \( \Pi = \{f_t, t = 1, \ldots, T\} \) where each \( f_t \) maps a pair of \( (h_t^s, h_t^c) \) to an assortment \( S_t \subseteq \mathcal{N}_t \). In this paper, we restrict our attention to deterministic policies, where each \( f_t \) is a deterministic function. We note that while randomized policies may be appealing in theory (see Appendix B.4 for an example where randomized policies perform better than deterministic policies), they are hard to implement in practice due to the limited rationality of customers and stronger information requirements.

In the following, we consider a customer’s choice under a policy of the seller. Note that a customer’s purchase decision in previous stages may affect the subsequent assortments she faces, which in turn affects her future utility. Therefore, a rational customer will take her future utility into account when she makes purchase decisions.\(^4\) In our model, we assume there is a discount factor \( \gamma \in [0, 1] \) for a customer’s utility per stage. A smaller \( \gamma \) means that the customer is more myopic (\( \gamma = 0 \) indicates a completely myopic customer), while a larger \( \gamma \) means that the customer is more forward-looking (\( \gamma = 1 \) indicates that the customer weighs utility in each stage equally).

Let \( U_t^\Pi(h_t^s, h_t^c) \) be the maximum expected discounted utility that a customer can receive onward from stage \( t \) when the history is \( (h_t^s, h_t^c) \) and when the seller uses policy \( \Pi \). Note that we assume that the customer is fully aware of the policy of the seller, which is a reasonable assumption, e.g., in the examples discussed in Section 1. In stage \( t \), the seller’s offered assortment is \( S_t = f_t(h_t^s, h_t^c) \). If the customer decides to purchase product \( i \) \( (i \in S_t \cup \{0\}) \), then she will gain an immediate surplus \( u_t^i = \mu_t^i + \epsilon_t^i \), and her expected future utility (from stage \( t+1 \) onward) will be

\(^4\) Note that the exact values of future options are uncertain at the time when a decision is made. Thus a customer’s future utility is in the expectation sense. For example, when a passenger books a flight ticket, she might not know the exact values of the options of upgrading eligibility or flexible changes/cancellation (which depend on random factors in the future), but she will form an expectation of the utilities of those options. Similarly, when a customer purchases an annual membership of an amusement park, she might not know the exact values of the additional options during her future visits, but she will form an expectation of the future utility brought by each membership level.
Thus, for stage $t$ ($t = 1, 2, \ldots, T$), we can write down the following Bellman equation for the customer:

$$U_t^{\Pi}(h^t_s, h^t_c) = \mathbb{E}_\epsilon \left[ \max_{i \in f_t(h^t_s, h^t_c) \cup \{0\}} \left( \mu_i^{(t)} + \epsilon_i^{(t)} \right) + \gamma U_{t+1}^{\Pi} ((h^t_s, f_t(h^t_s, h^t_c)), (h^t_c, i)) \right], \forall t = 1, \ldots, T - 1, h^t_s, h^t_c.$$ 

By using the fact that for standard Gumbel distributions $\epsilon_1, \ldots, \epsilon_k$, $\mathbb{E}_\epsilon [\max_{i=1,\ldots,k}(\mu_i + \epsilon_i)] = \log \left( \sum_{i=1}^{k} \exp(\mu_i) \right)$ (see Ben-Akiva and Lerman 1979), we can further express $U_t^{\Pi}(h^t_s, h^t_c)$ as

$$U_t^{\Pi}(h^t_s, h^t_c) = \log \left( \sum_{i \in f_t(h^t_s, h^t_c) \cup \{0\}} \exp \left( \mu_i^{(t)} + \gamma U_{t+1}^{\Pi} ((h^t_s, f_t(h^t_s, h^t_c)), (h^t_c, i)) \right) \right), \forall t = 1, \ldots, T - 1, h^t_s, h^t_c.$$ 

Note that in the last stage, the customer only considers his current utility, thus

$$U_T^{\Pi}(h^T_s, h^T_c) = \log \left( \sum_{i \in f_T(h^T_s, h^T_c) \cup \{0\}} \exp(\mu_i^{(T)}) \right), \forall h^T_s, h^T_c.$$ 

Therefore, given a seller’s policy, the customer’s optimal decision can be computed by backward induction. We refer to the above choice model as the multi-stage choice model. Our goal is to find the optimal policy for the seller (i.e., the $f_t(\cdot, \cdot)$s for $t = 1, \ldots, T$) such that the expected revenue under the multi-stage choice model is maximized. We call the problem for the seller the multi-stage assortment optimization problem.

### 4. Two-Stage Assortment Optimization Problem

In this section, we consider the case when there are two stages, i.e., $T = 2$. (We shall call the corresponding problem the two-stage assortment optimization problem.) In the following, we formulate this problem as an optimization problem, characterize some of its important properties, describe its computational complexity and propose efficient algorithms to solve it.

According to Section 3, in the two-stage assortment problem, the seller needs to decide the initial assortment $S_1$ as well as assortments $S_2 = f_2(S_1, c_1)$ when the customer chooses each $c_1 \in S_1 \cup \{0\}$. For notational simplicity, in this section, we omit the subscript from $f_2(\cdot)$ and just write it as $f(\cdot)$. The seller’s assortment policy can be represented in Figure 1.
In the following, we formulate the two-stage assortment problem as an optimization problem. We introduce some decision variables as follows. Let $x_i \in \{0, 1\}$ denote whether product $i \in N_1$ is included in $S_1$ in stage 1, and let $y_{ij} \in \{0, 1\}$ denote whether product $j \in N_2$ is included in $f(S_1, i)$ in stage 2. Note that if $x_i = 0$, then $f(S_1, i)$ does not exist, therefore for such $i$ we just let $y_{ij} = 0$ for all $j$. Since the customer can choose the no-purchase option in each stage, we let $x_0 = 1$ for stage 1, and let $y_{i0} = x_i$ ($i \in N_1 \cup \{0\}$) for stage 2. Therefore, we have defined the variables $x_i$ and $y_{ij}$ for $i \in N_1 \cup \{0\}$ and $j \in N_2 \cup \{0\}$, and according to our discussions, they should satisfy the following constraints:

$$
\mathcal{F} = \{(x, y) \mid x_0 = 1, \ y_{ij} \leq x_i, \ y_{i0} = x_i, \ x_i, y_{ij} \in \{0, 1\}, \forall i \in N_1 \cup \{0\}, j \in N_2 \cup \{0\}\}.
$$

Apparently, each feasible $(x, y) \in \mathcal{F}$ corresponds to an assortment policy of the seller, and $\mathcal{F}$ corresponds to the set of all assortment policies.

We now analyze the customer’s choice decision in the two-stage problem. In stage 1, the seller offers $S_1$, and the customer realizes her utilities of the offered products. When making a choice, she considers both her realized current utility and the expected utility in the next stage. If the customer purchases product $i \in S_1 \cup \{0\}$, then she will gain an immediate surplus $u_i^{(1)} = \mu_i^{(1)} + \epsilon_i^{(1)}$, and her expected future utility will be $\log \left( \sum_{j=0}^{n_2} e^{\mu_j^{(2)} y_{ij}} \right)$. Thus, her overall (discounted) utility of choosing $i$ is:

$$
\text{current utility} + \gamma \log \left( \sum_{j=0}^{n_2} e^{\mu_j^{(2)} y_{ij}} \right).
$$

![Figure 1 An Illustration for the Case with $T=2$.](https://ssrn.com/abstract=3243742)
Therefore, according to the MNL model, in stage 1, the customer’s choice probability of product \( i \in \mathcal{N}_1 \cup \{0\} \) is

\[
P_i := e^{\mu_i^{(1)}} \left( \sum_{j=0}^{n_2} e^{\mu_j^{(2)}} y_{ij} \right)^\gamma x_i.
\]

Then, in stage 2, the seller offers \( S_2 = f(S_1, i) \), and the customer realizes her utilities of the offered products to make a choice. Again, according to the MNL model, her conditional probability of choosing \( j \in \mathcal{N}_2 \cup \{0\} \) in stage 2 given choosing \( i \in S_1 \cup \{0\} \) in stage 1 is

\[
P_{ji} := \frac{e^{\mu_j^{(2)}} y_{ij}}{\sum_{j=0}^{n_2} e^{\mu_j^{(2)}} y_{ij}},
\]

and her overall probability of first choosing \( i \) then choosing \( j \) is

\[
P_{ij} := P_i P_{ji} = \frac{e^{\mu_i^{(1)}} \left( \sum_{j=0}^{n_2} e^{\mu_j^{(2)}} y_{ij} \right)^{\gamma-1} e^{\mu_j^{(2)}} x_i y_{ij}}{\sum_{i=0}^{n_1} e^{\mu_i^{(1)}} \left( \sum_{j=0}^{n_2} e^{\mu_j^{(2)}} y_{ij} \right) \gamma x_i}.
\] (1)

Therefore, we can formulate the assortment optimization under the two-stage model as the following optimization problem:

\[
\max_{(x, y) \in \mathcal{F}} \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} P_{ij} \left( p_i^{(1)} + p_j^{(2)} \right).
\] (2)

In the following, we will consider three different cases: \( \gamma = 0, \gamma = 1 \) and \( 0 < \gamma < 1 \) and study the properties of the problem in each of the three cases and propose solution methods.

Before we proceed, we remark that the formulation in (1) is reminiscent of the choice probability in a nested logit choice model (see Anderson et al. 1992). However, there are several key differences between the multi-stage choice model and a nested logit choice model. The main distinction is that in the nested logit choice model, a customer essentially still only chooses one product, while in the multi-stage choice model, the customer chooses a product in each stage, and the seller collects revenue from the choice made by the customer in each stage. Therefore, the choice scenarios captured by these two models are very different. Also, in the multi-stage choice model, the products in each stage could be chosen from different sets and there can be overlapping in each second stage sets (in fact, as we will see, the optimal policy sometimes entails inclusion relations between each second stage sets). This is very different from the nested logit model in which the products in each nest usually are non-overlapping.\(^6\) Furthermore, because of the context of the multi-stage choice

\(^5\) \( P_i \) is a function of \((x, y)\). Here we just write it as \( P_i \) for simplicity. Similar for \( P_{ji} \) and \( P_{ij} \) in this page.

\(^6\) There are some variants of nested choice models that allow overlapping products, e.g., the paired combinatorial logit model, see Koppelman and Wen (2000). However, there are still restrictions on the way the products can overlap in each nest, while we allow basically arbitrary overlapping structure. Thus the structures of those models are different from ours.
model, there must be a no-purchase option in each second-stage set, which is often not the case in a nested logit model. As one will see in the following subsections, these differences will lead to very different optimization problems of the multi-stage assortment optimization problem as well as different solution properties.

4.1. Fully Myopic Customer ($\gamma = 0$)

We first consider the case when the customer is fully myopic, i.e., $\gamma = 0$. In this case, the customer’s decision in the first stage is independent of her decision in the second stage. In other words, she just makes two separate choices in two stages. Therefore, the seller just needs to optimize his assortment in each stage separately. We have the following optimal policy for the seller.

**Proposition 4.1.** Let $S_1^*$ be an optimal assortment for products $N_1$ under the standard MNL model and $S_2^*$ be an optimal assortment for products $N_2$ under the standard MNL model. Then offering $S_1 = S_1^*$ and $f(S_1,i) = S_2^*$, for $i \in S_1 \cup \{0\}$ is an optimal policy for the seller.

Proposition 4.1 is quite obvious thus we omit a formal proof. Next, we define the notion of revenue-ordered assortment.

**Definition 1** In an assortment optimization problem with $n$ products with prices $p_1 \geq p_2 \geq \cdots \geq p_n$, offer sets $\{1, 2, \ldots, j\}$ for $j \leq n$ are called revenue-ordered assortments.

In the seminal paper by Talluri and van Ryzin (2004), the authors showed that there exists a revenue-ordered assortment that is optimal for the assortment optimization problem under the standard MNL model. Therefore, we have the following results for the case when $\gamma = 0$:

**Theorem 4.1.** When $\gamma = 0$, there exists an optimal solution to the two-stage assortment problem with first-stage assortment being $S_1^*$ and second-stage assortments being $f^*(S_1^*, i)$, such that

1. $S_1^*$ is revenue-ordered;
2. $f^*(S_1^*, i)$ ($i \in S_1^* \cup \{0\}$) are identical and revenue-ordered.

Furthermore, the two-stage assortment problem can be solved in $O(n_1 + n_2)$.

Again, we omit the proof for Theorem 4.1. Note that the last statement can be seen from that an assortment optimization with prices $p_1, \ldots, p_n$ and deterministic utilities $u_1, \ldots, u_n$ can be solved by computing the values of $A_k = \sum_{i=1}^k p_i \exp(u_i)$ and $B_k = 1 + \sum_{i=1}^k \exp(u_i)$, and then compare the values of $A_k/B_k$ for $k = 1, \ldots, n$, which can be done in $O(n)$ time.

From Theorem 4.1, we can see that when customers are myopic, there is no dependency between the optimal assortments offered in each stage, and the multi-stage assortment optimization naturally decomposes to an assortment optimization problem in each stage. However, as we will show below, this may not be the case for forward-looking customers ($\gamma > 0$) in which the relation between different stages plays an important role in the seller’s decisions.
4.2. Fully Forward-Looking Customer \((\gamma = 1)\)

In this subsection, we consider the case when the customer is fully forward-looking, i.e., \(\gamma = 1\). In this case, \(P_{ij}\) in (1) can be reduced to the following:

\[
P_{ij} = \frac{e^{\mu_i^{(1)} + \mu_j^{(2)} x_i y_{ij}}}{\sum_{i=0}^{n_1} \sum_{j=0}^{n_2} e^{\mu_i^{(1)} + \mu_j^{(2)} x_i y_{ij}}},
\]

Define \(\tilde{\mu}_{ij} = \mu_i^{(1)} + \mu_j^{(2)}\) and \(\tilde{x}_{ij} = x_i y_{ij}\). We can further rewrite \(P_{ij}\) as follows:

\[
P_{ij} = \frac{e^{\tilde{\mu}_{ij} \tilde{x}_{ij}}}{\sum_{i=0}^{n_1} \sum_{j=0}^{n_2} e^{\tilde{\mu}_{ij} \tilde{x}_{ij}}}.
\]

Let \((i, j)\) denote the customer’s action of “first choosing \(i\) then choosing \(j\)”, which we refer to as her two-stage choice path. It can be observed that \(\tilde{\mu}_{ij} = \mu_i^{(1)} + \mu_j^{(2)}\) is the total expected utility of \((i, j)\). In addition, since \((i, j)\) is a feasible path if and only if \(i \in S_1\) and \(j \in f(S_1, i)\), i.e., \(x_i = y_{ij} = 1\), \(\tilde{x}_{ij} = x_i y_{ij}\) can be viewed as an indicator variable of the availability of \((i, j)\). Therefore, \(P_{ij}\) can be interpreted as the customer’s probability of choosing path \((i, j)\) among all available paths, which follows an MNL model with parameter \(\tilde{\mu}_{ij}\).

Furthermore, we define \(\tilde{p}_{ij} = p_i^{(1)} + p_j^{(2)}\) to be the total revenue of a path \((i, j)\). Then we can rewrite problem (2) as the following optimization problem:

\[
\max_{\tilde{x}} \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} e^{\tilde{\mu}_{ij} \tilde{x}_{ij}} \tilde{p}_{ij}
\]

\[
\text{s.t.} \quad \tilde{x}_{00} = 1
\]
\[
\tilde{x}_{ij} \leq \tilde{x}_{i0}, \quad \forall i = 0, \ldots, n_1, \quad j = 0, \ldots, n_2
\]
\[
\tilde{x}_{ij} \in \{0, 1\}, \quad \forall i = 0, \ldots, n_1, \quad j = 0, \ldots, n_2.
\]

Note that if we remove the constraint \(\tilde{x}_{ij} \leq \tilde{x}_{i0}\) from (5), then (5) is equivalent to a single-stage assortment optimization problem with \(n_1 n_2 + n_1 + n_2\) products (each path \((i, j)\) except \((0, 0)\) is a product with associated price \(\tilde{p}_{ij}\) and utility \(\tilde{\mu}_{ij}\)). However, there is an additional set of constraints \(\tilde{x}_{ij} \leq \tilde{x}_{i0}\), which arises from that the customer can always choose the “no purchase” option in the second stage. This difference makes the problem interesting yet more challenging.

First, we claim that despite the added complexity due to the constraint \(\tilde{x}_{ij} \leq \tilde{x}_{i0}\), problem (5) can still be solved in polynomial time. In order to solve (5), one could adopt the method introduced in Davis et al. (2013) (see also Sumida et al. 2020 for an updated version), in which the authors show that for any assortment optimization problem under the standard MNL model, if there is a set of constraints on the offered products that is totally unimodular, then the assortment optimization problem can be solved via a linear program.
Although the above method can solve the problem efficiently using linear optimization, because of the added constraint, the revenue-ordered property may no longer hold. That is, there may not be an optimal assortment that is revenue-ordered according to the values $\tilde{p}_{ij}$. We show such an example in Appendix B.1. Despite this, we show in the following that problem (5) has very nice structural properties which can provide important guidance in selecting optimal assortments in practice. We also show that using a novel merging idea, we can solve the two-stage assortment problem in $O(n_1 n_2 \log(n_1 n_2))$, which significantly improves the complexity of solving a linear program.

To start with, we show some important properties about the optimal solution to (5). We have the following proposition (the proof is given in Appendix C):

**Proposition 4.2.** For problem (5), there exists an optimal solution $\tilde{x}^*$ such that

1. For any $i \in \mathcal{N}_1 \cup \{0\}$ and $j \in \mathcal{N}_2 \cup \{0\}$, if $\tilde{x}^*_{ij} = 1$, then for any $0 \leq k \leq j$, $\tilde{x}^*_{ik} = 1$;
2. For any $i \in \mathcal{N}_1 \cup \{0\}$ and $j \in \mathcal{N}_2 \cup \{0\}$, if $\tilde{x}^*_{ij} = 1$, then for any $1 \leq k \leq i$, $\tilde{x}^*_{kj} = 1$.

To interpret the results in Proposition 4.2, we note that part 1 says that for each option $i$ that the customer may choose in the first stage, the subsequently offered assortment in the second stage is revenue-ordered, which means, all second-stage assortments must be revenue-ordered. For part 2, we first take $j = 0$. In this case, $\tilde{x}^*_{i0} = 1$ means that the seller offers product $i$ in the first stage, then the implied $\tilde{x}^*_{k0} = 1$ for all $1 \leq k \leq i$ means that the seller must also offer all the products that have higher revenue in the first stage, which means, the first-stage assortment must be revenue-ordered. Furthermore, part 2 also means that the second-stage assortment must be nested — a higher revenue product in the first stage must lead to a larger offered set in the second stage. To summarize the results in Proposition 4.2, we have the following theorem:

**Theorem 4.2.** When $\gamma = 1$, there exists an optimal solution to the two-stage assortment optimization problem with first-stage assortment being $S^*_1$ and second-stage assortments being $f^*(S^*_1, i)$, such that

1. (Revenue-Ordered in First Stage) $S^*_1$ is revenue-ordered;
2. (Revenue-Ordered in Second Stage) $f^*(S^*_1, i)$ is revenue-ordered for each $i \in S^*_1 \cup \{0\}$;
3. (Nested in Second Stage) $f^*(S^*_1, 1) \supseteq f^*(S^*_1, 2) \supseteq \cdots \supseteq f^*(S^*_1, |S^*_1|)$.

Theorem 4.2 has strong practical implications. First, it says that despite the multi-stage nature of the problem, the optimal offering in each stage must still be revenue-ordered. This means that the seller should always offer a continuous set of products from the top in each stage. Second, it also says that the offering in second stage must be nested, meaning if customer purchases a higher-valued first stage products, then in the next stage, the seller should offer this customer a larger choice set. In some sense, this decision is to encourage the customer to purchase the more
expensive product in the first stage, so she will be able to access a larger set of options later (although the value of being access to more products in the second stage is often unclear at the time of making the first-stage purchase decision). Note that this phenomenon is prevalent in practice. For example, in an airline context, it is often that a customer who bought more expensive tickets would have more options regarding upgrades, changing flights, standby earlier flights, etc. In the case of United Airlines, only customers who bought fare class W or above can request upgrade using a global upgrade certificate. Similarly, for changing flights, customer with higher fare class will have priority to access flight alternatives, and certain low fare classes (e.g., basic economy) may not be allowed to change flights, even for its highest premier members. Note that upgrading using a global upgrade certificate or changing flights may still need a fee so it should be viewed as a second-stage purchase rather than included as a benefit in the first-stage purchase. See, e.g., United Airlines (2018) for details of its policies.

Note that even though Proposition 4.2 and Theorem 4.2 provide important intuition about the optimal assortment, they themselves do not directly lead to a polynomial-time algorithm for (5) since the number of possible assortments that satisfy Theorem 4.2 can still be exponentially many. In the following, we exploit some further structures of problem (5), and show that we can improve the computational complexity of this problem to $O(n_1 n_2 \log(n_1 n_2))$. We start from the following proposition (the proof is given in Appendix D).

PROPOSITION 4.3. Let $m_1$ be the largest value such that $\{1, \ldots, m_1\}$ is an optimal assortment for products $\mathcal{N}_1$ under the standard MNL model. Let $m_2$ be the smallest value such that $\{1, \ldots, m_2\}$ is an optimal assortment for products $\mathcal{N}_2$ under the standard MNL model. When $\gamma = 1$, there exists an optimal solution $S_1^*$ and $f^*$ to the two-stage assortment optimization problem such that

$$S_1^* \subseteq \{1, 2, \ldots, m_1\}, \quad f^*(S_1^*, 1) \supseteq \cdots \supseteq f^*(S_1^*, |S_1^*|) \supseteq \{1, 2, \ldots, m_2\}.$$ 

Proposition 4.3 shows that, under the optimal policy, products that do not belong to $\{1, 2, \ldots, m_1\}$ (i.e., the optimal single-stage assortment for stage 1) should never be provided in stage 1. Moreover, for those products that are provided in stage 1, products $\{1, 2, \ldots, m_2\}$ (i.e., the optimal single-stage assortment for stage 2) should always be provided in the corresponding stage 2 offering. Therefore, without computing the optimal policy, we are able to exclude $\{m_1 + 1, m_1 + 2, \ldots, n_1\}$ from $S_1^*$ and include $\{1, 2, \ldots, m_2\}$ in $S_2^*$ in advance. In other words, compared with the optimal assortments for stage 1 and stage 2, the optimal policy always offers fewer products in the first stage, and offers more products in the second stage. To intuitively understand this, in order to encourage customers to buy more expensive products in the first stage, the firms should use “more options in the second stage” as an incentive, thus should offer more products in
the second stage than otherwise it is optimal if there were only the second stage. In the meantime, because the firm adds more value to the higher value product in the first stage (by offering more second-stage products), the “value differences” among products in the first stage will be larger, which will make the firm offering fewer products in the first stage.

Next we use the property in Proposition 4.3 to derive an efficient algorithm for the two-stage assortment optimization problem. We first start with a corollary of Proposition 4.3.

COROLLARY 4.1. For problem (5), there exists an optimal solution \( \tilde{x}^* \) such that for all \( i \neq 0 \), either
\[ \tilde{x}^*_{i0} = \tilde{x}^*_{i1} = \cdots = \tilde{x}^*_{im_2} = 1, \]
or
\[ \tilde{x}^*_{i0} = \tilde{x}^*_{i1} = \cdots = \tilde{x}^*_{im_2} = 0. \]

Corollary 4.1 suggests that, in problem (5), for all \( i \neq 0 \), paths \((i, 0), (i, 1), \ldots, (i, m_2)\) either appear in the optimal solution altogether, or are absent from it altogether. Based on this insight, we can merge them into a single new path when we compute the optimal assortment. Specifically, for all \( i \in N_1 \), we define
\[ \tilde{\mu}_{i\sim} := \log \left( \sum_{j=0}^{m_2} e^{\tilde{\mu}_{ij}} \right) \quad \text{and} \quad \tilde{p}_{i\sim} := \frac{\sum_{j=0}^{m_2} e^{\tilde{\mu}_{ij}} \tilde{p}_{ij}}{\sum_{j=0}^{m_2} e^{\tilde{\mu}_{ij}}}, \]
and replace paths \((i, 0), (i, 1), \ldots, (i, m_2)\) with a single new path \((i, \sim)\) whose price is \( \tilde{p}_{i\sim} \) and mean utility is \( \tilde{\mu}_{i\sim} \). After this merging procedure, we obtain a new set of product-like paths
\[ \tilde{N} := \{(0, j) \mid j \in N_2\} \cup \{(i, \sim) \mid i \in N_1\} \cup \{(i, j) \mid i \in N_1, m_2 + 1 \leq j \leq n_2\}. \]

Since \( \{1, \ldots, m_2\} \) is the optimal single-stage assortment for the second stage, it must be that \( \tilde{p}_{i\sim} \geq \tilde{p}_{i(m_2+1)} \),\(^7\) which means that the merged paths still have higher prices. Therefore, if we find an optimal assortment under an MNL model from \( \tilde{N} \), then a path \((i, j) \) \( (i \in N_1, m_2 + 1 \leq j \leq n_2) \) is included in the optimal assortment only when path \((i, \sim)\) is included. As a result, the constraint \( \tilde{x}_{ij} \leq \tilde{x}_{i0} \) in (5) is naturally satisfied in our new problem. That is, we can solve (5) equivalently by solving the following problem:
\[ \max_{S \subseteq \tilde{N}} \sum_{(i,j) \in S} e^{\tilde{\mu}_{ij}} \left( \frac{\tilde{p}_{ij}}{1 + \sum_{(i,j) \in S} e^{\tilde{\mu}_{ij}} \tilde{p}_{ij}} \right), \]
which is exactly a standard assortment optimization problem under the MNL model with \( O(n_1 n_2) \) products. With the help of the revenue-ordered property of single-stage assortment problems, we can solve problem (5) in \( O(n_1 n_2 \log(n_1 n_2)) \) time (the main complexity is attributed to sorting the corresponding revenues for the paths in \( \tilde{N} \)). Therefore, by the above discussion, we can also find an optimal solution to (5) in \( O(n_1 n_2 \log(n_1 n_2)) \) time. We formalize the above procedure in Algorithm 1, and obtain Proposition 4.4.

\(^7\) For an assortment optimization under the MNL model, it is known that \( \tilde{p}_{i\sim} \) is equal to the optimal revenue. In addition, it is well known that the optimal assortment includes all products with price higher than the optimal revenue and does not include any product with price lower than the optimal revenue, see, e.g., Talluri and van Ryzin (2004).
In this section, we consider the case when the customer is partially forward-looking, i.e., $0 < \gamma < 4.3$. Partially Forward-Looking Customer ($i,j$ as an indicator variable of the availability of $\gamma$)

Similar to Section 4.2, we view the problem from the perspective of determining an assortment of optimal solution to (2) in $O(n_1 n_2 \log(n_1 n_2))$ time.

### Algorithm 1 Merged Path Selection (MPS)

**Input:** $\mu^{(1)}_i, p^{(1)}_i, i = 0, \ldots, n_1$ and $\mu^{(2)}_j, p^{(2)}_j, j = 0, \ldots, n_2$.

1. Compute the “smallest” optimal single-stage assortment for stage 2: $\{1, 2, \ldots, m_2\}$.
2. for $i = 0, 1, \ldots, n_1$ do
3.   for $j = 0, 1, \ldots, n_2$ do
4.     Compute $\widetilde{\mu}_{ij} = \mu^{(1)}_i + \mu^{(2)}_j$ and $\widetilde{p}_{ij} = p^{(1)}_i + p^{(2)}_j$.
5.   if $i \neq 0$ then
6.     Merge paths $(i, 0), (i, 1), \ldots, (i, m_2)$ into a new path $(i, \sim)$ with $\widetilde{\mu}_{i,\sim}$ and $\widetilde{p}_{i,\sim}$ defined in (6).
7. Solve problem (7) to find an optimal assortment $\tilde{S}^*$.
8. (For problem (5)) Recover the corresponding $\tilde{x}^*$ from $\tilde{S}^*$.
9. (For problem (2)) Recover the corresponding $(x^*, y^*)$ from $\tilde{x}^*$.
10. return $(x^*, y^*)$

**Proposition 4.4.** When $\gamma = 1$, the Merged Path Selection algorithm (Algorithm 1) can find an optimal solution to (2) in $O(n_1 n_2 \log(n_1 n_2))$ time.

### 4.3. Partially Forward-Looking Customer ($0 < \gamma < 1$)

In this section, we consider the case when the customer is partially forward-looking, i.e., $0 < \gamma < 1$. Similar to Section 4.2, we view the problem from the perspective of determining an assortment of paths. Similar to the previous section, we let $(i, j)$ denote the customer’s action of “first choosing $i$ then choosing $j$”, and refer to it as her two-stage choice path. In this case, let $\breve{\mu}_{ij} = \mu^{(1)}_i / \gamma + \mu^{(2)}_j$ be the total expected utility of $(i, j)$, and $\breve{p}_{ij} = p^{(1)}_i + p^{(2)}_j$ be the total revenue of $(i, j)$. (Note that this time there is a discount factor $\gamma$ appearing in the definition of $\breve{\mu}_{ij}$.) Also, $\breve{x}_{ij} = x_i y_{ij}$ can be viewed as an indicator variable of the availability of $(i, j)$, and $P_{ij}$ can be interpreted as the customer’s probability of choosing path $(i, j)$ among all available paths.

Using the above definitions, we can rewrite $P_{ij}$ as

$$P_{ij} = \frac{e^{\mu^{(1)}_i} \left( \sum_{j=0}^{n_2} e^{\mu^{(2)}_j} y_{ij} \right)^{\gamma} x_i}{\sum_{i=0}^{n_1} e^{\mu^{(1)}_i} \left( \sum_{j=0}^{n_2} e^{\mu^{(2)}_j} y_{ij} \right)^{\gamma} x_i} \cdot \frac{e^{\mu^{(2)}_j} y_{ij}}{\sum_{j=0}^{n_2} e^{\mu^{(2)}_j} y_{ij}} = \frac{\left( \sum_{j=0}^{n_2} e^{\breve{\mu}_{ij}} \breve{x}_{ij} \right)^{\gamma}}{\sum_{i=0}^{n_1} \left( \sum_{j=0}^{n_2} e^{\breve{\mu}_{ij}} \breve{x}_{ij} \right)^{\gamma}} \cdot \frac{e^{\breve{\mu}_{ij}} \breve{x}_{ij}}{\sum_{j=0}^{n_2} e^{\breve{\mu}_{ij}} \breve{x}_{ij}}.$$  

Then we can reformulate problem (2) as:

$$\max_{\tilde{x}} \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} \frac{\left( \sum_{j=0}^{n_2} e^{\breve{\mu}_{ij}} \breve{x}_{ij} \right)^{\gamma-1} e^{\breve{\mu}_{ij}} \breve{x}_{ij}}{\sum_{i=0}^{n_1} \left( \sum_{j=0}^{n_2} e^{\breve{\mu}_{ij}} \breve{x}_{ij} \right)^{\gamma}} - \tilde{p}_{ij}$$

s.t. \( \tilde{x}_{00} = 1 \)

\( \tilde{x}_{ij} \leq \tilde{x}_{i0}, \quad \forall i = 0, \ldots, n_1, \quad j = 0, \ldots, n_2 \)
\[ \tilde{x}_{ij} \in \{0,1\}, \quad \forall i = 0, \ldots, n_1, \ j = 0, \ldots, n_2. \]

Note that problem (9) looks similar to an assortment optimization problem under a two-level nested logit model with \((n_1 + 1)\) nests and \((n_1 n_2 + n_1 + n_2)\) products. However, compared with the standard nested logit assortment optimization problem, there are two main differences in problem (9): first, there exists a “nest 0” that contains \(n_2\) paths; second, there exists a no-purchase option in every nest, and there is an additional set of constraints \(\tilde{x}_{ij} \leq \tilde{x}_{i0}\). In the following, we will show that these two features make the problem more complicated. We also remark that the meaning of \(\gamma\) is different in the two choice models. In our model, the \(\gamma\) indicates the time-discount factor of the customer, while in the nested logit model, the \(\gamma\) is the dissimilarity factor capturing the similarity among different nests.

To start with, we show some important properties about the optimal solution to (9). We have the following proposition:

**Proposition 4.5.** For problem (9), there exists an optimal solution \(\tilde{x}^*\) such that

1. For any \(i \in N_1\) and \(j \in N_2 \cup \{0\}\), if \(\tilde{x}_{ij}^* = 1\), then for any \(0 \leq k \leq j\), \(\tilde{x}_{ik}^* = 1\);
2. For any \(i \in N_1\) and \(j \in N_2 \cup \{0\}\), if \(\tilde{x}_{ij}^* = 1\), then for any \(1 \leq k \leq i\), \(\tilde{x}_{kj}^* = 1\).

It is worth noting that Proposition 4.5 actually holds true for \(0 \leq \gamma \leq 1\) (though in this subsection we are discussing the case of \(0 < \gamma < 1\)). Thus, Proposition 4.5 can be viewed as a generalization of Proposition 4.2. While the proof of Proposition 4.5 is more complicated than Proposition 4.2, observing the close relationship between Proposition 4.5 and Proposition 4.2, we put them together and prove them simultaneously in Appendix C.

In the following, we interpret the results in Proposition 4.5, and point out their differences with our previous results for \(\gamma = 1\) (see Proposition 4.2 and Theorem 4.2). We first look at part 1. Part 1 says that for each product \(i\) \((i \neq 0)\) that is offered in the first stage, the subsequently offered assortment in the second stage \(f^*(S^*_1, i)\) is revenue-ordered. Note that compared with Proposition 4.2, here our result requires \(i \neq 0\), which means that the customer does not choose the no-purchase option in the first stage. Indeed, we can find that when \(0 < \gamma < 1\), if the customer does not buy any product in the first stage, i.e., the requirement in the proposition is violated, then in the second stage, the revenue-ordered property may not hold for \(f^*(S^*_1, 0)\).\(^8\) To better illustrate this fact, we give a numerical example and provide insights for it in Appendix B.2. This is different from the case of \(\gamma = 1\), in which we know that the seller can always offer revenue-ordered assortments in the

\(^8\) Note that \(f^*(S^*_1, 0)\) is not necessarily empty: in many contexts, even if a customer buys nothing in the first stage, she will still have purchase options in future stages. For example, a sports fan without a regular season ticket membership can still purchase playoffs tickets, though she may have fewer options and have to face higher prices.
second stage in an optimal solution, no matter what the customer chooses in the first stage (see Theorem 4.2).

For part 2, we first take \( j = 0 \). In this case, we can derive the fact that the first-stage assortment must be revenue-ordered. Furthermore, part 2 also means that the second-stage assortment must be nested — a higher revenue product in the first stage must lead to a larger offered set in the second stage. Thus, this part is the same as part 2 in Proposition 4.2.

To summarize the results in Proposition 4.5, we have the following theorem:

**Theorem 4.3.** When \( 0 < \gamma < 1 \), there exists an optimal solution to the two-stage assortment optimization problem with first-stage assortment being \( S^*_1 \) and second-stage assortments being \( f^*(S^*_1, i) \), such that

1. (Revenue-Ordered in First Stage) \( S^*_1 \) is revenue-ordered;
2. (Revenue-Ordered in Second Stage) \( f^*(S^*_1, i) \) is revenue-ordered for each \( i \in S^*_1 \);
3. (Nested in Second Stage) \( f^*(S^*_1, 1) \supseteq f^*(S^*_1, 2) \supseteq \cdots \supseteq f^*(S^*_1, |S^*_1|) \).

Compared with Theorem 4.2, there is only one difference in the statement of Theorem 4.3: for part 2, Theorem 4.2 says that \( f^*(S^*_1, i) \) is revenue-ordered for all \( i \in S^*_1 \cup \{0\} \), while Theorem 4.3 just applies to \( i \in S^*_1 \). That is, \( f^*(S^*_1, 0) \) no longer has the revenue-ordered structure. Later we will show that this difference brings significant challenge for solving this problem.

In the following, we exploit some further structures of problem (9). We start from the following proposition.

**Proposition 4.6.** Let \( m_1 \) be the largest value such that \( \{1, \ldots, m_1\} \) is an optimal assortment for products \( N_1 \) under the standard MNL model. Let \( m_2 \) be the smallest value such that \( \{1, \ldots, m_2\} \) is an optimal assortment for products \( N_2 \) under the standard MNL model. When \( 0 \leq \gamma \leq 1 \), there exists an optimal solution \( S^*_1 \) and \( f^* \) to the two-stage assortment optimization problem such that

\[ S^*_1 \subseteq \{1, 2, \ldots, m_1\}, \quad f^*(S^*_1, 1) \supseteq \cdots \supseteq f^*(S^*_1, |S^*_1|) \supseteq \{1, 2, \ldots, m_2\}. \]

Since Proposition 4.6 holds true for \( 0 \leq \gamma \leq 1 \), it is actually a strict generalization of Proposition 4.3 (which states the same fact for \( \gamma = 1 \)). Therefore, we prove Proposition 4.6 directly and thus prove Proposition 4.3 in Appendix D.

Similar to Proposition 4.3, Proposition 4.6 shows that, under the optimal policy, the firm should offer fewer products in the first stage than otherwise it is optimal if there were no second stage; and the firm should offer more products in the second stage than otherwise it is optimal if there were no first stage. The intuition is the same as before: by offering more products in the second stage, the firm can encourage the customer to buy more expensive products in the first stage; and
because of that, the “value differences” among products in the first stage is larger, which makes the firm to offer fewer products in the first stage.

Next we use the property in Proposition 4.6 to characterize the optimal solution to the two-stage assortment optimization problem. We first start with a corollary of Proposition 4.6.

**Corollary 4.2.** For problem (9), there exists an optimal solution \( \tilde{x}^* \) such that for all \( i \neq 0 \), either \( \tilde{x}^*_{i0} = \tilde{x}^*_{i1} = \cdots = \tilde{x}^*_{im_2} = 1 \), or \( \tilde{x}^*_{i0} = \tilde{x}^*_{i1} = \cdots = \tilde{x}^*_{im_2} = 0 \).

Corollary 4.2 suggests that, in problem (9), for each \( i \neq 0 \), paths \((i,0),(i,1),\ldots,(i,m_2)\) either appear in \( \tilde{S}^* \) altogether, or are absent from \( \tilde{S}^* \) altogether. Therefore, they can be merged into a single “new” path when we compute the optimal assortment. Specifically, for all \( i \in \mathcal{N}_1 \), we define

\[
\tilde{\mu}_{i\sim} := \log \left( \sum_{i=0}^{m_2} e^{\tilde{\mu}_{ij}} \right), \quad \tilde{p}_{i\sim} := \frac{\sum_{i=0}^{m_2} e^{\tilde{\mu}_{ij} \tilde{p}_{ij}}}{\sum_{i=0}^{m_2} e^{\tilde{\mu}_{ij}}} \tag{10}
\]

and replace paths \((i,0),(i,1),\ldots,(i,m_2)\) with a single new path \((i,\sim)\) whose price is \( \tilde{p}_{i\sim} \) and mean utility is \( \tilde{\mu}_{i\sim} \). After the merging procedure, we obtain \( n_1 \) new nests (of paths):

\[
\operatorname{Nest}_i = \{(i,\sim),(i,m_2+1),(i,m_2+2),\ldots,(i,n_2)\}, \quad i = 1, \ldots, n_1.
\]

Since \( \{1,2,\ldots,m_2\} \) is the optimal single-stage assortment for stage 2, it can be showed that \( \tilde{p}_{i\sim} \geq \tilde{p}_{i(m_2+1)} \) (see footnote 6 for a discussion), which means that the merged path in \( \operatorname{Nest}_i \) still has a higher price than the other paths in \( \operatorname{Nest}_i \) \((i \in \mathcal{N}_1)\).

Consider the new assortment problem after merging. Let the seller’s assortment (of paths) be

\[
X_0 \cup \bigcup_{i=1}^{n_1} X_i,
\]

where \( X_0 \subseteq \{(0,1),\ldots,(0,n_1)\} \) and \( X_i \subseteq \operatorname{Nest}_i \) \((i \in \mathcal{N}_1)\). It can be verified that problem (9) is equivalent to:

\[
\max_{X_0,X_1,\ldots,X_{n_1}} \left( 1 + \sum_{(0,j) \in X_0} e^{\tilde{\mu}_{0j}} \right)^{\gamma-1} \sum_{(0,j) \in X_0} e^{\tilde{\mu}_{0j} \tilde{p}_{0j}} + \sum_{i=1}^{n_1} \left( 1 + \sum_{(i,j) \in X_i} e^{\tilde{\mu}_{ij}} \right)^{\gamma-1} \sum_{(i,j) \in X_i} e^{\tilde{\mu}_{ij} \tilde{p}_{ij}} \right)^{-\gamma}, \tag{11}
\]

s.t.

\[
X_0 \subseteq \{(0,1),\ldots,(0,n_1)\},
\]

\[
X_i \subseteq \operatorname{Nest}_i, \quad \forall i \in \mathcal{N}_1.
\]

In the following, we first fix \( X_0 \) and consider the remaining optimization problem. Define

\[
V_0 = \log \left( 1 + \sum_{(0,j) \in X_0} e^{\tilde{\mu}_{0j}} \right), \quad r_0 = \frac{\sum_{(0,j) \in X_0} e^{\tilde{\mu}_{0j} \tilde{p}_{0j}}}{1 + \sum_{(0,j) \in X_0} e^{\tilde{\mu}_{0j}}}.
\]
When $X_0$ is fixed, $r_0$ and $V_0$ are constants. By simple algebra, (11) is equivalent to

$$\max_{X_1, \ldots, X_n} \left( r_0 + \sum_{i=1}^{n_1} \sum_{(i,j) \in X_i} e^{\tilde{\mu}_{ij}} \right)^{\gamma^{-1}} e^{\hat{\mu}_{ij}} \gamma (\tilde{p}_{ij} - r_0),$$

which is equivalent to an assortment optimization problem under a two-level nested logit model with dissimilarity factor less than 1 and there is no no-purchase option in the second stage. According to existing literature in assortment optimization under nested logit models, problem (12) can be solved by the greedy algorithm of Li and Rusmevichientong (2014) in $O(n_1 n_2 \log n_2)$ time. Thus we have the following proposition.

**Proposition 4.7.** When $0 < \gamma < 1$, if $f(S_1, 0)$ is fixed, then we can find an optimal solution to problem (2) in $O(n_1 n_2 \log n_2)$ time (the main complexity is attributed to sorting the $O(n_1 n_2)$ paths). Let $A_0 \subseteq 2^N_2$ be a given set of assortments. If $f(S_1, 0)$ is not fixed but restricted to be an assortment in $A_0$ (i.e., $f(S_1, 0) \in A_0$), then the running time becomes $O(n_1 n_2 (\log n_1 + |A_0| \log n_2))$ — this is achieved by enumerating over $A_0$ and calculating the solution for each choice of $f(S_1, 0)$.

As a concrete example, for the airline ancillary example described in Section 1, since $f(S_1, 0)$ must be $\emptyset$ (i.e., a customer who did not purchase an airline ticket in the first stage cannot purchase any ancillary in the second stage), by Proposition 4.7, we can find the optimal assortment policy in polynomial time.

Nevertheless, in general, $f^*(S^*_1, 0)$ can be any subset of $N_2$ (recall that the revenue-ordered property does not necessarily hold for $f^*(S^*_1, 0)$ in Theorem 4.3), and $A_0$ can be the power set of $N_2$. As a result, $|A_0|$ can be exponential in $n_2$, and Proposition 4.7 cannot imply the polynomial-time solvability of problem (2). In fact, such a limitation is fundamental — as we show in the next proposition, the two-stage assortment optimization problem is an NP-hard problem when $0 < \gamma < 1$, meaning that there is no hope to find a polynomial-time algorithm unless P=NP.

**Proposition 4.8.** For any fixed rational $\gamma \in (0, 1)$, problem (2) is NP-hard, even when $n_1 = 1$.

The proof of Proposition 4.8 is given in Appendix E. Since problem (2) is NP-hard even when $n_1 = 1$, we can conclude that it is $n_2$ rather than $n_1$ that makes problem (2) hard when $0 < \gamma < 1$.

Knowing that we cannot efficiently find the exact optimal solution to problem (2) when $0 < \gamma < 1$, a natural and important question is whether we can always find a good approximate solution in polynomial time. We answer this question in the affirmative by designing a fast and simple algorithm, called the Nested Merged Path Selection (NMPS) algorithm, whose expected revenue
is always at least half as large as the optimal one; see Algorithm 2 for details. Note that in step 10 of Algorithm 2, we need to solve an assortment optimization problem under a nested logit model, which can be done in $O(n_1 n_2 \log n_2)$ time (Li and Rusmevichientong 2014). The performance guarantee of Algorithm 2 is stated in Theorem 4.4 and proved in Appendix F.

**Algorithm 2** Nested Merged Path Selection (NMPS)

**Input:** $\mu^{(1)}_i, p^{(1)}_i, i = 0, \ldots, n_1$ and $\mu^{(2)}_j, p^{(2)}_j, j = 0, \ldots, n_2$.

1: Compute the “smallest” optimal single-stage assortment for stage 2: $\{1, 2, \ldots, m_2\}$.

2: for $i = 0, 1, \ldots, n_1$ do

3: for $j = 0, 1, \ldots, n_2$ do

4: Compute $\tilde{\mu}_{ij} = \mu^{(1)}_i + \mu^{(2)}_{ij}/\gamma$ and $\tilde{p}_{ij} = p^{(1)}_i + p^{(2)}_{ij}$.

5: if $i \neq 0$ then

6: Merge paths $(i, 0), (i, 1), \ldots, (i, m_2)$ into a new path $(i, \sim)$ with $\tilde{\mu}_{i, \sim}$ and $\tilde{p}_{i, \sim}$ defined in (10).

7: Let $\tilde{S}^*$ be $\emptyset$.

8: for $S \in \{\emptyset, \{1, \ldots, m_2\}\}$ do

9: Fix $f(S_1, 0) = S$ and formulate problem (12).

10: Solve problem (12) to find an optimal assortment and construct a solution to problem (11).

11: if the new solution is better than $\tilde{S}^*$ then

12: Change $\tilde{S}^*$ to the new solution.

13: (For problem (9)) Recover the corresponding $\tilde{x}^*$ from $\tilde{S}^*$.

14: (For problem (2)) Recover the corresponding $(x^*, y^*)$ from $\tilde{x}^*$.

15: return $(x^*, y^*)$

**Theorem 4.4.** Let $R^*$ denote the optimal expected revenue of the two-stage assortment optimization problem (i.e., the optimal objective value of problem (2)). When $0 < \gamma < 1$, the expected revenue achieved by Algorithm 2 (whose running time is $O(n_1 n_2 (\log n_1 n_2))$) is at least $R^*/2$.

Before ending this section, we provide intuition on the design of Algorithm 2. The hardness of the case of $0 < \gamma < 1$ essentially originates from the following trade-off in determining $f(S_1, 0)$: setting $f(S_1, 0)$ to be smaller reduces the second-stage revenue if the customer chooses 0 in the first stage, whereas setting $f(S_1, 0)$ to be larger increases the probability that the customer chooses 0 in the first stage, which is costly because the customer buys less in the first stage. Intuitively, the set $\{\emptyset, \{1, \ldots, m_2\}\}$ in step 8 of Algorithm 2 contains two “extreme” choices of $f(S_1, 0)$: setting $f(S_1, 0) = \emptyset$ minimizes the probability that the customer chooses 0 in the first stage, whereas setting $f(S_1, 0) = \{1, \ldots, m_2\}$ maximizes the second-stage revenue if the customer chooses 0 in the first stage.
stage. At a high level, Algorithm 2 proceeds by first optimizing all other assortments conditional on \( f(S_1,0) \) (via solving (12)) and then optimizing over only two extreme choices of \( f(S_1,0) \). Such a procedure is directly motivated by Proposition 4.7 — note that Algorithm 2 essentially replaces \( A_0 \) in Proposition 4.7 by \( \{\emptyset, \{1,\ldots,m_2\} \} \), which (though not necessarily containing \( f^*(S_1^*,0) \)) is sufficient for our approximation purpose.

5. Performance of Static Policies

One main feature of the policies that we consider in the previous section is that they are adaptive. That is, the assortments offered in each stage can depend on the purchase decision of the customer in previous stages. While the adaptive feature gives the seller more flexibility, it also makes the multi-stage assortment optimization problem quite challenging. Thus, some questions naturally arise: *Is considering adaptive policies necessary? Can we use simpler non-adaptive policies to achieve similar performance?*

To answer the above questions, in this section, we study the performance of non-adaptive, i.e., static policies. The precise definition of static policies is given as below:

**Definition 2** A policy \( \Pi = \{f_1, f_2, \ldots, f_T\} \) is said to be static if \( f_1, f_2, \ldots, f_T \) are all constant mappings. In other words, under a static policy, the seller’s assortments \( S_1, S_2, \ldots, S_T \) are fixed and independent of the customer’s choices.

Under a static policy, different stages are independent, so a multi-stage problem can be decomposed into multiple single-stage problems, and can be solved separately. Indeed, we have shown that for the multi-stage assortment optimization problem with myopic customer (\( \gamma = 0 \)), a static policy is optimal. In this case, simply optimizing the assortment in each stage leads to the optimal solution (see Section 4.1 for the two-stage case).

However, when the customer is not myopic, i.e., \( \gamma > 0 \), static policies may not necessarily be optimal. In the following, we discuss the performance of static policies in such cases. For the simplicity of discussion, we still focus on the two-stage problem. We use \([S_1, S_2]\) to denote the static policy when the seller offers \( S_1 \) in stage 1 and \( S_2 \) in stage 2.

We first consider the case when \( \gamma = 1 \). Let \( \{1,\ldots,m_1\} \) be an optimal assortment for products \( \mathcal{N}_1 \) under the standard MNL model, with expected revenue \( R_1^* = \left( \sum_{i=1}^{m_1} e^{\mu_i(1)} p_i(1) \right) / \left( 1 + \sum_{i=1}^{m_1} e^{\mu_i(1)} \right) \). Let \( \{1,\ldots,m_2\} \) be an optimal assortment for products \( \mathcal{N}_2 \) under the standard MNL model, with expected revenue \( R_2^* = \left( \sum_{i=1}^{m_2} e^{\mu_i(2)} p_i(2) \right) / \left( 1 + \sum_{i=1}^{m_2} e^{\mu_i(2)} \right) \). Obviously, \([\{1,\ldots,m_1\}, \{1,\ldots,m_2\}]\) is the “optimal static policy”, with expected two-stage revenue being \( R_1^* + R_2^* \). We have the following proposition (the proof is given in Appendix G).
**Proposition 5.1.** When $\gamma = 1$, $[\{1, \ldots, m_1\}, \{1, \ldots, m_2\}]$ is an optimal solution to the two-stage assortment optimization problem if and only if

\[
\begin{align*}
    p^{(2)}_{m_2} &\geq R_1^* + R_2^* \geq p^{(1)}_1 + p^{(2)}_{m_2+1}, & \text{if } m_2 < n_2, \\
    p^{(2)}_{m_2} &\geq R_1^* + R_2^* & \text{if } m_2 = n_2.
\end{align*}
\]  

(13)

Proposition 5.1 gives a necessary and sufficient condition for static policies to be optimal for the two-stage assortment optimization problem. Now we provide some explanations for condition (13). First, (13) requires $p^{(2)}_{m_2} \geq R_1^* + R_2^*$, which means that $R_1^*$ is very small (recall that $p^{(2)}_{m_2}$ is the smallest price larger than $R_2^*$ in stage 2). This further means that either the prices in stage 1 are very small, or the customer’s demand in stage 1 is very low. Moreover, when $m_2 < n_2$, (13) requires $p^{(1)}_1 \leq p^{(2)}_{m_2} - p^{(2)}_{m_2+1}$, which means that the prices in stage 1 are very low (recall that $p^{(1)}_1$ is the largest price in stage 1). When condition (13) is met, the prices in the first stage are so low that there is no incentive for the firm to adjust the second-stage offering (i.e., to deviate from the optimal second-stage offering if there were no first-stage problem) to attract the customer to buy more expensive products in the first stage. Such cases may not occur too frequently in practice. It may only occur when the main spending of the customer occurs in the second stage and the first stage is only a small membership fee or an entry fee. However, even in those cases, the seller may not want to charge the first-stage fee too small because it will not enable different offering in second-stage, which will not serve the original purpose of the first stage choices (e.g., if there are multiple membership levels, the fee should be large enough so that it will reflect the difference in value of the second stage offering, otherwise it does not make much sense).

Next we consider the case when $0 < \gamma < 1$. We have the following result (the proof is given in Appendix G).

**Proposition 5.2.** When $0 < \gamma < 1$, if $[\{1, \ldots, m_1\}, \{1, \ldots, m_2\}]$ is an optimal solution to the two-stage assortment optimization problem and $m_2 < n_2$, then

\[
\gamma p^{(1)}_{m_1} + p^{(2)}_{m_2} \geq \gamma R_1^* + R_2^* \geq \gamma p^{(1)}_1 + p^{(2)}_{m_2+1}.
\]  

(14)

Proposition 5.2 gives a necessary condition for static policies to be optimal. While condition (14) is not a sufficient condition thus is not as strong as condition (13), it conveys similar insights: $p^{(1)}_1$ has to be small in order for a static policy to be optimal. Moreover, from (14), we can see that as $\gamma$ approaches to 0, the necessary condition (14) is easier to satisfy, which is consistent with the fact that when $\gamma = 0$ static policies are optimal.

Understanding that a static policy may not be optimal when $\gamma > 0$, we turn to understand whether static policies can at least provide a good approximation of the optimal policy when $\gamma > 0$. Unfortunately, the answer is “No”. As Proposition 5.3 shows, for any $\gamma > 0$, static policies do not provide a constant-factor approximation for the two-stage assortment optimization problem.
Proposition 5.3. Given $\gamma > 0$, for any $\epsilon > 0$, there exists an instance of the two-stage assortment optimization problem with optimal revenue $R^*$, such that no static policy can obtain an expected revenue larger than $\epsilon R^*$.

We prove Proposition 5.3 by providing an instance with parameter $z$ where static policies only obtain constant expected revenue while the optimal expected revenue is $\Omega(z)$. As $z$ is arbitrary, this means there is an unbounded performance gap between the optimal static policy and the optimal policy. Another interesting interpretation of this result is that there is an unbounded gap between the revenue obtainable from myopic customers (equivalent to the performance of static policies in Proposition 5.3) and the revenue obtainable from forward-looking customers. The construction of such an instance and the proof of Proposition 5.3 are given in Appendix B.3.

6. Choice-Dependent Product Sets

In the previous sections, we assume that $\mathcal{N}_t$, i.e., the set of products that the seller can offer in stage $t$, is independent with the seller’s offering history $h^t_s$ and the customer’s choice history $h^t_c$. In other words, no matter what assortment the seller offers and which product the customer chooses in previous stages, in stage $t$, the seller always selects products from the same set $\mathcal{N}_t$ to form his assortment. In such cases, we have found some interesting structures of the optimal assortments.

However, in reality, customer’s purchasing history may affect the set of products that the seller can offer in the future, i.e., $\mathcal{N}_t$ may depend on $h^t_c$. For example, if a customer buys an iPhone, while another customer buys an iPad, then the further accessories and apps that they can buy in the future are different (though there could be some overlaps). In this section, we consider a generalized model that allows such possibilities. Specifically, we allow $\mathcal{N}_t$ to depend on $h^t_c$.

We still focus on the two-stage assortment optimization problem.\footnote{Extension to multi-stage problem with choice-depend product sets can be obtained by combining the arguments in this section with Appendix A.} In stage 1, the set of all products available for sale is $\mathcal{N}_1 = \{1, 2, \ldots, n_1\}$, with prices $p^{(1)}_1, p^{(1)}_2, \ldots, p^{(1)}_{n_1}$ and mean utilities $\mu^{(1)}_1, \mu^{(1)}_2, \ldots, \mu^{(1)}_{n_1}$. In stage 2, let $\mathcal{N}_2(i)$ be the set of all products available for sale given that the customer chooses $i$ in stage 1 ($i \in \mathcal{N}_1 \cup \{0\}$). Without loss of generality, we assume that $|\mathcal{N}_2(i)| = n_2$ for all $i \in \mathcal{N}_1 \cup \{0\}$,\footnote{The assumption that $\mathcal{N}_2(i)$ with different $i$ includes the same number of products is without loss of generality because if some $\mathcal{N}_2$ includes fewer than $n_2$ products, then we can include additional products $j$ in this set with mean utility $-\infty$ and these products would never be purchased.} index the products in $\mathcal{N}_2(i)$ such that $\mathcal{N}_2(i) = \{i(1), i(2), \ldots, i(n_2)\}$, and assume that the products in $\mathcal{N}_2(i)$ are ordered by revenue, i.e., $p^{(2)}_{i(1)} \geq p^{(2)}_{i(2)} \geq \cdots \geq p^{(2)}_{i(n_2)}$ for all $i \in \mathcal{N}_1 \cup \{0\}$, with $p^{(2)}_{i(j)}$ denoting the revenue of product $i(j)$ and $\mu^{(2)}_{i(j)}$ denoting the customer’s mean utility of product $i(j)$. Let $i(0)$ denote the no-purchase option in stage 2 after choosing $i$ in stage 1, with revenue $p^{(2)}_{i(0)} = 0$ and mean utility $\mu^{(2)}_{i(0)}$; to fully capture the heterogeneity of $\mathcal{N}_2(i)$ for
different \(i\), we allow \(\mu_{0(0)}^{(2)}\) to depend on \(i\) for \(i \neq 0\) (i.e., the mean utility of the no-purchase option in stage 2 can depend on the choice in stage 1). Without loss of generality, we normalize \(\mu_{0(0)}^{(2)} = 0\).

It is evident that the above two-stage model is a strict generalization of our previous two-stage model. In particular, if we let \(N_2(0) = N_2(1) = \cdots = N_2(n_1)\) and \(\mu_{0(0)}^{(2)} = \mu_{1(0)}^{(2)} = \cdots = \mu_{n_2(0)}^{(2)}\), then the above model reduces to our previous model.

We refer to the associated two-stage assortment optimization problem under the above model as the \textit{generalized two-stage assortment optimization} problem. In the following, we obtain the results for this problem. All proofs of the results in this section are given in Appendix H.

We start with the case when the customer is fully myopic (which generalizes the problem in Section 4.1).

**Theorem 6.1.** When \(\gamma = 0\), there exists an optimal solution to the generalized two-stage assortment optimization problem with first-stage assortment being \(S_1^*\) and second-stage assortments being \(f^*(S_1^*, i)\), such that

1. (Revenue-Ordered in First Stage) \(S_1^*\) is revenue-ordered;
2. (Revenue-Ordered in Second Stage) \(f^*(S_1^*, i)\) is revenue-ordered for each \(i \in S_1^* \cup \{0\}\).

Furthermore, the generalized two-stage assortment optimization problem can be solved in \(O(n_1 n_2)\).

When the customer is fully myopic, like Section 4.1, the optimal solution is to simply offer the optimal single-stage assortment in each stage. Thus, one can solve the two-stage problem by decomposing it into multiple single-stage problems and considering them separately. Since there is one single-stage problem (with product set \(N_1\)) in stage 1 and \(O(n_1)\) possible single-stage problems (with product sets \(N_2(i)\)) to consider in stage 2, the whole problem can be solved in \(O(n_1 n_2)\).

Next, we consider the case when the customer is fully forward-looking (which generalizes the problem in Section 4.2).

**Theorem 6.2.** When \(\gamma = 1\), there exists an optimal solution to the generalized two-stage assortment optimization problem with first-stage assortment being \(S_1^*\) and second-stage assortments being \(f^*(S_1^*, i)\), such that

1. (Revenue-Ordered in First Stage) \(S_1^*\) is revenue-ordered;
2. (Revenue-Ordered in Second Stage) \(f^*(S_1^*, i)\) is revenue-ordered for each \(i \in S_1^* \cup \{0\}\).

Note that compared with the result in Theorem 4.2, the “Nested in Second Stage” property does not hold any more. This is because when customers choose different products in stage 1, the product set of the firm in the second stage may change, and thus there is not necessarily such a nested structure. However, we can show that a key property similar to Proposition 4.3 still holds for the generalized model, which enables us to find the optimal assortment policy in polynomial
time. We state this key property in Proposition 6.1 — note that the property itself actually holds generally for all $0 \leq \gamma \leq 1$.\footnote{We would like to point out that Proposition 6.1 does not completely recover Proposition 4.3, as the result of $S_1^* \subseteq \{1, 2, \ldots, m_1\}$ may not hold in the generalized model (due to the variability of second-stage product sets). Nevertheless, this does not affect our design of polynomial-time algorithms.}

**Proposition 6.1.** Let $m(i)$ be the smallest value such that $\{i(1), \ldots, i(m(i))\}$ is an optimal assortment for the products $N_2(i)$ under the standard MNL model ($i \in N_1$). For all $0 \leq \gamma \leq 1$, there exists an optimal solution $S_1^*$ and $f^*$ to the generalized two-stage assortment optimization problem such that

$$f^*(S_1^*, i) \supseteq \{i(1), i(2), \ldots, i(m(i))\}, \quad \forall i \in N_1.$$  

Proposition 6.1 states that even when the product sets in the second stage depend on the first-stage choice, there exists an optimal solution such that the firm should offer more products in the second stage than when there were no first stage. Because of this property, we can still merge paths $(i, 0), (i, 1), \ldots, (i, m(i))$ into a new path $(i, \sim(i))$ for each $i \in N_1$, and run a modified version of MPS algorithm (see Appendix H.3) to efficiently solve the problem in $O(n_1 n_2 \log(n_1 n_2))$ time when $\gamma = 1$.

Finally, we consider the case when the customer is partially forward-looking (which generalizes the problem in Section 4.3).

**Theorem 6.3.** When $0 < \gamma < 1$, there exists an optimal solution to the generalized two-stage assortment optimization problem with first-stage assortment being $S_1^*$ and second-stage assortments being $f^*(S_1^*, i)$, such that

1. (Revenue-Ordered in First Stage) $S_1^*$ is revenue-ordered;
2. (Revenue-Ordered in Second Stage) $f^*(S_1^*, i)$ is revenue-ordered for each $i \in S_1^*$;

Again, compared with the previous result in Theorem 4.3, the “Nested in Second Stage” property does not hold any more. Yet based on Proposition 6.1 (which generally holds for $0 \leq \gamma \leq 1$), we can easily extend Proposition 4.7 and Theorem 4.4 to the setting of choice-dependent product sets here. Specifically, when $f(S_1, 0)$ is fixed, we can find an optimal assortment policy in $O(n_1 n_2 \log(n_1 n_2))$ time; when $f(S_1, 0)$ is not fixed (i.e., in general), we can run a modified version of NMPS algorithm (see Appendix H.3) to find an assortment policy whose expected revenue is at least half as large as the optimal one in $O(n_1 n_2 \log(n_1 n_2))$ time.

Finally, following from Proposition 4.8, when $0 < \gamma < 1$, the generalized two-stage assortment optimization problem must be NP-Hard.
6.1. Choice-Dependent Product Constraints

We have fully understood the setting of choice-dependent product sets, which allows very general modeling. Before ending this section, we discuss an additional extension which can be directly reduced to the setting of choice-dependent product sets that we just studied.

In our current model, the seller has full flexibility in designing what to include in the second-stage assortment. The model can be extended to capture the scenarios where some products have to be included if a particular purchase in the first stage has been made. For example, a car with a V6 engine (compared against a V4 engine) is usually expected to come with the option of leather seats; failing to provide such upgrade options may cause customer dissatisfaction. Such constraints of including certain products in the second stage depend on the choice in the first stage, and we call them “choice-dependent product constraints”.

The problem of choice-dependent product constraints can be reduced to a two-stage assortment optimization problem with choice-dependent product sets. Hence existing results in Section 6 directly apply. The reduction is as follows:

- Consider a problem with both choice-dependent product constraints and choice-dependent product sets (we work in the most general setting).
- For any \( i \in \mathcal{N}_1 \) (note that we do not consider choice-dependent constraints after \( i = 0 \) as they are unrealistic), if product \( i(j) \in \mathcal{N}_2(i) \) must be included in \( f(S_1,i) \), then we merge product \( i(j) \) into \( i(0) \) in the second stage and update its price and mean utility to be \( p_{i(0)}^{(2)} \leftarrow \exp(\mu_{i(j)}^{(2)} - \mu_{i(0)}^{(2)}) \exp(\mu_{i(0)}^{(2)} + \exp(\mu_{i(j)}^{(2)})) \) and \( \mu_{i(0)}^{(2)} \leftarrow \log(\exp(\mu_{i(0)}^{(2)} + \exp(\mu_{i(j)}^{(2)})) \exp(\mu_{i(j)}^{(2)})) \). There is no product \( i(j) \) any more.
- We then do a normalization step to make the prices of no-purchase options in the second stage be zero: for all \( i \in \mathcal{N}_1 \), let \( p_{i}^{(1)} \leftarrow p_{i}^{(1)} + p_{i(0)}^{(2)} \) and \( p_{i(j)}^{(2)} = p_{i(j)}^{(2)} - p_{i(0)}^{(2)} \) for all \( i(j) \in \mathcal{N}_2(i) \cup \{i(0)\} \). Note that some prices in the second stage might become negative — this does not affect our analysis because products with negative prices can be directly ignored.
- We then face a problem with only choice-dependent product sets (there is no choice-dependent product constraint anymore).

Remark. While the problem of choice-dependent product constraints can be technically solved via the above reduction, we should not expect clean structural results for this extension, as the transformations that we use in the reduction make the structural results in Section 6 hard to interpret when they are transformed back to the original problem of choice-dependent product constraints. This is somehow unavoidable as one cannot hope for revenue-ordered-type results when some low-price products must be included in certain assortments. Still, all the algorithmic results (including the 2-approximation) hold in this extension.
7. Concluding Remarks
Motivated by various selling scenarios that require previous purchases to unlock future options, we consider the multi-stage assortment optimization problem, where a seller makes sequential assortment decisions with commitment, and a customer makes subsequent choices based on a multi-stage choice model in which she maximizes her expected intertemporal payoffs. We thoroughly study the multi-stage assortment optimization problem. While we focus on the two-stage setting in the main text, our results naturally extend to the general multi-stage setting; see Appendix A for details. In what follows, we discuss some additional research questions that are worth studying.

7.1. Alternative Perspectives
The problem studied in this paper can be viewed in the following abstract way: suppose the seller can “adjust” the prices and mean utilities of the products in the first stage through certain structured decisions (in our case, through manipulating the second-stage assortments), how should they do so to maximize revenue? More specifically, in our problem, the possible “adjustments” are the possible expected revenue and utilities generated by the second-stage assortments, which affect the first stage through certain parametric formulas (derived from the multi-stage choice model). How might one tackle this problem if we consider a generic decision set of “adjustments” specified by arbitrarily given formulas? Can we identify generic structural properties and find more interesting interpretations of such generalizations in practice? We leave these questions for future research.

7.2. Future Directions
Our work opens exciting perspectives for future research. First, in this paper, we formulate the multi-stage choice model based on the classical MNL model. A natural lead is to derive new multi-stage choice models based on other popular single-stage choice models, and study the associated assortment optimization / multi-product pricing problems. Second, in Section 4.3, we provide a 2-approximation algorithm for the two-stage assortment optimization problem when $0 < \gamma < 1$. An interesting research direction is to develop approximation schemes with even stronger performance guarantees for this case. Last, it would be very interesting to validate our multi-stage choice model using real data and see how effective the optimal assortment policy for the multi-stage choice model is in practice.

References


A. Extensions to More than Two Stages

In this section, we extend the results for the two-stage assortment optimization problem to the multi-stage assortment optimization problem, where the number of stages is larger than two. We focus on the non-trivial case where the customer is fully forward-looking, i.e., $\gamma = 1$. In this case, we will show that almost all our results established for the two-stage problem extend to the multi-stage problem, which indicates that our insights obtained from the two-stage problem are quite general. All proofs of the results in this section are given in Appendix I.

Consider a $T$-stage assortment optimization problem where $T \geq 2$. Recall that in Section 3, we define a $T$-stage assortment policy $\Pi$ as a set of (deterministic) mappings $\Pi = \{f_t, t = 1, \ldots, T\}$ where each $f_t$ maps an "(offering history, choice history)" pair $(h^*_1, h^*_2)$ to an assortment $S_t \subseteq \mathcal{N}_t$. When such an assortment policy $\Pi$ is given, it is evident that for all $t \geq 2$, the seller’s assortment decision at stage $t$ is completely determined by the customer’s choice history $(c_1, \ldots, c_{t-1})$. This motivates us to introduce a more concise way to represent the seller’s assortment decisions under a given policy $\Pi$ by defining set-valued functions that only take the customer’s choice history as input — by doing so, we can greatly simplify the notations when $T$ is large. Specifically, for any $T$-stage assortment policy $\Pi$, we define a set $g^\Pi_1$ and a series of set-valued functions $g^\Pi_2, \ldots, g^\Pi_T$ such that

- $g^\Pi_1$ takes no input. The assortment $g^\Pi_1 \subseteq \mathcal{N}_1$ is the seller’s assortment at stage 1.
- $g^\Pi_t(\cdot)$ takes the customer’s choice at stage 1 as input. For all $i_1 \in g^\Pi_1 \cup \{0\}$, the assortment $g^\Pi_2(i_1) \subseteq \mathcal{N}_2$ is the seller’s assortment at stage 2 given that the customer chooses $i_1$ at stage 1.
- Generally, for stage $t = 2, \ldots, T$, $g^\Pi_t(\cdot)$ takes the customer’s choice history before stage $t$ as input. For all $i_1 \in g^\Pi_1 \cup \{0\}, i_2 \in g^\Pi_2(i_1) \cup \{0\}, \ldots, i_{t-1} \in g^\Pi_{t-1}(i_1, \ldots, i_{t-2}) \cup \{0\}$, the assortment $g^\Pi_t(i_1, \ldots, i_{t-1})$ is the seller’s assortment at stage $t$ given that the customer chooses $i_1$ at stage 1, chooses $i_2$ at stage 2, ..., and chooses $i_{t-1}$ at stage $t-1$.

For notational convenience, we also define $i_0 = \emptyset$ and $g^\Pi_T(\emptyset) = g^\Pi_1$.

In the following, we formulate the $T$-stage assortment problem as an optimization problem. Similar to Section 4.2, we view the problem from the perspective of determining an assortment of paths. For any $i_1 \in \mathcal{N}_1 \cup \{0\}, i_2 \in \mathcal{N}_2 \cup \{0\}, \ldots, i_T \in \mathcal{N}_T \cup \{0\}$, let $(i_1, i_2, \ldots, i_T)$ denote the customer’s (multi-stage) action of "choosing $i_1$ at stage 1, choosing $i_2$ at stage 2, ..., and finally choosing $i_T$ in stage $T"$, which we refer to as her $T$-stage choice path. Let $\tilde{\mu}_{i_1 \cdots i_T} = \sum_{t=1}^T \mu_{i_t}^{(t)}$ denote the total expected utility of $(i_1, \ldots, i_T)$. Given any $T$-stage assortment policy $\Pi$, let $\tilde{x}_{i_1 \cdots i_T}^\Pi \in \{0, 1\}$ be the (induced) indicator variable of the availability of $(i_1, \ldots, i_T)$. In particular, we know that $\tilde{x}_{i_1 \cdots i_T}^\Pi = 1$ (i.e., $(i_1, \ldots, i_T)$ is an available path under $\Pi$) if and only if

$$i_1 \in g^\Pi_1 \cup \{0\}, i_2 \in g^\Pi_2(i_1) \cup \{0\}, \ldots, i_T \in g^\Pi_T(i_1, \ldots, i_{T-1}) \cup \{0\}.$$  \hspace{1cm} (15)

Incorporating our new notations into the dynamic programming framework established in Section 3, we obtain the following proposition.

**Proposition A.1.** When $\gamma = 1$, for any assortment policy $\Pi$, the customer’s probability of following the choice path $(i_1, \ldots, i_T)$ is

$$P_{i_1 \cdots i_T}^\Pi := \frac{e^{\tilde{\mu}_{i_1 \cdots i_T}} \tilde{x}_{i_1 \cdots i_T}^\Pi}{\sum_{i_1=0}^{\tilde{x}_{i_1 \cdots i_T}^\Pi} \sum_{i_T=0}^{\tilde{x}_{i_1 \cdots i_T}^\Pi} e^{\tilde{\mu}_{i_1 \cdots i_T}}}.$$  

Therefore, $P_{i_1 \cdots i_T}^\Pi$ can be interpreted as the customer’s probability of choosing path $(i_1, \ldots, i_T)$ among all available paths, which follows an MNL model with parameter $\tilde{\mu}_{i_1 \cdots i_T}$.

Furthermore, we define $\tilde{\pi}_{i_1 \cdots i_T} = \sum_{t=1}^T \tilde{p}_{i_t}^{(t)}$ to be the total revenue of a path $(i_1, \ldots, i_T)$. Then we can rewrite the $T$-stage assortment optimization problem as the following optimization problem:

**Optimization Problem A.1.**

$$\text{Maximize} \quad \tilde{\pi}_{i_1 \cdots i_T}^\Pi \quad \text{subject to} \quad P_{i_1 \cdots i_T}^\Pi = \frac{e^{\tilde{\mu}_{i_1 \cdots i_T}} \tilde{x}_{i_1 \cdots i_T}^\Pi}{\sum_{i_1=0}^{\tilde{x}_{i_1 \cdots i_T}^\Pi} \sum_{i_T=0}^{\tilde{x}_{i_1 \cdots i_T}^\Pi} e^{\tilde{\mu}_{i_1 \cdots i_T}}}.$$  

Electronic copy available at: https://ssrn.com/abstract=3243742
Proposition A.2. When $\gamma = 1$, the $T$-stage assortment optimization problem is equivalent to

$$
\max \mathbf{x} \quad \sum_{t=1}^{n_T} \sum_{i_T - 1}^{n_T} e^{\tilde{\mu}_{i_1 \ldots i_T} x_{i_1 \ldots i_T}} \tilde{p}_{i_1 \ldots i_T}
$$

subject to

$$
\tilde{x}_{i_1 \ldots i_T} \leq \tilde{x}_{i_1 \ldots i_T - 1} 0, \quad \forall t \in \{1, \ldots, T\}, \forall i_1 \in N_1 \cup \{0\}, \ldots, i_T \in N_T \cup \{0\}.$$

Problem (16) is a multi-stage generalization of problem (5) developed in Section 4.2. Similarly to (5), (16) can be viewed as a single-stage assortment optimization problem with $\Pi_{t=1}^{n_T} (n_t + 1) - 1$ products (where each path $(i_1, \ldots, i_T)$ except $(0, \ldots, 0)$ is a product with associated price $\tilde{p}_{i_1 \ldots i_T}$ and utility $\tilde{\mu}_{i_1 \ldots i_T}$). Furthermore, it is intractible to solve (16) in polynomial time. However, the multi-stage structure makes (16) more complicated than (5), in what follows, we will show that almost all our results established for (5) naturally extend to (16).

First, we claim that even when there are multiple stages, i.e., $T > 1$, the multi-stage assortment optimization problem is still polynomial-time solvable when $\gamma = 1$. This is because the constraints in (16) remain totally unimodular — specifically, if we represent the constraints using a matrix, then each row of the matrix contains only a $+1$ and a $-1$, thus the matrix is totally unimodular (see, e.g., Chapter III of Wolsey and Nemhauser 1999). This enables us to adopt the linear programming approach in Davis et al. (2013) to solve (16) in polynomial time. However, the above linear programming approach may become impractical when $T$ increases, as the dimension of the decision variable of the LP is $O(n_1 \cdots n_T)$. Later we will introduce a novel algorithm that significantly improves the complexity of solving this problem.

Second, we claim that similar to (5), problem (16) has very nice structural properties which can provide important guidance in selecting optimal assortments in practice. We have the following theorem on the structural properties of the optimal assortment policy.

Theorem A.1. When $\gamma = 1$, there exists an optimal solution $\Pi^*$ to the $T$-stage assortment optimization problem with $g_1^* = g_1^{P^*}, g_2^* = g_2^{P^*}, \ldots, g_T^* = g_T^{P^*}$ such that

1. (Revenue-Ordered in First Stage) The assortment $g_1^*$ is revenue-ordered;
2. (Revenue-Ordered in Each Stage) For all $t = 2, \ldots, T$, for all $i_1 \in g_1^* \cup \{0\}, i_2 \in g_2^*(i_1) \cup \{0\}, \ldots, i_{t-1} \in g_{t-1}^*(i_1, \ldots, i_{t-2}) \cup \{0\}$, the assortment $g_t^*(i_1, \ldots, i_{t-1})$ is always revenue-ordered.
3. (Nested in Each Stage) For all $t = 2, \ldots, T$, for all $i_1 \in g_1^* \cup \{0\}, i_2 \in g_2^*(i_1) \cup \{0\}, \ldots, i_{t-2} \in g_{t-2}^*(i_1, \ldots, i_{t-3}) \cup \{0\}$, we have

$$
g_t^*(i_1, \ldots, i_{t-2}, 1) \supseteq g_t^*(i_1, \ldots, i_{t-2}, 2) \supseteq \cdots \supseteq g_t^*(i_1, \ldots, i_{t-2}, |g_{t-1}^*(i_1, \ldots, i_{t-2})|).
$$

4. (Higher Total Revenue Leads to Larger Assortment) For all $t = 2, \ldots, T$, for all $i_1 \in g_1^* \cup \{0\}, i_2 \in g_2^*(i_1) \cup \{0\}, \ldots, i_{t-2} \in g_{t-2}^*(i_1, \ldots, i_{t-3}) \cup \{0\}, i_{t-1} \in g_{t-1}^*(i_1, \ldots, i_{t-2})$, and $i'_1 \in g_1^* \cup \{0\}, i'_2 \in g_2^*(i'_1) \cup \{0\}, \ldots, i'_{t-2} \in g_{t-2}^*(i'_1, \ldots, i'_{t-3}) \cup \{0\}, i'_{t-1} \in g_{t-1}^*(i'_1, \ldots, i'_{t-2})$, we have

$$
\begin{cases}
\begin{align*}
g_t^*(i_1, \ldots, i_{t-1}) & \supseteq g_t^*(i'_1, \ldots, i'_{t-1}), & \text{if } p_1^{(1)} + \cdots + p_{i_{t-1}}^{(1)} \geq p_1^{(i_1)} + \cdots + p_{i_{t-1}}^{(i'}), \\
g_t^*(i_1, \ldots, i_{t-1}) & \subseteq g_t^*(i'_1, \ldots, i'_{t-1}), & \text{otherwise}.
\end{align*}
\end{cases}
$$

We provide some interpretations here. Parts 1, 2, and 3 of Theorem A.1 generalize the results of Theorem 4.2, saying that the insights that we obtain from the two-stage model hold true for the multi-stage model in a strong per-stage sense. In particular, no matter how the customer makes choices, the seller’s assortment provided in each stage is always revenue-ordered, and a product with higher revenue should always lead to a wider range of future options. Part 4 of Theorem A.1 further generalizes part 3, saying that under the optimal policy, for any two available $(t - 1)$-stage
choice paths that do not end with 0, the path with a higher total revenue always leads to a larger assortment at stage \( t \). This generalized property is appealing because it does not require the two paths to have overlaps — even if they represent two completely different purchase trajectories with no overlap, we can still apply part 4 by simply comparing the total revenue of the two paths.

We then state some important properties of the optimal policy that would eventually enable us to reduce the challenging \( T \)-stage assortment optimization problem to a standard single-stage assortment optimization problem with \( O(n_1 \cdots n_T) \) products.

**LEMMA 1.** Suppose \( \gamma = 1 \). For any \( t \)-stage assortment optimization problem \( (t = 1, 2, \ldots) \), there exists a unique assortment policy \( \Pi^\text{min} \) among all the optimal assortment policies, such that for any other optimal policy \( \Pi^\text{opt} \), all \( t \)-stage choice paths available (defined in (15)) under \( \Pi^\text{min} \) are also available under \( \Pi^\text{opt} \). We call this unique \( \Pi^\text{min} \) the “minimum optimal assortment policy”.

**PROPOSITION A.3.** Suppose \( \gamma = 1 \). Consider a \( T \)-stage assortment optimization problem \( \Phi_{1:T} \) with \( N_1, \ldots, N_T \) being the product set for each stage. For any \( 1 \leq t_1 \leq t_2 \leq T \), we can construct an “embedded” \((t_2 - t_1 + 1)\)-stage assortment optimization problem \( \Phi_{t_1:t_2} \), where \( N_{t_1}, \ldots, N_{t_2} \) is the product set for each stage. Let \( \Pi_{t_1:t_2}^\text{min} \) be the minimum optimal assortment policy for \( \Phi_{t_1:t_2} \). Let \( m_1 \) be the largest value such that \( \{1, \ldots, m_1\} \) is an optimal assortment for \( \Phi_{1:t_1} \).

There exists an optimal solution \( \Pi^* \) to the \( T \)-stage assortment optimization problem \( \Phi_{1:T} \), with \( g^*_1 = g_1^\Pi^*, g^*_2 = g_2^\Pi^*, \ldots, g_T^* = g_T^\Pi^* \) such that

1. \( g^*_i \subseteq \{1, \ldots, m_1\} \);
2. For all \( t = 1, \ldots, T \), if \((j_1, \ldots, j_T)\) is an available \((T-t+1)\)-stage choice path under \( \Pi_{t_1:t_2}^\text{min} \) for the embedded problem \( \Phi_{t_1:T} \), then for all \( i_1 \in g^*_1 \cup \{0\}, i_2 \in g^*_2(i_1) \cup \{0\}, \ldots, i_{t_2-2} \in g^*_{t_2-2}(i_1, \ldots, i_{t_2-3}) \cup \{0\}, i_{t_2-1} \in g^*_{t_2-1}(i_1, \ldots, i_{t_2-2}) \), the \( T \)-stage choice path \((i_1, \ldots, i_{t_2-1}, j_1, \ldots, j_T)\) is available under \( \Pi^* \) for the original problem \( \Phi_{1:T} \).

We provide some insights on how Proposition A.3 helps us understand the structure of the optimal solution to the \( T \)-stage assortment optimization problem. From part 1, we know that under the optimal \( T \)-stage policy, the first-stage assortment should be smaller than the optimal single-stage assortment if there were only stage 1; by taking \( t = T \) in part 2, we know that under the optimal \( T \)-stage policy, the final-stage assortment should be larger than the optimal single-stage assortment if there were only stage \( T \) — notably, these results generalize Proposition 4.3 proved for the two-stage setting. Moreover, by letting \( t \) take other values between 1 and \( T \) in part 2, we can clearly observe some inclusion relations between the optimal solution to the \( T \)-stage assortment optimization problem and the optimal solutions to some “embedded” \((T-1)\)-stage, \((T-2)\)-stage, \ldots, and single-stage assortment optimization problems. In particular, if we visualize the optimal \( T \)-stage assortment policy using a \( T \)-stage tree (see Figure 1 for an example for the two-stage setting), then part 2 of Proposition A.3 can be intuitively understood as follows: from every non-zero node at stage \( t \) of the \( T \)-stage tree, looking down, the \((T-t+1)\)-stage subtree under this node should completely contain “the \((T-t+1)\)-stage tree representing the optimal \((T-t+1)\)-stage policy if there were only stages \( t \) to \( T \)”.

Proposition A.3 motivates us to design a recursive algorithm that solves the \( T \)-stage assortment optimization problem by using the procedures of solving fewer-stage problems as sub-routines.\(^{12}\)

\(^{12}\)We note that a similar idea of solving an assortment problem via recursion appears in Li et al. (2015), where the authors study the single-stage assortment optimization problem under the multi-level nested logit model. Their recursion strategy relies on the fact that the optimal solution to their full problem simultaneously solves all embedded “local problems”. Our problem requires different recursion strategies and new algorithmic ideas, as 1) the optimal solution to our full problem is not an optimal solution to each embedded problem (there are only some inclusion relations), and 2) the existence of the no-purchase option in each stage makes the solution structure more complicated.
In particular, since the main complexity of directly solving the optimization problem (16) comes from a huge set of constraints

\[ \tilde{x}_{i_t \cdots i_{t-1}i_t \cdots i_T} \leq \tilde{x}_{i_1 \cdots i_{t-1}0 \cdots 0}, \quad \forall t \in \{1, \ldots, T\}, \forall i_1 \in \mathcal{N}_1 \cup \{0\}, \ldots, i_T \in \mathcal{N}_T \cup \{0\}, \]

we seek to recursively reduce the number of the constraints by recursively utilizing the structural properties we just developed, such that problem (16) can be eventually reduced to a standard single-stage assortment optimization problem under the MNL model.

Recall that in the two-stage setting, the above task is achieved by the merging idea (see step 7 of Algorithm 1), which merges every “two-stage choice paths ending with the ‘no-purchase’ option” with multiple “two-stage choice paths ending with high-price products” based on the guidance of Proposition 4.3, such that a certain number of constraints would disappear. Fortunately, we find that this merging idea can be effectively generalized to the multi-stage setting based on the guidance of Proposition A.3, though the detailed implementation requires more careful design and analysis. Based on the generalized merging techniques we develop, we devise the Recursive Merged Path Selection (RMPS) algorithm that efficiently solves the \(T\)-stage assortment optimization in \(O(n_1 \cdots n_T \log(n_1 \cdots n_T))\) time, which significantly improves the complexity of solving a large-scale linear program with \(\Omega(n_1 \cdots n_T)\) variables. The RMPS algorithm generalizes the MPS algorithm developed for the two-stage setting, and we present the algorithmic details of RMPS in Appendix I.4.

**Proposition A.4.** When \(\gamma = 1\), the Recursive Merged Path Selection algorithm (see Appendix I.4) can find an optimal solution to (16) in \(O(n_1 \cdots n_T \log(n_1 \cdots n_T))\).

### B. Some (Counter)examples

**B.1. Revenue-ordered property may not hold for Problem (5)**

Consider an instance of the two-stage assortment optimization problem where \(\gamma = 1\). There is one product in stage 1 (i.e., \(\mathcal{N}_1 = \{1\}\)), with \(p_1^{(1)} = 1\) and \(\mu_1^{(1)} = 1\). There is one product in stage 2 (i.e., \(\mathcal{N}_2 = \{1\}\)), with \(p_2^{(2)} = 2\) and \(\mu_2^{(2)} = 1\).

We consider the equivalent Problem (5), where there are three non-zero paths: \((1,1), (0,1)\) and \((1,0)\). The revenues of these paths are \((\tilde{p}_{11}, \tilde{p}_{01}, \tilde{p}_{10}) = (3, 2, 1)\), and the mean utilities of these paths are \((\tilde{\mu}_{11}, \tilde{\mu}_{01}, \tilde{\mu}_{10}) = (2, 1, 1)\).

Based on the set of constraints \(\tilde{x}_{ij} \leq \tilde{x}_{i0}\), we know that there are only five feasible non-empty assortments for Problem (5): \{(1, 1), (0, 1), (1, 0)\}, \{(1, 1), (1, 0)\}, \{(0, 1), (1, 0)\}, \{(0, 1)\} and \{(1, 0)\}. Among them, \{(1, 1), (0, 1), (1, 0)\} is the only revenue-ordered one, and its expected revenue is \((3e^2 + 2e + 1e)/(1 + e^2 + e + e) = 2.1932\). However, the expected revenue of \{(1, 1), (1, 0)\} is \((3e^2 + 1e)/(1 + e^2 + e) = 2.2405 > 2.1932\). Therefore, we know that the revenue-ordered property fails for Problem (5).

**B.2. Revenue-ordered property may not hold for \(f^*(S_t^*, 0)\) when \(0 < \gamma < 1\)**

Consider an instance of the two-stage assortment optimization problem where \(\gamma = 0.5\). There is one product in stage 1 (i.e., \(\mathcal{N}_1 = \{1\}\)), with \(p_1^{(1)} = 5\) and \(\mu_1^{(1)} = 0.5\). There are two products in stage 2 (i.e., \(\mathcal{N}_2 = \{1, 2\}\)), with \((\tilde{p}_1^{(2)}, \tilde{p}_2^{(2)}) = (4, 3.9)\) and \((\mu_1^{(2)}, \mu_2^{(2)}) = (1, -1)\).

We next find the optimal assortment policy. Since there is only one product in stage 1, it is easy to check that offering nothing in the first stage is not optimal; thus \(S_1^* = \{1\}\). Since \(\{1, 2\}\) is the optimal single-stage assortment for stage 2, according to Proposition 4.6, \(f^*(S_1^*, 1)\) must be \(\{1, 2\}\). Thus, the only task left is to determine \(f^*(S_t^*, 0)\), which can be \(\emptyset\), \{1\}, \{2\} or \{1, 2\}. We calculate the expected revenue of each case, and the results are shown in Table 1. It can be found that under the optimal policy, \(f^*(S_t^*, 0) = \{2\}\), which is not revenue-ordered.
We provide some insights on why \( f^*(S_1^*,0) \) is not revenue-ordered in this example. The two products in stage 2 have very similar prices (4 and 3.9), but product 2 leads to considerably lower customer utility. When deciding the optimal assortment policy, setting \( f^*(S_1^*,0) = \{2\} \) has some unique advantages: compared with setting \( f^*(S_1^*,0) = \{1,2\} \) (which would be the optimal strategy if \( \gamma = 0 \)), setting \( f^*(S_1^*,0) = \{2\} \) incentivizes the customer to purchase more in stage 1 and leads to higher revenue in stage 1 when \( \gamma = 0.5 \); compared with setting \( f^*(S_1^*,0) = \emptyset \) (which would be the optimal strategy if \( \gamma = 1 \)), setting \( f^*(S_1^*,0) = \{2\} \) does not significantly affect the customer’s purchase probability in stage 1 when \( \gamma = 0.5 \) (since \( \mu_2^{(2)} = -1 \) is low) but can contribute to higher revenue in stage 2 if the customer chooses 0 in stage 1 (although \( \mu_2^{(2)} = -1 \) is low, the price \( p_2^{(2)} = 3.9 \) is quite good for the seller). Here, the choice of \( \gamma = 0.5 \) (not too small but not too large) plays a key role in making \( f^*(S_1^*,0) = \{2\} \).

### B.3. Static policy fails to provide a constant-factor approximation when \( \gamma > 0 \)

Consider an instance of the two-stage assortment optimization problem where \( \gamma > 0 \). There is one product in stage 1 (i.e., \( N_1 = \{1\} \)), with \( p_1^{(1)} = z \) and \( \mu_1^{(1)} = \log(z-1) \) (it is an expensive product with low purchasing probability, like a luxury good). There is one product in stage 2 (i.e., \( N_2 = \{1\} \)), with \( p_1^{(2)} = \frac{z}{z-1} \) and \( \mu_1^{(2)} = \log(z-1) \) (it is an affordable product with high utility, like a bargain).

If the seller uses a static policy, then he will offer: \( S_1 = \{1\}, S_2 = \{1\} \), and his two-stage expected revenue will be 1 + 1 = 2. If the seller commits to the following policy: \( S_1 = \{1\}, f(S_1,0) = \emptyset, f(S_1,1) = \{1\} \), then his two-stage expected revenue will be \( \frac{z^2}{z-1} \). There are two products in stage 2 (i.e., \( N_2 = \{1,2\} \)), with \( (p_1^{(2)}, p_2^{(2)}) = (10,1) \) and \( (\mu_1^{(2)}, \mu_2^{(2)}) = (0.1,10) \).

We next find the optimal deterministic assortment policy. Since there is only one product in stage 1, it is easy to check that offering nothing in the first stage is not optimal; thus \( S_1^* = \{1\} \). Since \( \{1\} \) is the optimal single-stage assortment for stage 2, according to Proposition 4.6, \( f^*(S_1^*,1) \) must be either \( \{1\} \) or \( \{1,2\} \). Thus, to search for the optimal deterministic policy, we only need to consider eight cases, where \( f^*(S_1^*,0) \in \{\emptyset, \{1\}, \{2\}, \{1,2\}\} \) and \( f^*(S_1^*,1) \in \{\{1\}, \{1,2\}\} \). We calculate the expected revenue of each case, and the results are shown in Table 2. It can be found that the expected revenue of the optimal deterministic policy is 10.9500.

We now provide a randomized assortment policy that achieves higher expected revenue than all deterministic policies. Consider the randomized policy described in the last row of Table 2; the expected revenue of this policy is

\[
\frac{1}{2} \log(1+e^{\mu_1^{(1)}}) + \frac{1}{2} \log(1+e^{\mu_1^{(1)}+\mu_2^{(2)}}) + \frac{1}{2} \log(1+e^{\mu_1^{(1)}+\mu_2^{(2)}})
\]

<table>
<thead>
<tr>
<th>Assortment Policy</th>
<th>Expected Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_1 = {1}, f(S_1,0) = \emptyset, f(S_1,1) = {1,2} )</td>
<td>6.1269</td>
</tr>
<tr>
<td>( S_1 = {1}, f(S_1,0) = {1}, f(S_1,1) = {1,2} )</td>
<td>6.1473</td>
</tr>
<tr>
<td>( S_1 = {1}, f(S_1,0) = {2}, f(S_1,1) = {1,2} )</td>
<td><strong>6.2033</strong></td>
</tr>
<tr>
<td>( S_1 = {1}, f(S_1,0) = {1,2}, f(S_1,1) = {1,2} )</td>
<td>6.1244</td>
</tr>
</tbody>
</table>

Table 1: Expected Revenue of Different Policies

Electronic copy available at: https://ssrn.com/abstract=3243742
Table 2: Expected Revenue of Different Policies

<table>
<thead>
<tr>
<th>Deterministic Assortment Policy</th>
<th>Expected Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1 = {1}, f(S_1, 0) = \emptyset, f(S_1, 1) = {1}$</td>
<td>9.3924</td>
</tr>
<tr>
<td>$S_1 = {1}, f(S_1, 0) = \emptyset, f(S_1, 1) = {1, 2}$</td>
<td>10.9337</td>
</tr>
<tr>
<td>$S_1 = {1}, f(S_1, 0) = {1}, f(S_1, 1) = {1}$</td>
<td>10.4996</td>
</tr>
<tr>
<td>$S_1 = {1}, f(S_1, 0) = {1}, f(S_1, 1) = {1, 2}$</td>
<td><strong>10.9500</strong></td>
</tr>
<tr>
<td>$S_1 = {1}, f(S_1, 0) = {2}, f(S_1, 1) = {1}$</td>
<td>1.1523</td>
</tr>
<tr>
<td>$S_1 = {1}, f(S_1, 0) = {1, 2}, f(S_1, 1) = {1}$</td>
<td>1.1527</td>
</tr>
<tr>
<td>$S_1 = {1}, f(S_1, 0) = {1, 2}, f(S_1, 1) = {1, 2}$</td>
<td>6.2502</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Randomized Assortment Policy</th>
<th>Expected Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1 = {1}, f(S_1, 0) = {1}, f(S_1, 1) = \begin{cases} {1}, &amp; \text{with probability 0.5} \ {1, 2}, &amp; \text{with probability 0.5} \end{cases}$</td>
<td><strong>12.4784</strong></td>
</tr>
</tbody>
</table>

which equals 12.4784. Since its expected revenue is larger than 10.9500, it outperforms all deterministic policies.

Finally, we provide some explanations on why randomization helps in this example. The two products in stage 2 are completely different: product 1 has a much higher price and leads to much lower customer utility, while product 2 has a much lower price and leads to much higher customer utility. When deciding the optimal deterministic assortment policy, the seller faces a difficult trade-off between setting $f(S_1, 1) = \{1\}$ and setting $f(S_1, 1) = \{1, 2\}$: the former (which excludes product 2 in stage 2) leads to higher revenue in stage 2, but the latter (which uses product 2 in stage 2 to incentivize the customer to buy more in stage 1) leads to higher revenue in stage 1. When randomization is allowed, the seller faces a much “smoother” trade-off, thus can do better (in other words, the decision space is relaxed to a continuous one). We mention again that while randomized policies may be appealing in theory, they are hard to implement in practice due to the limited rationality of customers and stronger information requirements.

### C. Proof of Proposition 4.2 and Proposition 4.5

We first give a generalized proposition that includes both Proposition 4.2 and Proposition 4.5 as special cases. Then we prove our new proposition, thus prove Proposition 4.2 and Proposition 4.5 together. For convenience, now we generalize the range of $\gamma$ in problem (9), from $(0, 1)$ to $[0, 1]$.

**PROPOSITION C.1.** For problem (9), there exists an optimal solution $\tilde{x}^*$ satisfying the following properties:

1. For any $i \in N_1$ and $j \in N_2 \cup \{0\}$, if $\tilde{x}_{ij}^* = 1$, then for any $0 \leq k \leq j$, $\tilde{x}_{ik}^* = 1$;
2. For any $i \in N_1 \cup \{0\}$ and $j \in N_2 \cup \{0\}$, if $\tilde{x}_{ij}^* = 1$, then for any $1 \leq k \leq i$, $\tilde{x}_{kj}^* = 1$.

If $\gamma = 1$, then there is an additional property:

3. (When $\gamma = 1$) For any $j \in N_2 \cup \{0\}$, if $\tilde{x}_{0j}^* = 1$, then for any $0 \leq k \leq j$, $\tilde{x}_{0k}^* = 1$.

In the following, we provide a proof for Proposition C.1. We start with some definitions. According to the equivalence between problem (9) and (2), any feasible solution $\tilde{x}$ to problem (9) corresponds to a feasible solution $(x, y)$ to problem (2) (and thus corresponds to a feasible assortment policy in the two-stage assortment optimization problem). For any feasible $\tilde{x}$, we define

$$V_i(\tilde{x}) = \log \left( \sum_{j=0}^{n_2} e^{\nu_j^{(2)}} y_{ij} \right) = \log \left( \sum_{j=0}^{n_2} e^{\nu_j^{(2)}} \tilde{x}_{ij} \right), \quad i \in S_1 \cup \{0\},$$
\[ r_i(\bar{x}) = p_i^{(1)} + \sum_{j=0}^{n_2} e^{(2)}_{ij} x_{ij} p_j^{(2)} = p_i^{(1)} + \sum_{j=0}^{n_2} e^{(2)}_{ij} \hat{x}_{ij} p_j^{(2)}, \quad i \in S_1 \cup \{0\}, \]

where \( V_i(\bar{x}) \) and \( r_i(\bar{x}) \) represent the customer’s expected utility in stage 2 and the seller’s expected two-stage revenue, respectively, given that the customer chooses \( i \) in stage 1. Also, for \( i \notin S_1 \cup \{0\} \), we let \( V_i(\bar{x}) = 0 \), \( r_i(\bar{x}) = 0 \). Thus, using our new definitions, given any feasible solution \( \bar{x} \) to problem (9), we can represent the seller’s expected overall revenue as

\[ R(\bar{x}) = \frac{\sum_{i=0}^{n_1} e^{(1)}_{i0} + \gamma V_i(\bar{x}) x_{i0} r_i(\bar{x})}{\sum_{i=0}^{n_1} e^{(1)}_{i0} + \gamma V_i(\bar{x}) \hat{x}_{i0}}. \]

We have the following lemmas:

**Lemma 2.** Let \( \bar{x}^* \) be an optimal solution to (9). For any \( i \in N_1 \), if \( \hat{x}_{i0} = 1 \), then \( r_i(\bar{x}^*) \geq R(\bar{x}^*) \).

**Proof of Lemma 2.** For notational simplicity, let \( r_i^* = r_i(\bar{x}^*) \) and \( V_i^* = V_i(\bar{x}^*) \) (\( i \in N_1 \cup \{0\} \)). We prove Lemma 2 by contradiction. Suppose there exist \( i' \in N_1 \) such that \( \hat{x}^*_{i'0} = 1 \) and \( r_i^* < R(\bar{x}^*) \). Now we construct another solution \( \bar{x}' \) by first letting \( \bar{x}' = \bar{x}^* \) then modifying \( \hat{x}'_{i'0}, \hat{x}'_{i'1}, \ldots, \hat{x}'_{i'n_2} \) to be 0 (i.e., no longer offering product \( i' \)). We have

\[ R(\bar{x}') = \frac{\sum_{i=0}^{n_1} e^{(1)}_{i0} + \gamma V_i(\bar{x}') x_{i0} r_i(\bar{x}')}{\sum_{i=0}^{n_1} e^{(1)}_{i0} + \gamma V_i(\bar{x}') \hat{x}_{i0}} > \frac{\sum_{i=0}^{n_1} e^{(1)}_{i0} + \gamma V_i^* x_{i0} r_i^*}{\sum_{i=0}^{n_1} e^{(1)}_{i0} + \gamma V_i^* \hat{x}_{i0}} = R(\bar{x}^*), \]

where the inequality is because \( r_i^* < R(\bar{x}^*) \). However, this contradicts the assumption that \( \bar{x}^* \) is optimal. Therefore, Lemma 2 is established.

**Lemma 3.** Let \( \bar{x} \) be a feasible solution to (9), then \( \tilde{S} = \{(i,j) | \hat{x}_{ij} = 1, (i,j) \neq (0,0)\} \) is the corresponding assortment of offered paths (excluding (0,0)). We have the following properties:

1. Assume there exists a path \((i,j) \in \tilde{S} \) such that \( \tilde{p}_{ij} < \gamma R(\bar{x}) + (1-\gamma)r_i(\bar{x}), r_i(\bar{x}) \geq R(\bar{x}) \), and \( i,j \neq 0 \). Then, removing path \((i,j) \) from \( \tilde{S} \) yields a strictly larger expected revenue than \( R(\bar{x}) \).
2. Assume there exists a path \((i,j) \notin \tilde{S} \) such that \( \tilde{p}_{ij} \geq \gamma R(\bar{x}) + (1-\gamma)r_i(\bar{x}), r_i(\bar{x}) \geq R(\bar{x}) \), and \( i,j \neq 0 \). If \((i,0) \in \tilde{S} \), then, adding path \((i,j) \) into \( \tilde{S} \) yields an expected revenue larger than or equal to \( R(\bar{x}) \).
3. Assume \( \gamma = 1 \). Assume there exists a path \((0,j) \in \tilde{S} \) such that \( \tilde{p}_{0j} < R(\bar{x}), r_i(\bar{x}) \neq 0 \). Then, removing path \((0,j) \) from \( \tilde{S} \) yields a strictly larger expected revenue than \( R(\bar{x}) \).

**Proof of Lemma 3.** We first prove part 1 of Lemma 3. Since \( i,j \neq 0 \), we are able to remove path \((i,j) \) from \( \tilde{S} \) without violating \( \hat{x}_{ij} \leq \hat{x}_{i0} \). Let \( \tilde{S}' \) be the assortment constructed by removing path \((i,j) \) from \( \tilde{S} \), and let \( \bar{x}' \) be the corresponding solution. We show that \( \bar{x}' \) yields an expected revenue of \( R(\bar{x}') > R(\bar{x}) \). For notational simplicity, let \( V_k = V_k(\bar{x}), V'_k = V_k(\bar{x}') \) and \( r_k = r_k(\bar{x}), r'_k = r_k(\bar{x}') \) (\( k \in N_1 \cup \{0\} \)). Obviously, for all \( k \neq i \), we have \( V_k = V'_k, r_k = r'_k \) and \( \hat{x}_{k0} = \hat{x}_{k0} \). Since \((i,j) \in \tilde{S} \) (i.e., \( \hat{x}_{ij} = 1 \) and \( \hat{x}_{i0} \geq \hat{x}_{ij} \), we have \( \hat{x}_{i0} = 1 \), then by \( j \neq 0 \) we have \( \hat{x}_{j0} = 1 \). Therefore, we can write the expected revenue from the assortment \( \tilde{S} \) as

\[ R(\bar{x}) = \frac{\sum_{k=0}^{n_1} e^{(1)}_{i0} + \gamma V_k \hat{x}_{k0} r_k}{\sum_{k=0}^{n_1} e^{(1)}_{i0} + \gamma V_k \hat{x}_{k0}} = \frac{e^{(1)}_{i0} + \gamma V_0 x_{i0} r_i}{e^{(1)}_{i0} + \gamma V_0 \hat{x}_{i0}} = \frac{e^{(1)}_{i0} + \gamma V_0 x_{i0} r_i + e^{(1)}_{i0} + \gamma V_0 x_{i0} r_i}{e^{(1)}_{i0} + \gamma V_0 \hat{x}_{i0}} \]

Electronic copy available at: https://ssrn.com/abstract=3243742
Since \( \gamma \) violating \( \tilde{\gamma} \) we have \( r_i \). We want to show that \( \gamma V \) violating \( \tilde{\gamma} \). Let \( \gamma V \) be the corresponding solution. We next prove part 2 of Lemma 3. For notational simplicity, let \( V_k = V_k(\tilde{x}) \), \( V_k' = V_k(\tilde{x}') \) and \( r_k = r_k(\tilde{x}) \). Therefore, \( e^\gamma V_i r_i - e^\gamma V_i' r_i' \leq (e^\gamma V_i - e^\gamma V_i') R(\tilde{x}) \) holds if and only if

\[
e^\gamma V_i r_i - e^\gamma V_i' \left[ \frac{1}{\alpha} r_i - \frac{1 - \alpha}{\alpha} \tilde{p}_{ij} \right] \leq (e^\gamma V_i - e^\gamma V_i') R(\tilde{x}).
\]

By simple algebra, we can find that the above inequality holds if and only if \( \tilde{p}_{ij} < g(\alpha)R(\tilde{x}) + (1 - g(\alpha))r_i \), where \( g(\alpha) = (1 - \alpha^\gamma)/(\alpha^\gamma - 1 - \alpha^\gamma) \). Since \( g(\alpha) \) is increasing in \( \alpha \) when \( \gamma \leq 1 \), by L’Hospital’s rule, we have \( g(\alpha) \leq \lim_{\alpha \to 1} g(\alpha) = \gamma \). Along with the facts that \( \tilde{p}_{ij} < \gamma R(\tilde{x}) + (1 - \gamma)r_i \) and \( R(\tilde{x}) \leq r_i \), we have \( \tilde{p}_{ij} < \gamma R(\tilde{x}) + (1 - \gamma)r_i \). Let \( \alpha = e^\gamma V_i' - e^\gamma V_i \), so \( e^\gamma V_i' = \alpha e^\gamma V_i \). Using the fact that \( e^\gamma V_i' = e^\gamma V_i' \), we can write \( r_i' = \alpha r_i + (1 - \alpha)\tilde{p}_{ij} \). Therefore, \( (e^\gamma V_i' - e^\gamma V_i) R(\tilde{x}) \leq e^\gamma V_i r_i' - e^\gamma V_i r_i \) holds if and only if

\[
\frac{e^\gamma V_i}{\alpha^\gamma} [\alpha r_i + (1 - \alpha)\tilde{p}_{ij}] - e^\gamma V_i' r_i \geq (e^\gamma V_i' - e^\gamma V_i) R(\tilde{x}).
\]

By simple algebra, we can find that the above inequality holds if and only if \( \tilde{p}_{ij} \geq h(\alpha)R(\tilde{x}) + (1 - h(\alpha))r_i \), where \( h(\alpha) = (1 - \alpha^\gamma)/(1 - \alpha) \). Since \( h(\alpha) \) is decreasing in \( \alpha \) when \( \gamma \leq 1 \), by L’Hospital’s rule, we have \( h(\alpha) \geq \lim_{\alpha \to 1} h(\alpha) = \gamma \). Along with the facts that \( \tilde{p}_{ij} \geq \gamma R(\tilde{x}) + (1 - \gamma)r_i \) and \( R(\tilde{x}) \leq r_i \), we have \( \tilde{p}_{ij} \geq \gamma R(\tilde{x}) + (1 - \gamma)r_i \geq h(\alpha)R(\tilde{x}) + (1 - h(\alpha))r_i \).

We then prove part 3 of Lemma 3. Since \( j \neq 0 \), we are able to remove path \((0, j)\) from \( \tilde{S} \) without violating \( \tilde{x}_{0j} \leq \tilde{x}_{00} = 1 \). Let \( \tilde{S}' \) be the assortment constructed by removing path \((0, j)\) from \( \tilde{S} \), and let \( \tilde{x}' \) be the corresponding solution. We show that \( \tilde{x}' \) yields an expected revenue of \( R(\tilde{x}') > R(\tilde{x}) \). When \( \gamma = 1 \), problem (9) reduces to problem (5). Since \( \tilde{p}_{0j} < R(\tilde{x}) \), we have

\[
R(\tilde{x}') = \frac{\sum_{k=0}^{n_1} \sum_{l=0}^{n_2} e^{\tilde{\beta}_{kl}} \tilde{x}_{kl} \tilde{p}_{kl} - e^{\tilde{\beta}_{0j}} \tilde{p}_{0j}}{\sum_{k=0}^{n_1} \sum_{l=0}^{n_2} e^{\tilde{\beta}_{kl}} \tilde{x}_{kl} - e^{\tilde{\beta}_{0j}}} > \frac{\sum_{k=0}^{n_1} \sum_{l=0}^{n_2} e^{\tilde{\beta}_{kl}} \tilde{x}_{kl} \tilde{p}_{kl}}{\sum_{k=0}^{n_1} \sum_{l=0}^{n_2} e^{\tilde{\beta}_{kl}} \tilde{x}_{kl}} = R(\tilde{x}).
\]

Therefore, part 3 of Lemma 3 is established. □
Remark. In the proof of part 1 and part 2 of Lemma 3, we use similar techniques to the proof of Lemma 3 and Theorem 4 in Davis et al. (2014). However, due to the unique structure of our problem, we have to carefully check the indices of paths in our analysis, and add more requirements in the statement of our lemma. In particular, in part 1 and part 2, we require that \( i \neq 0 \) and \( j \neq 0 \). These requirements are closely related to the constraints appearing in problem (9): since \( \bar{x}_{00} = 1 \), all paths whose first index is 0 have to be excluded from part 1 and part 2, as Lemma 2 may fail for \( i = 0 \); since \( \bar{x}_{ij} \leq \bar{x}_{i0} \), all paths whose second index is 0 have to be excluded from part 1 and part 2, as adding or removing any of these paths may violate the constraints.

Having established Lemmas 2 and 3, we now proceed to prove Proposition C.1.

**Proof of Proposition C.1.** Suppose \( \bar{x}^* \) is an optimal solution to (9). In the following, we show that we can adjust \( \bar{x}^* \) such that it satisfies all the properties in Proposition C.1.

We first show part 1 of Proposition C.1. Suppose there exist \( i' \in \mathcal{N}_1 \), \( j', k' \in \mathcal{N}_2 \cup \{0\} \) such that \( \bar{x}_{i'j'} = 1, k' \leq j' \) and \( \bar{x}_{i'k'} = 0 \). Since \( \bar{x}_{i'j'} = 1 \), according to the constraint \( \bar{x}_{i'j'}^* \leq \bar{x}_{i'0}^* \), we know that \( \bar{x}_{i'0}^* = 1 \). Also, from \( \bar{x}_{i'k'} = 0 \) we know that \( k' \neq 0 \). Now define a new solution \( \bar{x}' \) which is the same as \( \bar{x}^* \) except that we let \( \bar{x}_{i'k'} = 1 \). First, \( \bar{x}' \) is still feasible because \( \bar{x}_{i'0} = 1 \). Since \( \bar{x}_{i'j'} = 1 \) and \( j' \neq 0 \), according to part 1 of Lemma 3, \( \bar{p}_{i'j'} \geq \gamma R(\bar{x}^*) + (1 - \gamma) r_{i'}(\bar{x}^*) \). Furthermore, because \( 1 \leq k' \leq j' \), we know that \( \bar{p}_{i'k'} = p_{i'k'}^{(1)} + p_{i'k'}^{(2)} \geq p_{i'k'}^{(1)} + p_{i'k'}^{(2)} = \bar{p}_{i'k'} \geq \gamma R(\bar{x}^*) + (1 - \gamma) r_{i'}(\bar{x}^*) \). Therefore, according to part 2 of Lemma 3, we have \( R(\bar{x}') \geq R(\bar{x}^*) \), which indicates that \( \bar{x}^* \) is also an optimal solution. Repeat the argument until there is no such \( i', j', k' \) exist and we will obtain an optimal solution that satisfies the property in part 1 of Proposition C.1.

We next show part 2 of Proposition C.1. Let \( \bar{x}^* \) be an optimal solution that satisfies part 1. We consider two cases: \( j = 0 \) and \( j \in \mathcal{N}_2 \).

Case 1: \( j = 0 \). Suppose there exist \( i', k' \in \mathcal{N}_1 \) such that \( \bar{x}_{i'0} = 1, k' \leq i' \) and \( \bar{x}_{k'0} = 0 \). We construct a new feasible solution \( \bar{x}' \) by first letting \( \bar{x}' = \bar{x}^* \) and then modifying it to make \( \bar{x}'_{k'} = \bar{x}'_{i'j'} \) for all \( j \in \mathcal{N}_2 \). Since \( \bar{x}_{i'0} = 1 \) and \( i' \neq 0 \), according to Lemma 2, \( r_{i'}(\bar{x}^*) \geq R(\bar{x}^*) \). Since \( 1 \leq k' \leq i' \), we know that \( p_{k'}^{(1)} \geq p_{i'}^{(1)} \). Along with the conditions that \( \bar{x}'_{k'} = \bar{x}'_{i'j'} \), we have

\[
\sum_{j=0}^{n_2} e_{i'j'}^{(2)} \bar{x}'_{i'j'}^{*} \geq p_{i'}^{(2)} + \sum_{j=0}^{n_2} e_{i'j'}^{(2)} \bar{x}'_{i'j'}^{*} \geq p_{i'}^{(2)}
\]

Therefore, we have

\[
R(\bar{x}') = \sum_{i=0}^{n_1} e_{i0}^{(1)} + \gamma V_i(\bar{x}^*) \bar{x}_{i0} \bar{x}_{i0} + e_{i0}^{(1)} + \gamma V_{i'}(\bar{x}^*) r_{i'}(\bar{x}') \geq \sum_{i=0}^{n_1} e_{i0}^{(1)} + \gamma V_i(\bar{x}^*) \bar{x}_{i0} \bar{x}_{i0} + \sum_{i=0}^{n_1} e_{i0}^{(1)} + \gamma V_{i'}(\bar{x}^*) \bar{x}_{i0} \bar{x}_{i0} = R(\bar{x}^*)
\]

where the inequality is because \( r_{i'}(\bar{x}') \geq R(\bar{x}^*) \). Therefore, \( \bar{x}' \) is also optimal. Repeat the argument until there is no such \( i', k' \) and we will obtain an optimal solution that satisfies the property in part 2 for \( j = 0 \).

Case 2: \( j \in \mathcal{N}_2 \). Let \( \bar{x}^* \) be an optimal solution we obtain from the last step (which satisfies part 1, as well as part 2 for \( j = 0 \)). Suppose there exist \( i', k' \in \mathcal{N}_1 \), \( j' \in \mathcal{N}_2 \) such that \( \bar{x}_{i'j'} = 1, k' \leq j' \) and \( \bar{x}_{k'j'} = 0 \). Since \( \bar{x}_{i'j'} = 1 \), we have \( \bar{x}_{i'0} \geq \bar{x}_{i'j'} = 1 \). Since \( \bar{x}_{i'j'} = 1 \), according to Case 1, we know that \( \bar{x}_{i'0}^* = 1 \). Now we construct a new solution \( \bar{x}' \) such that \( \bar{x}' = \bar{x}^* \) except that \( \bar{x}_{i'0} = 1 \). First, \( \bar{x}' \) is still feasible because the constraint \( \bar{x}_{i'0}^* \leq \bar{x}_{i'0} \) still holds. Also, since \( \bar{x}_{i'j'}^* = 1 \) and \( j' \neq 0 \), according to part 1 of Lemma 3, \( \bar{p}_{i'j'} \geq \gamma R(\bar{x}^*) + (1 - \gamma) r_{i'}(\bar{x}^*) \). Furthermore, from \( 1 \leq k' \leq i' \), we know that \( \bar{p}_{i'j'} = p_{i'j'}^{(1)} + p_{i'j'}^{(2)} \geq p_{i'j'}^{(1)} + p_{i'j'}^{(2)} = \bar{p}_{i'j'} \geq \gamma R(\bar{x}^*) + (1 - \gamma) r_{i'}(\bar{x}^*) \). Therefore, according to part 2 of Lemma 3, we have \( R(\bar{x}') \geq R^* \), which indicates that \( \bar{x}' \) is also an optimal solution. Repeat the argument until there is no such \( i', j', k' \) and we will obtain an optimal solution that satisfies all the properties in part 1 and part 2 of Proposition C.1.
We then show part 3 of Proposition C.1. When \( \gamma = 1 \), problem (9) reduces to problem (5). Let \( \tilde{x}^* \) be an optimal solution that satisfies part 1 and part 2. Suppose there exist \( j', k' \in \mathcal{N}_2 \cup \{0\} \) such that \( \tilde{x}_{0j'} = 1, k' \leq j' \) and \( \tilde{x}_{0k'} = 0 \). Since \( \tilde{x}_{0j'} = 1 \) and \( \tilde{x}_{0k'} = 0 \) we know that \( k' \neq 0 \). Now define a new feasible solution \( \tilde{x}' \) which is the same as \( \tilde{x}^* \) except that we let \( \tilde{x}_{0k'} = 1 \). Since \( \tilde{x}_{0j'} = 1 \) and \( j' \neq 0 \), according to part 3 of Lemma 3, \( \bar{p}_{0j'} \geq R(\tilde{x}^*) \). Furthermore, because \( 1 \leq k' \leq j' \), we know that \( \bar{p}_{0k'} = p_{0}^{(1)} + p_{k'}^{(2)} \geq p_{0}^{(1)} + p_{k'}^{(2)} = \bar{p}_{0j'} \geq R(\tilde{x}^*) \). Therefore, we have

\[
R(\tilde{x}') = \frac{\sum_{i=0}^{n_1} \sum_{j=0}^{n_2} e^{\mu_{ij}} \tilde{x}_{ij} \tilde{p}_{ij} + e^{\mu_{0j'}} \bar{p}_{0j'}}{\sum_{i=0}^{n_1} \sum_{j=0}^{n_2} e^{\mu_{ij}} \tilde{x}_{ij} + e^{\mu_{0j'}}} \geq \frac{\sum_{i=0}^{n_1} \sum_{j=0}^{n_2} e^{\mu_{ij}} \tilde{x}_{ij} \tilde{p}_{ij}}{\sum_{i=0}^{n_1} \sum_{j=0}^{n_2} e^{\mu_{ij}} \tilde{x}_{ij}} = R(\tilde{x})
\]

which indicates that \( \tilde{x}' \) is also an optimal solution. Repeat the argument until there is no such \( j', k' \) exist and we will obtain an optimal solution that satisfies all the properties in part 1, part 2 and part 3 of Proposition C.1.

\[\square\]

D. Proof of Proposition 4.3 and Proposition 4.6

Since Proposition 4.6 is a strict generalization of Proposition 4.3, we prove Proposition 4.6 directly, and thus prove Proposition 4.3 as a special case.

**Proof of Proposition 4.6.** Let \((x^*, y^*)\) be an optimal solution to problem (2) that satisfies the properties of Theorem 4.3 (if \( 0 < \gamma < 1 \)), Theorem 4.2 (if \( \gamma = 1 \)) or Theorem 4.1 (if \( \gamma = 0 \)). Then the corresponding \( S_1^* \) and \( f^*(S_1^*, i) \) \((i \in S_1^*) \) are all revenue-ordered. Let \( R^* \) be the corresponding expected overall revenue. For notational simplicity, we define

\[
V_i^* = \log \left( \sum_{j=0}^{n_2} e^{\mu_{ij}} y_{ij}^* \right), \quad r_i^* = p_i^{(1)} + \sum_{j=0}^{n_2} e^{\mu_{ij}} y_{ij}^* p_j^{(2)}, \quad i \in \mathcal{N}_1 \cup \{0\},
\]

\[
V_i' = \log \left( \sum_{j=0}^{n_2} e^{\mu_{ij}} \right), \quad r_i' = p_i^{(1)} + \sum_{j=0}^{n_2} e^{\mu_{ij}} p_j^{(2)}, \quad i \in \mathcal{N}_1 \cup \{0\}.
\]

We first show \( S_1^* \subseteq \{1, \ldots, m_1\} \). When \( m_1 = n_1 \), this naturally holds. When \( m_1 < n_1 \), since \( m_1 \) is the largest value that make \( \{1, ..., m_1\} \) optimal for stage 1, we have \( p_k^{(1)} < \left( \sum_{j=0}^{m_1} e^{\mu_{ij}} \right) / \left( \sum_{j=0}^{m_1} e^{\mu_{ij}} \right) \) for all \( k \geq m_1 + 1 \). Therefore, for all \( k \geq m_1 + 1 \),

\[
r_k^* = p_k^{(1)} + \sum_{j=0}^{m_2} e^{\mu_{kj}} y_{kj}^* p_j^{(2)} \geq \sum_{j=0}^{m_1} e^{\mu_{ij}} p_j^{(1)} + \sum_{j=0}^{m_2} e^{\mu_{kj}} p_j^{(2)} = R^*.
\]

According to Lemma 2, \( k \notin S_1^* \) for all \( k \geq m_1 + 1 \). Thus \( S_1^* \subseteq \{1, \ldots, m_1\} \).

We then show \( f^*(S_1^*, k) \supseteq \{1, \ldots, m_2\} \) for all \( k \in S_1^* \). Suppose there exists a \( k \in S_1^* \) such that \( f^*(S_1^*, k) \not\supseteq \{1, \ldots, m_2\} \). Since \( f^*(S_1^*, k) \) is revenue-ordered, it must be \( f^*(S_1^*, k) \not\supseteq \{1, \ldots, m_2\} \). Now we construct a new solution \((x', y')\) by changing \( f^*(S_1^*, k) \) to \{1, ..., m_2\} while keeping all other assortments the same. Since \( k \in S_1^* \), according to Lemma 2, we have \( r_k^* \geq R^* \). Furthermore, as \( m \) is the smallest value that make \( \{1, ..., m_2\} \) optimal for stage 2 and \( f^*(S_1^*, k) \not\subseteq \{1, ..., m_2\} \), we know that \( f^*(S_1^*, k) \) is not optimal for stage 2, thus

\[
r_k' = p_k^{(1)} + \sum_{j=0}^{m_2} e^{\mu_{kj}} y_{kj}^* p_j^{(2)} > p_k^{(1)} + \sum_{j=0}^{m_2} e^{\mu_{kj}} y_{kj}^* p_j^{(2)} = r_k^* \geq R^*.
\]
According to (1) and (2), the seller’s expected two-stage revenue under \((x', y')\) is
\[
\begin{align*}
\left( \sum_{i \neq k} e^{(1)}_{i} + \gamma V_{i} x_{i} r_{k} \right) + e^{(1)}_{k} + \gamma V_{k} r_{k} & = \left( \sum_{i \neq k} e^{(1)}_{i} + \gamma V_{i} x_{i} r_{k} \right) + e^{(1)}_{k} + \gamma V_{k} r_{k} \\
\left( \sum_{i \neq k} e^{(1)}_{i} + \gamma V_{i} x_{i} r_{k} \right) & > \left( \sum_{i \neq k} e^{(1)}_{i} + \gamma V_{i} x_{i} r_{k} \right) + e^{(1)}_{k} + \gamma V_{k} r_{k} \\
\left( \sum_{i \neq k} e^{(1)}_{i} + \gamma V_{i} x_{i} r_{k} \right) & = \left( \sum_{i \neq k} e^{(1)}_{i} + \gamma V_{i} x_{i} r_{k} \right) + e^{(1)}_{k} + \gamma V_{k} r_{k} \geq R^*,
\end{align*}
\]
which contradicts the fact that \((x^*, y^*)\) is optimal. Therefore, Proposition 4.6 is established.

\(\square\)

E. Proof of Proposition 4.8

We prove Proposition 4.8 by establishing a polynomial-time reduction from the partition problem (a well-known NP-complete problem) to (the decision version of) problem (2) for any fixed rational \(\gamma \in (0, 1)\). The proof builds on a related but different reduction studied in Davis et al. (2014).

**Proof of Proposition 4.8.** Fix any rational \(\gamma \in (0, 1)\). The decision version of the two-stage assortment optimization problem, which we call the two-stage assortment feasibility problem, asks the following question: given two sets of products (for two stages) and a revenue threshold \(\tau\), does there exist a two-stage assortment policy that achieves an expected revenue of at least \(\tau\)?

The partition problem is the task of deciding whether a given multiset \(A\) of positive integers can be partitioned into two subsets such that the sum of elements in both subsets is the same. For any instance of the partition problem with \(A = \{a_1, \ldots, a_n\}\) and \(\sum_{j=1}^{n} a_j = 2A\), we construct an instance of the two-stage assortment feasibility problem, such that any efficient algorithm for the latter implies an efficient algorithm for the former. The two-stage assortment feasibility instance that we construct is as follows. In stage 1, there is only one product with \(p_1^{(1)} = 3(2A + 1)/(2\gamma) - A\) and \(\mu_1^{(1)} = \log ((2A + 1)/(2A + 1))\). In stage 2, there are \(n\) products with \(p_1^{(2)} = p_2^{(2)} = \cdots = p_n^{(2)} = \mu_2^{(2)} = (A + 1)(2A + 1)\) and \(\mu_2^{(2)} = \log (a_j)\) for all \(j = 1, \ldots, n\). The revenue threshold is \(\tau = (2A + 1)(A + 1/\gamma)\).

Let \(R^*\) be the optimal expected revenue of the constructed instance. In what follows, we analyze \(R^*\). Let \((S^*_1, f^*)\) be an optimal two-stage assortment policy that achieves \(R^*\) and satisfies Proposition 4.6. We first determine \(S^*_1\). If we do not offer any product in stage 1, then the best thing we can do is to offer \(\{1, \ldots, n\}\) in stage 2, and the expected overall revenue is \(\frac{2A}{1+2A} (A + 1)(2A + 1) = 2A(A + 1)\). However, by letting \(S_1 = \{1\}\) and \(S_2 = \{2, \ldots, n\}\), we can still obtain the same revenue in stage 2, while enjoying positive revenue in stage 1. Therefore, \(\emptyset\) is not optimal for \(S_1\) and thus \(S_1 = \{1\}\). We then determine \(f^*(S^*_1, 1)\). Since \(\{1, \ldots, n\}\) is the single-stage optimal assortment for stage 2, according to Proposition 4.6, we have \(f^*(\{1\}, 1) = \{1, \ldots, n\}\). We now try to express \(R^*\). For any \(S \subseteq \{1, \ldots, n\}\), let \(R_S^*\) denote the expected two-stage revenue of the following policy: \(S_1 = \{1\}, f(\{1\}, 1) = \{1, \ldots, n\}, f(\{1\}, 0) = S\). Then we have \(R^* = \max_{S \subseteq 1, \ldots, n} R_S^*\) and
\[
R_S^* = \left(1 + \sum_{j \in S} a_j\right)^{\gamma} \left(\frac{\sum_{j \in S} a_j \gamma V_j x_j r_{j}}{1 + \sum_{j \in S} a_j} (A + 1)(2A + 1) + 2(1 + A)^{\gamma} \left(\frac{3(2A + 1)}{2\gamma} - A + 2A(A + 1)\right)\right)
\]

In this construction, \(\mu_1^{(1)}\) and \(\mu_2^{(2)}\) may be irrational numbers (which cannot be represented in binary). The representation issue of irrational numbers can be addressed by introducing small perturbations to make the numbers rational and representable by polynomially many bits (which is a standard technique). Because of the discrete value setting of the partition problem, the reduction still holds after the perturbations.
= (2A + 1) \left( 1 + \sum_{j \in S} a_j \right)^{\gamma - 1} \left( \sum_{j \in S} a_j \right) (A + 1) + 2(1 + A)^\gamma \left( \frac{3}{2\gamma} + A \right).

For the constructed instance, the answer to the two-stage assortment feasibility question is “Yes” if and only if \( R^* = \max_{S \subset \{1, \ldots, n\}} R^*_S \geq (2A + 1)(1 + 1/\gamma) \). Let \( \lambda_S \) denote \((1 + A)/(1 + \sum_{j \in S} a_j)\). By simple algebra, for any \( S \subseteq \{1, \ldots, n\} \), we have

\[
R^*_S = (2A + 1) \left( 1 + A - \lambda_S \right) + 2\lambda_S \left( \frac{3}{2\gamma} + A \right) \frac{1 + 2\lambda^2_S}{1 + 2\gamma^2_S} = (2A + 1)(A + 1/\gamma) + 2A + 1 \frac{1 + 2\lambda^2_S}{1 + 2\gamma^2_S} \cdot 0 \quad \text{(equality holds if and only if} \lambda_S = 1),
\]

where the last inequality follows from the Bernoulli's inequality: \((1 + x)^\alpha \leq 1 + \alpha x\) for any \( \alpha \in (0, 1) \) and \( x > -1 \) (equality holds if and only if \( x = 0 \)). Specifically, since \( \gamma \in (0, 1) \), we have

\[
\lambda^2_S = (1 + (\lambda_S - 1))^\gamma \leq 1 + \gamma (\lambda_S - 1) \quad \text{(equality holds if and only if} \lambda_S = 1),
\]

which immediately implies (17). The equality condition of (17) tells us that \( R^* \geq (2A + 1)(A + 1/\gamma) \) if and only if there exists \( S \subseteq \{1, \ldots, n\} \) such that \( \lambda_S = 1 \), i.e., \( \sum_{j \in S} a_j = A = \sum_{j \in \{1, \ldots, n\} \setminus S} a_j \). Therefore, the ability to solve the two-stage assortment feasibility problem implies the ability to solve the partition problem, which is however NP-complete. \( \square \)

**F. Proof of Theorem 4.4**

**Proof of Theorem 4.4.** Let \((x^{opt}, y^{opt})\) be an optimal solution to problem (2). Let \((x^{alg}, y^{alg})\) be the solution returned by Algorithm 2. For any feasible \((x, y) \in F\), let \( R(x, y) \) denote the expected revenue obtained by \((x, y)\) (i.e., the objective value of problem (2)). We have \( R^* = \max_{(x, y) \in F} R(x, y) \).

Define \( F' = F \cap \{(x, y) \mid y_{0j} = 0, \forall j \in \mathcal{N}_2 \cup \{0\}\} \) and \( F'' = F \cap \{(x, y) \mid y_{0j} = \mathbb{1}_{(j \leq m_2)}, \forall j \in \mathcal{N}_2 \cup \{0\}\} \), where \( m_2 \) is defined in Line 1 of Algorithm 2 and \( \mathbb{1}_{(j \leq m_2)} \) is the indicator function. Intuitively, \( F' \) represents the set of policies that always set \( f(S_1, 0) = \emptyset \) and \( F'' \) represents the set of policies that always set \( f(S_1, 0) = \{1, \ldots, m_2\} \). By Lines 8 to 12 of Algorithm 2, we have \((x^{alg}, y^{alg}) \in F' \cup F'' \) and \( R(x^{alg}, y^{alg}) = \max_{(x, y) \in F' \cup F''} R(x, y) \).

In what follows, we prove \( R^* = R(x^{opt}, y^{opt}) \leq 2R(x^{alg}, y^{alg}) \), which implies Theorem 4.4.

For notational simplicity, we define

\[
V_i(y) = \log \left( \sum_{j=0}^{n_2} e^{\mu^{(2)}(y)} y_{ij} \right), \quad r_i(y) = p_i^{(1)} + \sum_{j=0}^{n_2} \frac{e^{\mu^{(2)}(y)} y_{ij}}{\sum_{j=0}^{n_2} e^{\mu^{(2)}(y)} y_{ij}} p_i^{(2)}, \quad i \in \mathcal{N}_1 \cup \{0\}.
\]

Then we have

\[
R(x, y) = \frac{\sum_{i=0}^{n_1} e_i^{(1)} + \gamma V_i(y) - r_i(y)}{\sum_{i=0}^{n_1} e_i^{(1)} + \gamma V_i(y) - r_i(y)} e^{\gamma V_0(y)} r_0(y) + \sum_{i=1}^{n_1} e_i^{(1)} + \gamma V_i(y) - r_i(y) e^{\gamma V_0(y)} + \sum_{i=1}^{n_1} e_i^{(1)} + \gamma V_i(y) - r_i(y) e^{\gamma V_0(y)}.
\]

Hence

\[
R(x^{opt}, y^{opt}) = \frac{e^{\gamma V_0(y^{opt})} r_0(y^{opt}) + \sum_{i=1}^{n_1} e_i^{(1)} + \gamma V_i(y^{opt}) - r_i(y^{opt}) e^{\gamma V_0(y^{opt})}}{e^{\gamma V_0(y^{opt})} + \sum_{i=1}^{n_1} e_i^{(1)} + \gamma V_i(y^{opt}) - r_i(y^{opt}) e^{\gamma V_0(y^{opt})}} = \frac{e^{\gamma V_0(y^{opt})} r_0(y^{opt}) + \sum_{i=1}^{n_1} e_i^{(1)} + \gamma V_i(y^{opt}) - r_i(y^{opt}) e^{\gamma V_0(y^{opt})}}{e^{\gamma V_0(y^{opt})} + \sum_{i=1}^{n_1} e_i^{(1)} + \gamma V_i(y^{opt}) - r_i(y^{opt}) e^{\gamma V_0(y^{opt})}}.
\]
where the last inequality follows from $V_0(y^{opt}) \geq \log(e^{\mu_0(2)}) = \mu_0(2)$.

Let $x' = x^{opt}$ and $y' = 1_{i \neq 0}b_{ij}$ for all $i \in \mathcal{N}_1 \cup \{0\}, j \in \mathcal{N}_2 \cup \{0\}$. The policy represented by $(x', y')$ has the following two properties: (i) it always provides the same $S_1$ and $f(S_1, 1), \ldots, f(S_1, |S_1|)$ as $(x^{opt}, y^{opt})$ provides and (ii) it restricts $f(S_1, 0)$ to be $0$. Thus we have $(x', y') \in F'$ and

\[
R(x', y') = \frac{e^{V_0(y')}r_0(y') + \sum_{i=1}^{n_1} e^{\mu_i(1) + \gamma V_i(y^{opt})} x'^{opt}_i r_i(y^{opt})}{e^{V_0(y')} + \sum_{i=1}^{n_1} e^{\mu_i(1) + \gamma V_i(y^{opt})} x'^{opt}_i} = \frac{\sum_{i=1}^{n_1} e^{\mu_i(1) + \gamma V_i(y^{opt})} x'^{opt}_i r_i(y^{opt})}{e^{\mu_0(2)} + \sum_{i=1}^{n_1} e^{\mu_i(1) + \gamma V_i(y^{opt})} x'^{opt}_i} \tag{19}
\]

where the last equality follows from $r_0(y') = p_0^{(1)} + p_0^{(2)} = 0$ and $V_0(y') = \log(e^{\mu_0(2)}) = \mu_0(2)$.

Let $x'' = 1_{i \neq 0}$ and $y'' = 1_{i \neq m_2}$ for all $i \in \mathcal{N}_1 \cup \{0\}, j \in \mathcal{N}_2 \cup \{0\}$. The policy represented by $(x'', y'')$ always provides $0$ in stage 1 and $\{1, \ldots, m_2\}$ in stage 2. Thus we have $(x'', y'') \in F''$ and

\[
R(x'', y'') = r_0(y'') = \sum_{j=0}^{m_2} \sum_{j=0}^{m_2} e^{\mu_i(1)} p_{2j}^{(2)} = \sum_{j=0}^{m_2} e^{\gamma V_i(0)} y_{ij} p_{2j}^{(2)} = r_0(y^{opt}), \tag{20}
\]

where the inequality follows from the single-stage optimality of $\{1, \ldots, m_2\}$ for stage 2.

Combining (18), (19) and (20), we have

\[
R(x^{opt}, y^{opt}) \leq R(x', y') + R(x'', y'') \leq 2 \max_{(x, y) \in F' \cup F''} R(x, y) = 2R(x^{alg}, y^{alg}),
\]

where the inequality follows from $(x', y') \in F'$ and $(x'', y'') \in F''$. We thus finish the proof. \hfill \Box

### G. Proofs of Results in Section 5

#### Proof of Proposition 5.1

We first prove the necessity. Let $\tilde{S}^* = \{(i, j)|i \in \{1, \ldots, m_1\}, j \in \{1, \ldots, m_2\}\}$ be the assortment of paths that corresponds to $\{(1, \ldots, m_1), \{1, \ldots, m_2\}\}$. We first consider the case when $m_2 < n_2$. Since $\{(1, \ldots, m_1), \{1, \ldots, m_2\}\}$ is optimal, removing $0, m_2$ from $\tilde{S}^*$ or adding $(1, m_2 + 1)$ into $\tilde{S}^*$ should not yield a strictly larger expected revenue. So according to part 3 and part 2\textsuperscript{14} of Lemma 3, $p_{2m_2}^{(2)} = \tilde{p}_{0m_2} \geq R_1^* + R_2^* \geq \tilde{p}_{1(m_2+1)} = p_1^{(1)} + p_{2m_2+1}^{(2)}$. When $m_2 = n_2$, the argument for $p_{2n_2}^{(2)} \geq R_1^* + R_2^*$ still holds.

We then prove the sufficiency. Let $\hat{S} = \{(i, j)|i \in \{1, \ldots, m_1\}, j \in \{1, \ldots, m_2\}\}$ be the assortment of paths that corresponds to $\{(1, \ldots, m_1), \{1, \ldots, m_2\}\}$. We first consider the case when $m_2 < n_2$. Since $\hat{p}_{1(m_2+1)} = p_1^{(1)} + p_{2m_2+1}^{(2)} \leq R_1^* + R_2^*$, we have $\hat{p}_{ij} \leq R_1^* + R_2^*$ for all $i \in \{1, \ldots, m_1\}, j \in \{1 + 1, \ldots, n_1\}$. Since $\tilde{p}_{0m_2} = p_{2m_2}^{(2)} \leq R_1^* + R_2^*$, we have $\hat{p}_{ij} \geq R_1^* + R_2^*$ for all $j \in \{1, \ldots, m_1\}$. By the definition of $m_1$, we have $p_1^{(1)} + R_2^* \geq R_1^* + R_2^*$ for all $i \in \{1, \ldots, m_1\}$. According to Theorem 4.3 and Proposition 4.6, we know that we can always obtain an optimal solution to the two-stage assortment optimization problem through some adjustments of $\tilde{S}$ that can be expressed in the

\[14\text{ Strictly speaking, part 2 is the “larger than or equal to” version, not the “strictly larger than” version that we need here. But this does not pose any difficulty as the proofs of these two versions are basically the same.} \]
following three-step form: 1. adding some proper paths \((i, j)\) into \(\tilde{S}\) \((i \in \{1, \ldots, m_1\}, j \in \{m_1 + 1, \ldots, n_1\})\), 2. removing some improper paths from \(\tilde{S}\) \((i \in \{1, \ldots, m_1\}, j \in \{1, \ldots, m_1\})\), and 3. removing some improper nests of paths \(\{(i, 1), \ldots, (i, m_2)\}\) from \(\tilde{S}\) \((i \in \{1, \ldots, m_1\})\). However, since \(\tilde{p}_{ij} \leq R^*_1 + R^*_2\) for all \(i \in \{1, \ldots, m_1\}, j \in \{m_1 + 1, \ldots, n_1\}\), \(\tilde{p}_{ij} \geq R^*_1 + R^*_2\) for all \(j \in \{1, \ldots, m_1\}\) and \((p^{(1)}_i + R^*_2) \geq R^*_1 + R^*_2\) for all \(i \in \{1, \ldots, m_1\}\), all of these three types of adjustments just make the revenue smaller than or equal to \(R^*_1 + R^*_2\). As a result, \(\tilde{S}\) itself must be optimal, thus \([\{1, \ldots, m_1\}, \{1, \ldots, m_2\}]\) is optimal.

When \(m_2 = n_2\), we can use the same arguments except for those dealing with paths \((i, j)\) where \(i \in \{1, \ldots, m_1\}, j \in \{m_1 + 1, \ldots, n_1\}\) to obtain our result. \(\square\)

**Proof of Proposition 5.2.** Let \(\tilde{S}^* = \{(i, j) | i \in \{1, \ldots, m_1\}, j \in \{1, \ldots, m_2\}\}\) be the assortment of paths that corresponds to \([\{1, \ldots, m_1\}, \{1, \ldots, m_2\}]\). Since \([\{1, \ldots, m_1\}, \{1, \ldots, m_2\}]\) is optimal, removing \((m_1, m_2)\) from \(\tilde{S}^*\) or adding \((1, m_2 + 1)\) into \(\tilde{S}^*\) should not yield a strictly larger expected revenue. So according to part 1 and 2 of Lemma 3, we have \(p^{(1)}_{m_1} + p^{(2)}_{m_2} = \tilde{p}_{m_1 m_2} \geq \gamma(R^*_1 + R^*_2) + \gamma(\gamma p^{(1)}_1 + R^*_2)\) for all \(j \in \{1, \ldots, m_1\}\) and \(p^{(1)}_{m_1} + p^{(2)}_{m_2 + 1} = \tilde{p}_{1(m_2 + 1)} \leq \gamma(R^*_1 + R^*_2) + (1 - \gamma)(\gamma p^{(1)}_1 + R^*_2)\). Thus \(\gamma p^{(1)}_{m_1} + p^{(2)}_{m_2} \geq \gamma R^*_1 + R^*_2 \geq \gamma p^{(1)}_1 + p^{(2)}_{m_2 + 1}\). \(\square\)

### H. Supplements for Section 6

#### H.1. Proof of Theorem 6.2 and Theorem 6.3

To begin with, we reformulate the generalized two-stage assortment optimization problem in the forms of problem (2) and problem (9).

Let \(x_i \in \{0, 1\}\) denote whether product \(i \in N_1\) is included in \(S_1\) in stage 1, and let \(y_{ij} \in \{0, 1\}\) denote whether product \(i(j) \in N_2(i)\) is included in \(f(S_1, i)\) in stage 2. Similar to what we do in Section 4, we obtain the following constraints:

\[
F = \{(x, y) \mid x_0 = 1, y_{ij} \leq x_i, y_{0o} = x_i, x_i, y_{ij} \in \{0, 1\}, \forall i \in N_1 \cup \{0\}, j \in N_2 \cup \{0\}\},
\]

and we can formulate the generalized two-stage assortment optimization problem as the following optimization problem:

\[
\max_{(x, y) \in F} \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} P_{ij} \left( p^{(1)}_i + p^{(2)}_{i(j)} \right). \tag{21}
\]

Similar to what we do in Section 4.2 and Section 4.3, for \(0 < \gamma \leq 1\), we view the generalized two-stage problem from the perspective of determining an assortment of paths. Let \((i, j)\) denote the customer’s action of “first choosing \(i\) then choosing \(i(j)\)”, and refer to it as her two-stage choice path. Let \(\mu_{ij} = \mu^{(1)}_i / \gamma + \mu^{(2)}_{i(j)}\) be the total expected utility of \((i, j)\), and \(\tilde{p}_{ij} = p^{(1)}_i + p^{(2)}_{i(j)}\) be the total revenue of \((i, j)\). Then \(\tilde{x}_{ij} = x_i y_{ij}\) can be viewed as an indicator variable of the availability of \((i, j)\), and we can reformulate problem (21) as:

\[
\max_{x} \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} \left( \frac{\sum_{j=0}^{n_2} e^{\tilde{p}_{ij}} \tilde{x}_{ij}}{\sum_{i=0}^{n_1} \sum_{j=0}^{n_2} e^{\tilde{p}_{ij}} \tilde{x}_{ij}} \right) y_{ij} \tag{22}
\]

s.t.
\[
x_{00} = 1,
\]
\[
x_{ij} \leq \tilde{x}_{00}, \quad \forall i = 0, \ldots, n_1, j = 0, \ldots, n_2,
\]
\[
\tilde{x}_{ij} \in \{0, 1\}, \quad \forall i = 0, \ldots, n_1, j = 0, \ldots, n_2.
\]

It is worth noting that problem (22) “looks” exactly the same as problem (9). However, there is also an essential difference between them, hidden in their parameters:

- In problem (9), there must be \(\tilde{p}_{kj} \geq p^{(1)}_{ij}\) when \(1 \leq k \leq i\). (The reason is that \(\tilde{p}_{kj} = p^{(1)}_k + p^{(2)}_{k} \geq p^{(1)}_i + p^{(2)}_j = p^{(2)}_{ij}\).

Electronic copy available at: https://ssrn.com/abstract=3243742
In problem (22), there is not necessarily such a relationship between \( \tilde{p}_{kj} \) and \( \tilde{p}_{ij} \) when \( 1 \leq k \leq i \).
(The reason is that, while \( \tilde{p}_{kj} = p_{k}^{(1)} + p_{k(j)}^{(2)} \) and \( \tilde{p}_{ij} = p_{i}^{(1)} + p_{i(j)}^{(2)} \), this time, \( k(j) \) and \( i(j) \) are two different products with different prices.)

The strong similarity and subtle difference between problem (9) and (22) remind us of that, a result that is true for problem (9) usually remains true for problem (22), unless it relies on the inequality "\( \tilde{p}_{kj} \geq \tilde{p}_{ij} \) when \( 1 \leq k \leq i \)." Based on this idea, we obtain the following proposition.

**Proposition H.1.** For problem (22), there exists an optimal solution \( \tilde{x}^* \) satisfying the following properties:

1. For any \( i \in \mathcal{N}_1 \), if \( \tilde{x}_{io} = 1 \), then for any \( 0 \leq k \leq j \), \( \tilde{x}_{ik}^* = 1 \);
2. For any \( i \in \mathcal{N}_1 \cup \{0\} \) and \( j \in \mathcal{N}_2 \cup \{0\} \), if \( \tilde{x}_{ij} = 1 \), then for any \( 1 \leq k \leq i \), \( \tilde{x}_{ik}^* = 1 \).

If \( \gamma = 1 \), then there is an additional property:
3. (When \( \gamma = 1 \)) For any \( j \in \mathcal{N}_2 \cup \{0\} \), if \( \tilde{x}_{0j} = 1 \), then for any \( 0 \leq k \leq j \), \( \tilde{x}_{0k}^* = 1 \).

**Proof of Proposition H.1.** Go through the proofs of Lemma 2, Lemma 3 and Proposition C.1 in Appendix C. It can be verified that Lemma 2, Lemma 3, “part 1 of Proposition C.1”, “case 1 of part 2 of Proposition C.1” and “part 3 of Proposition C.1” still hold true for problem (22), as their proofs do not involve with the inequality “\( \tilde{p}_{kj} \geq \tilde{p}_{ij} \) when \( 1 \leq k \leq i \).” However, “case 2 of part 2 of Proposition C.1” does not hold true for problem (22), as its proof involves with the inequality “\( \tilde{p}_{kj} \geq \tilde{p}_{ij} \) when \( 1 \leq k \leq i \).” Summarizing “part 1 of Proposition C.1”, “case 1 of part 2 of Proposition C.1” and “part 3 of Proposition C.1” for problem (22) leads to Proposition H.1. □

Finally, we would like to point out that Theorem 6.2 and Theorem 6.3 are both corollaries that directly follow Proposition H.1. Therefore, Theorem 6.2 and Theorem 6.3 are established.

**H.2. Proof of Proposition 6.1**
Let \( (x^*, y^*) \) be an optimal solution to problem (21) that satisfies the properties of Theorem 6.3 (if \( 0 < \gamma < 1 \)), Theorem 6.2 (if \( \gamma = 1 \)) or Theorem 6.1 (if \( \gamma = 0 \)). Then the corresponding \( S_1^* \) and \( f^*(S_1^*, i) \) \( (i \in S_1^*) \) are all revenue-ordered. Let \( R^* \) be the corresponding expected overall revenue. For notational simplicity, we define

\[
V_i^* = \log \left( \sum_{j=0}^{n_2} e^{\mu(i(j))} y_{ij}^* \right), \quad r_i^* = p_i^{(1)} + \sum_{j=0}^{n_2} \frac{e^{\mu(i(j))} y_{ij}^*}{\sum_{j=0}^{n_2} e^{\mu(i(j))} y_{ij}^*} p_{i(j)}^{(2)}, \quad i \in \mathcal{N}_1 \cup \{0\}.
\]

\[
V_i^* = \log \left( \sum_{j=0}^{m(i)} e^{\mu(i(j))} y_{ij}^* \right), \quad r_i^* = p_i^{(1)} + \sum_{j=0}^{m(i)} \frac{e^{\mu(i(j))} y_{ij}^*}{\sum_{j=0}^{m(i)} e^{\mu(i(j))} y_{ij}^*} p_{i(j)}^{(2)}, \quad i \in \mathcal{N}_1 \cup \{0\}.
\]

Suppose there exists a \( k \in S_1^* \) such that \( f^*(S_1^*, k) \not\subseteq \{k(1), \ldots, k(m(k))\} \). Since \( f^*(S_1^*, k) \) is revenue-ordered, it must be \( f^*(S_1^*, k) \subseteq \{k(1), \ldots, k(m(k))\} \). Now we construct a new solution \( (x', y') \) by changing \( f^*(S_1^*, k) \) to \( \{k(1), \ldots, k(m(k))\} \) while keeping all other assortments the same. Since \( k \in S_1^* \), according to Lemma 2, we have \( r_k' \geq R^* \). Furthermore, as \( m \) is the smallest value that make \( \{k(1), \ldots, k(m(k))\} \) optimal for stage 2 and \( f^*(S_1^*, k) \subseteq \{k(1), \ldots, k(m(k))\} \), we know that \( f^*(S_1^*, k) \) is not optimal for stage 2, thus

\[
r_k' = p_k^{(1)} + \sum_{j=0}^{m(k)} \frac{e^{\mu(k(j))} y_{kj}}{\sum_{j=0}^{m(k)} e^{\mu(k(j))} y_{kj}} p_{k(j)}^{(2)} > p_k^{(1)} + \sum_{j=0}^{m(k)} \frac{e^{\mu(k(j))} y_{kj}}{\sum_{j=0}^{m(k)} e^{\mu(k(j))} y_{kj}} p_{k(j)}^{(2)} = r_k' \geq R^*.
\]

According to (2) and (3), the seller’s expected two-stage revenue under \( (x', y') \) is

\[
\frac{\sum_{i \neq k} e^{\mu(i)} y_{i}^* r_k^* + e^{\mu(k)} y_{k}^* r_k^*}{\sum_{i \neq k} e^{\mu(i)} y_{i}^* + e^{\mu(k)} y_{k}^*} = \frac{\sum_{i} e^{\mu(i) + \gamma V_i^*} y_{i}^* r_k^* + e^{\mu(k) + \gamma V_k^*} r_k^*}{\sum_{i} e^{\mu(i) + \gamma V_i^*} + e^{\mu(k) + \gamma V_k^*}} = \frac{e^{\mu(k) + \gamma V_k^*} r_k^*}{\sum_{i} e^{\mu(i) + \gamma V_i^*} + e^{\mu(k) + \gamma V_k^*}}.
\]
H.3. Modified Algorithms in Section 6

The modified Merged Path Selection algorithm is presented in Algorithm 3. The modified Nested Merged Path Selection algorithm is presented in Algorithm 4.

Algorithm 3 modified Merged Path Selection (m-MPS)

<table>
<thead>
<tr>
<th>Input: $\mu_i^{(1)}, p_i^{(1)}, i = 0, \ldots, n_1$ and $\mu_i^{(2)}, p_i^{(2)}, i = 0, \ldots, n_1, j = 0, \ldots, n_2$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: for $i = 0, 1, \ldots, n_1$ do</td>
</tr>
<tr>
<td>2: Find the “smallest” optimal single-stage assortment for stage 2: ${i(1), i(2), \ldots, i(m(i))}$.</td>
</tr>
<tr>
<td>3: for $j = 0, 1, \ldots, n_2$ do</td>
</tr>
<tr>
<td>4: Compute $\tilde{\mu}<em>{ij} = \mu_i^{(1)} + p</em>{i(j)}^{(2)}$ and $\tilde{p}<em>{ij} = p_i^{(1)} + p</em>{i(j)}^{(2)}$.</td>
</tr>
<tr>
<td>5: if $i \neq 0$ then</td>
</tr>
<tr>
<td>6: Merge paths $(i, 0), (i, 1), \ldots, (i, m(i))$ into a new path $(i, \sim (i))$ with $\tilde{\mu}<em>{i,\sim (i)} = \log\left(\sum</em>{j=0}^{m(i)} e^{\tilde{\mu}<em>{ij}}\right)$ and $\tilde{p}</em>{i,\sim (i)} = \sum_{j=0}^{m(i)} e^{\tilde{\mu}<em>{ij}}/\sum</em>{j=0}^{m(i)} e^{\tilde{p}_{ij}}$.</td>
</tr>
<tr>
<td>7: Solve problem (7) to find an optimal assortment $\tilde{S}^*$.</td>
</tr>
<tr>
<td>8: (For problem (22)) Recover the corresponding $\tilde{x}^<em>$ from $\tilde{S}^</em>$.</td>
</tr>
<tr>
<td>9: (For problem (21)) Recover the corresponding $(x^<em>, y^</em>)$ from $\tilde{x}^*$.</td>
</tr>
<tr>
<td>10: return $(x^<em>, y^</em>)$.</td>
</tr>
</tbody>
</table>

Algorithm 4 modified Nested Merged Path Selection (m-NMPS)

<table>
<thead>
<tr>
<th>Input: $\gamma, \mu_i^{(1)}, p_i^{(1)}, i = 0, \ldots, n_1$ and $\mu_i^{(2)}, p_i^{(2)}, i = 0, \ldots, n_1, j = 0, \ldots, n_2$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: for $i = 0, 1, \ldots, n_1$ do</td>
</tr>
<tr>
<td>2: Find the “smallest” optimal single-stage assortment for stage 2: ${1, 2, \ldots, m(i)}$.</td>
</tr>
<tr>
<td>3: for $j = 0, 1, \ldots, n_2$ do</td>
</tr>
<tr>
<td>4: Compute $\tilde{\mu}<em>{ij} = \mu_i^{(1)}/\gamma + \mu</em>{i(j)}^{(2)}$ and $\tilde{p}<em>{ij} = p_i^{(1)} + p</em>{i(j)}^{(2)}$.</td>
</tr>
<tr>
<td>5: if $i \neq 0$ then</td>
</tr>
<tr>
<td>6: Merge paths $(i, 0), (i, 1), \ldots, (i, m(i))$ into a new path $(i, \sim (i))$ with $\tilde{\mu}<em>{i,\sim (i)} = \log\left(\sum</em>{j=0}^{m(i)} e^{\tilde{\mu}<em>{ij}}\right)$ and $\tilde{p}</em>{i,\sim (i)} = \sum_{j=0}^{m(i)} e^{\tilde{\mu}<em>{ij}}/\sum</em>{j=0}^{m(i)} e^{\tilde{p}_{ij}}$.</td>
</tr>
<tr>
<td>7: Let $\tilde{S}^*$ be $\emptyset$.</td>
</tr>
<tr>
<td>8: for $S \in {\emptyset, {1, \ldots, m(i)}}$ do</td>
</tr>
<tr>
<td>9: Fix $f(S_i, 0) = S$ and formulate problem (12).</td>
</tr>
<tr>
<td>10: Solve problem (12) to find an optimal assortment and construct a solution to problem (11).</td>
</tr>
<tr>
<td>11: if the new solution is better than $\tilde{S}^*$ then</td>
</tr>
<tr>
<td>12: Change $\tilde{S}^*$ to the new solution.</td>
</tr>
<tr>
<td>13: (For problem (22)) Recover the corresponding $\tilde{x}^<em>$ from $\tilde{S}^</em>$.</td>
</tr>
<tr>
<td>14: (For problem (21)) Recover the corresponding $(x^<em>, y^</em>)$ from $\tilde{x}^*$.</td>
</tr>
<tr>
<td>15: return $(x^<em>, y^</em>)$</td>
</tr>
</tbody>
</table>
I. Supplements for Appendix A

I.1. Proof of Proposition A.1 and Proposition A.2

Proof of Proposition A.1. Given the seller’s policy \( \Pi \), we can calculate the following value functions via backward induction (see Section 3):

\[
U^\Pi_T(h_s^T, h_c^T) = \log \left( \sum_{i \in \Theta_T^\Pi(h_s^T \cup \{0\})} \exp(\mu_i^T) \right), \quad \forall h_s^T, h_c^T.
\]

\[
U^\Pi_{T-1}(h_s^{T-1}, h_c^{T-1}) = \log \left( \sum_{i \in \Theta_{T-1}^\Pi(h_c^{T-1} \cup \{0\})} \exp \left( \mu_i^{T-1} + \log \left( \sum_{j \in \Theta_j^\Pi(h_s^{T-1} \cup \{0\})} \exp(\mu_j^T) \right) \right) \right), \quad \forall h_s^{T-1}, h_c^{T-1}.
\]

\[
\ldots \ldots
\]

\[
U^\Pi_2(h_1^2, h_2^2) = \log \left( \sum_{i_2 \in \Theta_2^\Pi(h_2^2 \cup \{0\})} \exp \left( \mu_i^{(2)} + \ldots + \mu_i^{T} \right) \right), \quad \forall h_1^2, h_2^2.
\]

\[
U^\Pi_1(\text{null}, \text{null}) = \log \left( \sum_{i_1 \in \Theta_1^\Pi \cup \{0\}, i_2 \in \Theta_2^\Pi(i_1 \cup \{0\})} \exp \left( \mu_i^{(1)} + \mu_i^{(2)} + \ldots + \mu_i^{T} \right) \right).
\]

We can then calculate the choice probabilities. The customer’s probability of “choosing \( i_1 \in \mathcal{N}_1 \cup \{0\} \) at stage 1” is

\[
P^\Pi_{i_1} = \frac{\exp \left( \mu_i^{(1)} + U_2(g_1^\Pi, i_1) \right)}{\sum_{i_1 = 0}^{n_1} \exp \left( \mu_i^{(1)} + U_2(g_1^\Pi, i_1) \right)} = \frac{\sum_{i_2=0}^{n_2} \sum_{i_T=0}^{n_T} e^{\beta_{i_1 i_2 \ldots i_T} x_{i_1 i_2 \ldots i_T}}}{\sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \sum_{i_T=0}^{n_T} e^{\beta_{i_1 i_2 \ldots i_T} x_{i_1 i_2 \ldots i_T}}},
\]

The customer’s probability of “choosing \( i_1 \in \mathcal{N}_1 \cup \{0\} \) at stage 1 and choosing \( i_2 \in \mathcal{N}_2 \cup \{0\} \) at stage 2” is

\[
P^\Pi_{i_1 i_2} = \frac{\exp \left( \mu_i^{(2)} + U_3(g_2^\Pi, g_2^\Pi(i_1)), (i_1, i_2) \right)}{\sum_{i_2=0}^{n_2} \exp \left( \mu_i^{(2)} + U_3(g_2^\Pi, g_2^\Pi(i_1)), (i_1, i_2) \right)} = \frac{\sum_{i_2=0}^{n_2} \sum_{i_T=0}^{n_T} e^{\beta_{i_1 i_2 \ldots i_T} x_{i_1 i_2 \ldots i_T}}}{\sum_{i_2=0}^{n_2} \sum_{i_T=0}^{n_T} e^{\beta_{i_1 i_2 \ldots i_T} x_{i_1 i_2 \ldots i_T}}}.
\]

Continuing the above procedure and we can finally obtain the customer’s probability of “choosing \( i_1 \in \mathcal{N}_1 \cup \{0\} \) at stage 1, \ldots, choosing \( i_T \in \mathcal{N}_T \cup \{0\} \)"

\[
P^\Pi_{i_1 \ldots i_T} := \frac{e^{\beta_{i_1 \ldots i_T} x_{i_1 \ldots i_T}}}{\sum_{i_1=0}^{n_1} \ldots \sum_{i_T=0}^{n_T} e^{\beta_{i_1 \ldots i_T} x_{i_1 \ldots i_T}}},
\]
Proof of Proposition A.2. It is evident that problem (16) is equivalent to problem (23) below.

\[
\begin{align*}
\max_{\bar{x}} & \sum_{i_1=0}^{n_1} \cdots \sum_{i_T=0}^{n_T} e^{\bar{p}_{i_1 \cdots i_T}} \bar{x}_{i_1 \cdots i_T} \bar{p}_{i_1 \cdots i_T} \\
\text{s.t.} & \quad \bar{x}_{0 \cdots 0} = 1 \\
& \quad \bar{x}_{i_1 \cdots i_{t-1} t \cdots i_T} \leq \bar{x}_{i_1 \cdots i_{t-1} 0 \cdots 0}, \quad \forall t \in \{1, \ldots, T\}, \forall i_1 \in \mathcal{N}_1 \cup \{0\}, \ldots, i_T \in \mathcal{N}_T \cup \{0\} \\
& \quad \bar{x}_{i_1 \cdots i_T} \in \{0, 1\}, \quad \forall i_1 \in \mathcal{N}_1 \cup \{0\}, \ldots, i_T \in \mathcal{N}_T \cup \{0\}.
\end{align*}
\] (23)

For any $T$-stage assortment policy $\Pi$, for all $i_1 \in \mathcal{N}_1 \cup \{0\}, \ldots, i_T \in \mathcal{N}_T \cup \{0\}$, if $\bar{x}_{i_1 \cdots i_T} = 1$, i.e.,

\[i_1 \in g^\Pi_1 \cup \{0\}, \ldots, i_T \in g^\Pi_T(i_1, \ldots, i_{T-1}) \cup \{0\},\]

then for all $t \in \{1, \ldots, T\}$, we have

\[i_1 \in g^\Pi_1 \cup \{0\}, \ldots, i_{t-1} \in g^\Pi_{T-1}(i_1, \ldots, i_{t-2} \cup \{0\}, 0 \in g^\Pi_1(i_1, \ldots, i_{t-2} \cup \{0\}, \ldots, 0 \in g^\Pi_T(i_1, \ldots, i_1, 0, \ldots, 0) \cup \{0\},\]

which means that $\bar{x}_{i_1 \cdots i_{t-1} 0 \cdots 0} = 1$. Therefore, $\bar{x}^\Pi$ must be a feasible solution to problem (23).

Meanwhile, for any feasible solution $\bar{x}$ to problem (23), we can always construct a valid $T$-stage assortment policy $\Pi$ such that $g^\Pi_1 = \{i_1 \in \mathcal{N}_1 \mid \bar{x}_{i_1 0 \cdots 0} = 1\}$ and $g^\Pi_1(i_1, \ldots, i_{t-1}) = \{i_t \in \mathcal{N}_t \mid \bar{x}_{i_1 \cdots i_{t-1} i_t 0 \cdots 0} = 1\}$ for all $t \in \{1, \ldots, T\}$. This construction of $\Pi$ ensures that $\bar{x}^\Pi = \bar{x}$.

Combining the above three paragraphs, we know that the $T$-stage assortment optimization problem is equivalent to problem (16). \qed

I.2. Proof of Theorem A.1

We claim and prove Proposition I.1 below, which is an equivalent version of Theorem A.1.

**Proposition I.1.** For problem (16), there exists an optimal solution $\bar{x}^*$ such that

1. For any $t = 1, \ldots, T$, for any $i_1 \in \mathcal{N}_1 \cup \{0\}, i_1 \in \mathcal{N}_1 \cup \{0\}$ such that $\bar{x}_{i_1 \cdots i_{t-1} i_t 0 \cdots 0} = 1$, we have $\bar{x}_{i_1 \cdots i_{t-1} i_t 0 \cdots 0} = 1$ for all $0 \leq k \leq i_t$.

2. For any $t = 2, \ldots, T$, for any $i_1 \in \mathcal{N}_1 \cup \{0\}, \ldots, i_t \in \mathcal{N}_t \cup \{0\}$ such that $\bar{x}_{i_1 \cdots i_{t-1} i_t 0 \cdots 0} = 1$ and $i_{t-1} \neq 0$, for any $i'_1 \in \mathcal{N}_1 \cup \{0\}, \ldots, i'_{t-1} \in \mathcal{N}_{t-1} \cup \{0\}$ such that $\bar{x}_{i_1 \cdots i_{t-1} i_t 0 \cdots 0} = 1, i'_t \neq 0$, and $p_{i_t}^{(1)} + \cdots + p_{i_{t-1}}^{(t-1)} = p_{i_t}^{(1)} + \cdots + p_{i_{t-1}}^{(t-1)}$, we have $\bar{x}_{i'_1 \cdots i'_{t-1} i_t 0 \cdots 0} = 1$.

In what follows, we provide a proof of Proposition I.1. We start with some definitions. For any feasible $\bar{x}$, we define the seller’s expected overall revenue as

\[R(\bar{x}) = \sum_{i_1=0}^{n_1} \cdots \sum_{i_T=0}^{n_T} e^{\bar{p}_{i_1 \cdots i_T}} \bar{x}_{i_1 \cdots i_T} \bar{p}_{i_1 \cdots i_T}.
\]

We also define

\[R_{i_1, \ldots, i_T}(\bar{x}) = \sum_{s=1}^{t} p_s^{(s)} + \sum_{t_{s+1}=0}^{n_{s+1}} \cdots \sum_{i_T=0}^{n_T} e^{\bar{p}_{i_1 \cdots i_T}} \bar{x}_{i_1 \cdots i_T} \bar{p}_{i_1 \cdots i_T}
\]
for all \( t = 1, \ldots, T \) and \( i_1 \in \mathcal{N}_1 \cup \{0\}, \ldots, i_t \in \mathcal{N}_t \cup \{0\} \). Here \( r_{i_1 \ldots i_t}^t(\tilde{x}) \) can be understood as the seller’s expected \( T \)-stage revenue under \( \tilde{x} \), given that the customer chooses the \( t \)-stage path \((i_1, \ldots, i_t)\) in the first \( t \) stages. We then know that for all feasible \( \tilde{x} \), for all \( t = 1, \ldots, T \),

\[
R(\tilde{x}) = \frac{\sum_{i_1=0}^{n_1} \cdots \sum_{i_t=0}^{n_t} \left( \sum_{i_{t+1}=0}^{n_{t+1}} \cdots \sum_{i_{T}=0}^{n_{T}} e^{\tilde{R}_{i_1 \cdots i_T}^t \tilde{x}_{i_1 \cdots i_T}} \right) r_{i_1 \cdots i_t}^t(\tilde{x})}{\sum_{i_1=0}^{n_1} \cdots \sum_{i_t=0}^{n_t} \left( \sum_{i_{t+1}=0}^{n_{t+1}} \cdots \sum_{i_{T}=0}^{n_{T}} e^{\tilde{R}_{i_1 \cdots i_T}^t \tilde{x}_{i_1 \cdots i_T}} \right)}.
\]

(24)

We have the following lemmas:

**Lemma 4.** Let \( \tilde{x}^* \) be an optimal solution to (16). For any \( t \in \{1, \ldots, T\} \), for any \( i_1 \in \mathcal{N}_1 \cup \{0\}, \ldots, i_{t-1} \in \mathcal{N}_{t-1} \cup \{0\}, \), if \( \tilde{x}_{i_1 \ldots i_{t-1}} = 0 \), then \( r_{i_1 \ldots i_{t-1}}^t(\tilde{x}^*) \geq R(\tilde{x}^*) \).

**Proof of Lemma 4.** We prove Lemma 4 by contradiction. Suppose there exist \( t \in \{1, \ldots, T\} \), \( j_1 \in \mathcal{N}_1 \cup \{0\}, \ldots, j_{t-1} \in \mathcal{N}_{t-1} \cup \{0\}, j_t \in \mathcal{N}_t \) such that \( \tilde{x}_{j_1 \ldots j_{t-1} j_t} = 1 \) and \( r_{j_1 \ldots j_t}(\tilde{x}^*) < R(\tilde{x}^*) \). Now we construct another solution \( \tilde{x}' \) by first letting \( \tilde{x}' = \tilde{x}^* \) then modifying \( \tilde{x}_{j_1 \ldots j_{t-1} j_t} \) to be 0 for all \( i_1 \in \mathcal{N} \cup \{0\}, \ldots, i_T \in \mathcal{N} \cup \{0\} \). By (24), we have

\[
R(\tilde{x}') = \frac{\sum_{i_1=0}^{n_1} \cdots \sum_{i_t=0}^{n_t} \left( \sum_{i_{t+1}=0}^{n_{t+1}} \cdots \sum_{i_{T}=0}^{n_{T}} e^{\tilde{R}_{i_1 \cdots i_T}^t \tilde{x}_{i_1 \cdots i_T}} \right) r_{i_1 \cdots i_t}^t(\tilde{x}')}{\sum_{i_1=0}^{n_1} \cdots \sum_{i_t=0}^{n_t} \left( \sum_{i_{t+1}=0}^{n_{t+1}} \cdots \sum_{i_{T}=0}^{n_{T}} e^{\tilde{R}_{i_1 \cdots i_T}^t \tilde{x}_{i_1 \cdots i_T}} \right)} \geq \frac{\sum_{i_1=0}^{n_1} \cdots \sum_{i_t=0}^{n_t} \left( \sum_{i_{t+1}=0}^{n_{t+1}} \cdots \sum_{i_{T}=0}^{n_{T}} e^{\tilde{R}_{i_1 \cdots i_T}^t \tilde{x}_{i_1 \cdots i_T}} \right) r_{i_1 \cdots i_t}^t(\tilde{x}^*)}{\sum_{i_1=0}^{n_1} \cdots \sum_{i_t=0}^{n_t} \left( \sum_{i_{t+1}=0}^{n_{t+1}} \cdots \sum_{i_{T}=0}^{n_{T}} e^{\tilde{R}_{i_1 \cdots i_T}^t \tilde{x}_{i_1 \cdots i_T}} \right)} = R(\tilde{x}'),
\]

where the inequality is because \( r_{j_1 \ldots j_t}(\tilde{x}') < R(\tilde{x}^*) \). However, this contradicts the assumption that \( \tilde{x}^* \) is optimal. Therefore, Lemma 4 is established.

**Lemma 5.** Let \( \tilde{x} \) be a feasible solution to (16) with \( \tilde{x}_{0,0} = 1 \), then \( \tilde{S}(\tilde{x}) = \{(i_1, \ldots, i_T) | \tilde{x}_{i_1 \cdots i_T} = 1\} \) is the corresponding assortment of offered paths (including \((0,0,0)\)). Assume there exists \( t \in \{1, \ldots, T\} \) and a path \((j_1, \ldots, j_{t-1}, j_t, 0, \ldots, 0) \notin \tilde{S}(\tilde{x}) \) such that \((j_1, \ldots, j_{t-1}, 0, 0, \ldots, 0) \in \tilde{S}(\tilde{x}) \) and \( j_t \neq 0 \). Consider another feasible solution \( \tilde{x}' \) satisfying

\[
\tilde{x}'_{j_1 \ldots j_t, 0, \ldots, 0} = 1,
\]

\[
\tilde{x}'_{i_1 \cdots i_T} = \tilde{x}_{i_1 \cdots i_T}, \ \forall (i_1, \ldots, i_T) \neq (j_1, \ldots, j_t), \forall i_{t+1}, \ldots, i_T.
\]

If \( r_{j_1 \ldots j_t}(\tilde{x}') > R(\tilde{x}) \), then \( r_{j_1 \ldots j_t}(\tilde{x}') > R(\tilde{x}) \); if \( r_{j_1 \ldots j_t}(\tilde{x}') = R(\tilde{x}) \), then \( r_{j_1 \ldots j_t}(\tilde{x}') = R(\tilde{x}) = R(\tilde{x}') \).

**Proof of Lemma 5.** By (24) and \( \tilde{x}_{j_1 \ldots j_t, 0, \ldots, 0} = 1, \tilde{x}_{j_1 \ldots j_t, 0, \ldots, 0} = 0 \), \( \tilde{x}'_{j_1 \ldots j_t, 0, \ldots, 0} = 1 \), we have

\[
R(\tilde{x}') = \frac{\sum_{i_1=0}^{n_1} \cdots \sum_{i_t=0}^{n_t} \left( \sum_{i_{t+1}=0}^{n_{t+1}} \cdots \sum_{i_{T}=0}^{n_{T}} e^{\tilde{R}_{i_1 \cdots i_T}^t \tilde{x}_{i_1 \cdots i_T}} \right) r_{i_1 \cdots i_t}^t(\tilde{x}')}{\sum_{i_1=0}^{n_1} \cdots \sum_{i_t=0}^{n_t} \left( \sum_{i_{t+1}=0}^{n_{t+1}} \cdots \sum_{i_{T}=0}^{n_{T}} e^{\tilde{R}_{i_1 \cdots i_T}^t \tilde{x}_{i_1 \cdots i_T}} \right)} = \frac{\sum_{i_1=0}^{n_1} \cdots \sum_{i_t=0}^{n_t} \left( \sum_{i_{t+1}=0}^{n_{t+1}} \cdots \sum_{i_{T}=0}^{n_{T}} e^{\tilde{R}_{i_1 \cdots i_T}^t \tilde{x}_{i_1 \cdots i_T}} \right) r_{j_1 \ldots j_t}^t(\tilde{x}')}{\sum_{i_1=0}^{n_1} \cdots \sum_{i_t=0}^{n_t} \left( \sum_{i_{t+1}=0}^{n_{t+1}} \cdots \sum_{i_{T}=0}^{n_{T}} e^{\tilde{R}_{i_1 \cdots i_T}^t \tilde{x}_{i_1 \cdots i_T}} \right)}.
\]

Using \( \tilde{x}_{j_1 \ldots j_t, 0, \ldots, 0} = 1 \) and \( \tilde{x}'_{j_1 \ldots j_t, 0, \ldots, 0} = 1 \), we know that \( r_{j_1 \ldots j_t}(\tilde{x}') > R(\tilde{x}') \) when \( r_{j_1 \ldots j_t}(\tilde{x}') > R(\tilde{x}) \), and \( r_{j_1 \ldots j_t}(\tilde{x}') = R(\tilde{x}') = R(\tilde{x}) \) when \( r_{j_1 \ldots j_t}(\tilde{x}') = R(\tilde{x}') \).
Having established Lemmas 4 and 5, we now proceed to prove Proposition 1.1. 

**Proof of Proposition 1.1.** Suppose \( \tilde{x}^* \) is an optimal solution to (16). We know that \( \tilde{x}_{0\cdots0} \) must be 1. In the following, we show that we can adjust \( \tilde{x}^* \) such that it satisfies all the properties in Proposition 1.1.

We first show part 1 of Proposition 1.1. Suppose there exist \( t \in \{1, \ldots, T\}, j_1 \in \mathcal{N}_t \cup \{0\}, \ldots, j_t \in \mathcal{N}_t \cup \{0\} \) such that \( \tilde{x}_{j_1 \cdots j_t 1\cdots1_0\cdots0}^* = 1, k \leq j_t \) and \( \tilde{x}_{j_1 \cdots j_t 1\cdots1_0\cdots0}^* = 0 \). Since \( \tilde{x}_{j_1 \cdots j_t 1\cdots1_0\cdots0}^* = 1 \), according to Proposition 1.1, we know that \( \tilde{x}_{j_1 \cdots j_t 1\cdots1_0\cdots0}^* = 1 \). Also, from \( \tilde{x}_{j_1 \cdots j_t 1\cdots1_0\cdots0}^* = 0 \) we know that \( k \neq 0, j_t \neq 0 \). Now define a new solution \( \tilde{x}' \) such that

\[
\tilde{x}'_{j_1 \cdots j_t 1\cdots1_0\cdots1} = \tilde{x}_{j_1 \cdots j_t 1\cdots1 0\cdots0}^*, \quad \forall i_{t+1}, \ldots, i_T,
\]

\[
\tilde{x}'_{i_1 \cdots i_T} = \tilde{x}_{i_1 \cdots i_T}, \quad \forall (i_1, \ldots, i_{t-1}, i_t) \neq (j_1, \ldots, j_{t-1}, k), \forall i_{t+1}, \ldots, i_T.
\]

It is easy to verify that \( \tilde{x}' \) is a feasible solution to (16). Since \( \tilde{x}_{j_1 \cdots j_t 1\cdots1_0\cdots0}^* = 1 \) and \( j_t \neq 0 \), according to Lemma 4, \( r_{j_1 \cdots j_t}^{[t]} \geq R(\tilde{x}') \). Furthermore, because \( 1 \leq k \leq j_t \), we know that

\[
r_{j_1 \cdots j_t}^{[t]}(\tilde{x}') = \sum_{s=1}^{t-1} p_j^{(s)} + p_{j_t}^{(t)} + \sum_{i_{t+1}=0}^{n_{t+1}} \sum_{i_{t+1}=0}^{n_T} e^{\tilde{\mu}_{j_1 \cdots j_t 1\cdots1 i_{t+1} \cdots1_T}} \tilde{x}'_{j_1 \cdots j_t 1\cdots1 i_{t+1} \cdots1_T} \sum_{s=t+1}^T p_{j_s}^{(s)}
\]

\[
\geq \sum_{s=1}^{t-1} p_j^{(s)} + p_{j_t}^{(t)} + \sum_{i_{t+1}=0}^{n_{t+1}} \sum_{i_{t+1}=0}^{n_T} e^{\tilde{\mu}_{j_1 \cdots j_t 1\cdots1 i_{t+1} \cdots1_T}} \tilde{x}_{j_1 \cdots j_t 1\cdots1 0\cdots0}^* \sum_{s=t+1}^T p_{j_s}^{(s)}
\]

\[
= r_{j_1 \cdots j_t}^{[t]}(\tilde{x}^*).
\]

Therefore, according to Lemma 5, we have \( R(\tilde{x}') \geq R(\tilde{x}^*) \), which indicates that \( \tilde{x}' \) is also an optimal solution. Repeat the argument until there is no such \( t, j_1, \ldots, j_t \) exist and we will obtain an optimal solution that satisfies the property in part 1 of Proposition 1.1.

We next show part 2 of Proposition 1.1. Let \( \tilde{x}^* \) be an optimal solution that satisfies part 1. Suppose there exist \( t \in \{2, \ldots, T\}, j_1, \ldots, j_t \in \mathcal{N}_t \cup \{0\} \), such that \( \tilde{x}_{j_1 \cdots j_t 1\cdots1 0\cdots0}^* = 1 \), \( \tilde{x}_{j_1 \cdots j_t 1\cdots1 0\cdots0}^* = 1 \), \( \tilde{x}_{j_1 \cdots j_t 1\cdots1 0\cdots0}^* = 0 \), \( j_1, \ldots, j_{t-1} \neq 0 \). Since \( \tilde{x}_{j_1 \cdots j_t 1\cdots1 0\cdots0}^* = 1 \) and \( \tilde{x}_{j_1 \cdots j_t 1\cdots1 0\cdots0}^* = 0 \), we know \( j_t \neq 0 \). Now we construct a new solution \( \tilde{x}' \) such that

\[
\tilde{x}'_{j_1 \cdots j_{t-1} 1\cdots1 0\cdots1} = \tilde{x}_{j_1 \cdots j_{t-1} 1\cdots1 0\cdots0}^*, \quad \forall i_{t+1}, \ldots, i_T,
\]

\[
\tilde{x}'_{i_1 \cdots i_T} = \tilde{x}_{i_1 \cdots i_T}, \quad \forall (i_1, \ldots, i_{t-1}, i_t) \neq (j_1, \ldots, j_{t-1}, j_t), \forall i_{t+1}, \ldots, i_T.
\]

It is easy to verify that \( \tilde{x}' \) is a feasible solution to (16). Since \( \tilde{x}_{j_1 \cdots j_t 1\cdots1 0\cdots0}^* = 1 \) and \( \tilde{x}_{j_1 \cdots j_t 1\cdots1 0\cdots0}^* = 0 \), according to Lemma 4, \( r_{j_1 \cdots j_t}^{[t]} \geq R(\tilde{x}') \). Furthermore, because \( \tilde{p}_{j_1}^{(1)} + \cdots + \tilde{p}_{j_{t-1}}^{(t-1)} \geq \tilde{p}_{j_t}^{(t)} \), we know that

\[
r_{j_1 \cdots j_t}^{[t]}(\tilde{x}') = \sum_{s=1}^{t-1} p_j^{(s)} + p_{j_t}^{(t)} + \sum_{i_{t+1}=0}^{n_{t+1}} \sum_{i_{t+1}=0}^{n_T} e^{\tilde{\mu}_{j_1 \cdots j_t 1\cdots1 i_{t+1} \cdots1_T}} \tilde{x}'_{j_1 \cdots j_t 1\cdots1 i_{t+1} \cdots1_T} \sum_{s=t+1}^T p_{j_s}^{(s)}
\]

\[
\geq \sum_{s=1}^{t-1} p_j^{(s)} + p_{j_t}^{(t)} + \sum_{i_{t+1}=0}^{n_{t+1}} \sum_{i_{t+1}=0}^{n_T} e^{\tilde{\mu}_{j_1 \cdots j_t 1\cdots1 i_{t+1} \cdots1_T}} \tilde{x}_{j_1 \cdots j_t 1\cdots1 0\cdots0}^* \sum_{s=t+1}^T p_{j_s}^{(s)}
\]

\[
= r_{j_1 \cdots j_t}^{[t]}(\tilde{x}^*).
\]
Therefore, according to Lemma 5, we have \( R(\tilde{x}') \geq R(\tilde{x}^*) \), which indicates that \( \tilde{x}' \) is also an optimal solution. Repeat the argument until there is no such \( t, j_1, \ldots, j_t, j'_1, \ldots, j'_{t-1} \) exist and we will obtain an optimal solution that satisfies all the properties in part 1 and part 2 of Proposition I.1.

I.3. Proof of Proposition A.3

We first prove Lemma 1, then prove Proposition A.3.

Proof of Lemma 1. Let \( \tilde{\Psi} = \{ \tilde{x} \mid \tilde{x} \) is an optimal solution to (16)\( \} \). Let \( \tilde{x}^\text{min} \) denote the component-wise minimum of all \( \tilde{x} \in \tilde{\Psi} \), i.e.,

\[
\tilde{x}^\text{min}_{t_1, \ldots, t_T} = \min_{\tilde{x} \in \tilde{\Psi}} \tilde{x}_{t_1, \ldots, t_T}, \quad \forall i_1, \ldots, i_T.
\]

By repeatedly using Lemma 4 and Lemma 5, we can show that \( \tilde{x}^\text{min} \) corresponds to the minimum optimal assortment policy.

Proof of Proposition A.3. Let \( \tilde{x}^* \) be an optimal solution to problem (16) that satisfies the properties of Proposition I.1. In the following, we show that \( \tilde{x}^* \) satisfies all the properties in Proposition A.3.

We first show part 1. Suppose that there exist \( j_1 \in g^*_1 \) such that \( j_1 > m_1 \). Let \( R^*_{1:t} \) denote the optimal expected revenue for problem \( \Phi_{1:t} \), and \( R^*_{2:T} \) denote the optimal expected revenue for problem \( \Phi_{2:T} \). We have

\[
R(\tilde{x}^*) \geq R^*_{1:t} + R^*_{2:T}
\]

and

\[
r^*_t(\tilde{x}^*) = p^*_t + \sum_{i_2=0}^{n_2} \sum_{i_T=0}^{n_T} e^{\mu_{i_2} \cdots \mu_{i_T}} \tilde{x}^*_{i_2 \cdots i_T} \sum_{s=2}^{T} p^*_s 
\]

By Lemma 4, we also have

\[
r^*_t(\tilde{x}^*) \geq R(\tilde{x}^*)
\]

Combining the above three inequalities, we obtain

\[
p^*_t \geq R^*_{1:t}.
\]

However, this implies that \( \{1, \ldots, m_1, j_1\} \) is an optimal assortment for problem \( \Phi_{1:t} \), which is impossible given the definition of \( m_1 \). Thus part 1 holds true.

We then show part 2. Suppose that there exist \( t \in \{1, \ldots, T\} \), \( k_1 \in g^*_1 \cup \{0\} \), \( k_2 \in g^*_2(k_1) \cup \{0\} \), \( \ldots \), \( k_{t-2} \in g^*_t(k_{t-1} - 1) \cup \{0\} \), \( k_{t-1} \in g^*_t(k_{t-2}) \), and \( j_t \in \mathcal{N}_t \cup \{0\} \), \( \ldots \), \( j_T \in \mathcal{N}_T \cup \{0\} \) such that \( (j_1, \ldots, j_T) \) is an available \( (T - t + 1) \)-stage choice path under \( \Pi^\text{min}_{t:T} \) for the embedded problem \( \Phi_{t:T} \), but \( \tilde{x}^*_{i_1 \cdots i_T} = 0 \). Then there must exist \( t' \in \{t, \ldots, T\} \) such that \( \tilde{x}^*_{i_1 \cdots i_T} = 0 \) and \( \tilde{x}^*_{i_1 \cdots i_T} = 1 \). Now define a new solution \( \tilde{x}' \) such that

\[
\tilde{x}'_{i_1 \cdots i_T} = \begin{cases} 1, & \text{if } (j_1, \ldots, j_{t'}, i_{t'+1} \cdots, i_T) \text{ is available under } \Pi^\text{min}_{t:T} \text{ for } \Phi_{1:T}, \forall i_{t'+1}, \ldots, i_T, \\ 0, & \text{otherwise,} \end{cases}
\]
It is easy to verify that \( \hat{x} \) is a feasible solution to (16). Let \( R_{t,T}^{\text{opt}} \) denote the optimal expected revenue for problem \( \Phi_{t,T} \). Since \((j_t, \cdots, j_{t'}, 0, \cdots, 0)\) is available under \( \Pi_{t,T}^{\text{min}} \) for problem \( \Phi_{t,T} \) and \( j_{t'} \neq 0 \), by applying Lemma 4 to problem \( \Phi_{t,T} \), we have
\[
\sum_{s = t}^{t'} p_{j_s}(s) + \sum_{i = 0}^{n_{t+1}} \cdots \sum_{i' = 0}^{n_{T+1}} e^{\hat{a}_{k} \cdot k_{t-1} \cdots j_{t'} i_{t'+1} \cdots i_{T}} \hat{x}_{k_{1} \cdots k_{t-1} \cdots j_{t'} i_{t'+1} \cdots i_{T}} \sum_{s = t+1}^{T} s_{i_t} \geq R_{t,T}^{*}.
\]

Furthermore, by the definition of \( \Pi_{t,T}^{\text{min}} \), we have
\[
\sum_{s = t}^{t'} p_{j_s}(s) + \sum_{i = 0}^{n_{t+1}} \cdots \sum_{i' = 0}^{n_{T+1}} e^{\hat{a}_{k} \cdot k_{t-1} \cdots j_{t'} i_{t'+1} \cdots i_{T}} \hat{x}_{k_{1} \cdots k_{t-1} \cdots j_{t'} i_{t'+1} \cdots i_{T}} \sum_{s = t+1}^{T} s_{i_t} > R_{t,T}^{*},
\]
as otherwise \( \Pi_{t,T}^{\text{min}} \) would not be the minimum optimal assortment policy for \( \Phi_{t,T} \). Returning to the original problem \( \Phi_{1,T} \), we then have
\[
R_{t_{1} \cdots k_{t-1} \cdots j_{t'} i_{t'+1} \cdots i_{T}}^{*} \geq \sum_{s = t}^{T} p_{j_s}(s) + \sum_{s = t}^{T} p_{j_s}(s) + \sum_{i = 0}^{n_{t+1}} \cdots \sum_{i' = 0}^{n_{T+1}} e^{\hat{a}_{k} \cdot k_{t-1} \cdots j_{t'} i_{t'+1} \cdots i_{T}} \hat{x}_{k_{1} \cdots k_{t-1} \cdots j_{t'} i_{t'+1} \cdots i_{T}} \sum_{s = t+1}^{T} s_{i_t} \geq R_{t,T}^{*}.
\]

where the last inequality follows from \( \hat{x}_{k_{1} \cdots k_{t-1} \cdots j_{t'} i_{t'+1} \cdots i_{T}} = 1, k_t \neq 0 \) and Lemma 4. Therefore, by Lemma 5, we have
\[
R(\hat{x}) > R(\hat{x}^{*}),
\]
which contradicts the fact that \( \hat{x}^{*} \) is an optimal solution to (16). Thus part 2 must hold true. \( \square \)

### I.4. The Recursive Merged Path Selection Algorithm

The Recursive Merged Path Selection (RMPS) algorithm is presented in Algorithm 6, which takes \( T > 1 \) as the “number of stages” input and \( N_1, \ldots, N_T \) as the “product sets” input. Algorithm 6 calls Algorithm 5 as a sub-routine, which takes a single product set \( N \) as input. Algorithm 5 corresponds to the procedure of solving a single-stage assortment optimization problem under the standard MNL model, as well as the “merging” procedure. Clearly, the running time of Algorithm 5 is \( O(|N| \log |N|) \) given an input \( N \).

Given any \( T \)-stage assortment optimization problem \( (T \geq 2) \) with \( N_1, \ldots, N_T \) being the product set in each stage, we can efficiently solve the \( T \)-stage problem via the following procedure:

- We first compute RMPS(1; \( N_T \)), which takes \( O(n_T) \) time. We then store the output in a table.
- We then compute RMPS(2; \( N_{T-1}, N_T \)), which relies on reading the historical output of RMPS(1; \( N_T \)) from the table. Computing RMPS(2; \( N_{T-1}, N_T \)) requires \( O(n_T \cdot n_{T-1} \log (n_T \cdot n_{T-1})) \) time. We then add the output into the table.
- Eventually, we can compute RMPS(\( T; N_1, \ldots, N_T \)) in \( O(n_1 \cdots n_T \log (n_1 \cdots n_T)) \) time, as presented in Algorithm 6.

The total running time of the above procedure is \( O(n_1 \cdots n_T \log (n_1 \cdots n_T)) \). And one can recover the optimal \( T \)-stage assortment policy from the outputs in the table.
Algorithm 5 RMPS(1;\mathcal{N})

\textbf{Input:} product set \mathcal{N}.

1: Sort the products in \mathcal{N} according to their revenue (from large to small, breaking ties arbitrarily) and represent \mathcal{N} as
\{(\mu_1(\mathcal{N}), p_1(\mathcal{N})), \ldots, (\mu_{|\mathcal{N}|}(\mathcal{N}), p_{|\mathcal{N}|}(\mathcal{N}))\},
where \mu_i(\mathcal{N}) and p_i(\mathcal{N}) are the mean utility and the price of the i-th expensive product in \mathcal{N}, respectively (\forall i \in \mathcal{N}).

2: Solve the single-stage assortment optimization problem with product set \mathcal{N}, and let \textit{m} be the smallest value such that \mathcal{N}^{\ast} = \{(\mu_1(\mathcal{N}), p_1(\mathcal{N})), \ldots, (\mu_m(\mathcal{N}), p_m(\mathcal{N}))\} is an optimal solution.

3: Construct a new product set \mathcal{N}_{\text{new}} defined as
\[
\left\{ \left( \log \left( 1 + \sum_{i=1}^{m} e^{\mu_i(\mathcal{N})} \right), \frac{\sum_{i=1}^{m} e^{\mu_i(\mathcal{N})} p_i(\mathcal{N})}{1 + \sum_{i=1}^{m} e^{\mu_i(\mathcal{N})}} \right) : (\mu_{m+1}(\mathcal{N}), p_{m+1}(\mathcal{N})), \ldots, (\mu_{|\mathcal{N}|}(\mathcal{N}), p_{|\mathcal{N}|}(\mathcal{N})) \right\}.
\]

  \textit{N}_{\text{new}} \text{ contains } |\mathcal{N}| - \textit{m} + 1 \text{ products. Intuitively, the first product in } \mathcal{N}_{\text{new}} \text{ is generated by “merging” the } \textit{m} \text{ most expensive products in } \mathcal{N} \text{ with the no-purchase option.}

4: return \(\mathcal{N}_{\text{new}}, \mathcal{N}^{\ast}\)

Algorithm 6 RMPS(T;\mathcal{N}_1,\ldots,\mathcal{N}_T)

\textbf{Input:} number of stages \(T > 1\); product sets \mathcal{N}_1,\ldots,\mathcal{N}_T.

\textbf{Assumption:} RMPS(1;\mathcal{N}_T),\ldots,\text{RMPS}(T-1;\mathcal{N}_2,\ldots,\mathcal{N}_T) \text{ have been computed.}

1: for \(t = T, \ldots, 1\) do
2: \hspace{1em} If \(t < T\), set \((S_{t+1:T}, S_{t+1:T}^{\ast}) = \text{RMPS}(T-t;\mathcal{N}_{t+1},\ldots,\mathcal{N}_T)\); if \(t = T\), set \(S_{T+1:T} = \{(0,0)\}.

  \text{// Note that we do not call RMPS here; instead, we read } (S_{t+1:T}, S_{t+1:T}^{\ast}) \text{ from a table of historical outputs.}

3: \hspace{1em} Let \(\mu_i(S_{t+1:T})\) and \(p_i(S_{t+1:T})\) denote the mean utility and the price of the i-th expensive product in \(S_{t+1:T}\); respectively (\forall i \in S_{t+1:T}). \text{Similar for notations } \mu_i(\mathcal{N}_i) \text{ and } p_i(\mathcal{N}_i).

4: \hspace{1em} for \(i = 1, \ldots, |\mathcal{N}_t|\) do
5: \hspace{2em} for \(j = 1, \ldots, |S_{t+1:T}|\) do
6: \hspace{3em} Set \(\tilde{\mu}(t; i, j) = \mu_i(\mathcal{N}_i) + \mu_j(S_{t+1:T}) \text{ and } \tilde{p}(t; i, j) = p_i(\mathcal{N}_i) + p_j(S_{t+1:T}).

7: \hspace{1em} Construct a product set
\[
S = \bigcup_{t=1}^{T} \bigcup_{i=1}^{|\mathcal{N}_i|} \bigcup_{j=1}^{|S_{t+1:T}|} \{(\tilde{\mu}(t; i, j), \tilde{p}(t; i, j))\}.
\]

    \text{// The size of } S \text{ is } O(n_1 \cdots n_T).

8: Compute
\(\text{RMPS}(1; S)\).

    \text{// We call Algorithm 5 here. The running time is } O(|S| \log |S|) = O(n_1 \cdots n_T \log(n_1 \cdots n_T)).

9: return \((S_{\text{new}}, S^{\ast})\)