Coordinated Inventory Stocking and Assortment Customization

Yicheng Bai[1], Omar El Housni[1], Paat Rusmevichientong[2], Huseyin Topaloglu[1]

[1]School of Operations Research and Information Engineering, Cornell Tech, New York, NY 10044
[2]Marshall School of Business, University of Southern California, Los Angeles, CA 90089
{yb279,oe46}@cornell.edu, rusmevic@marshall.usc.edu, topaloglu@orie.cornell.edu

July 11, 2024

We study a joint inventory stocking and assortment customization problem. We have access to a set of products that can be used to stock a storage facility with limited capacity. At the beginning of the selling horizon, we decide how many units of each product to stock. Customers of different types with type-dependent preferences for the products arrive over the selling horizon. Depending on the remaining product inventories and the type of the customer, we offer a product assortment to the arriving customer. The customer makes a choice within the assortment according to a choice model based on her type. Our goal is to choose the stocking quantities at the beginning of the selling horizon and to find a policy to offer an assortment to the customer type arriving at each time period so that we maximize the total expected revenue over the selling horizon. Our work is motivated by online platforms making same-day delivery promises or selling groceries, which require operating out of a capacity-constrained urban warehouse to be close to customers, but allow offering a different assortment based on some knowledge of the type of each customer. Finding a good assortment customization policy requires approximating a high-dimensional dynamic program with a state variable that keeps track of the remaining inventories. Making the stocking decisions requires solving an optimization problem that involves the value functions of the dynamic program in the objective function. We give an approximation framework for the joint inventory stocking and assortment customization problem. Using our framework, we obtain a $\frac{1}{4}(1-\frac{1}{e})$-approximate solution when the customers choose under the multinomial logit model. Under a general choice model, letting n be the number of products and K be the total number of units we can stock, we give a $(1-(\sqrt{2}+1)\sqrt[3]{\frac{n}{K}})$-approximate solution, which is asymptotically optimal for large storage capacity. Our computational experiments on synthetically generated datasets, as well as on a real-world supermarket dataset, show that our approximation framework performs well against both upper bounds on the optimal performance and other possible heuristics.

1. Introduction

The ability to customize the product assortment offered to the customers is an important source of flexibility for online retailers. From the customer viewpoint, customizing the assortment based on some information on the type of a customer may align the products viewed by the customer with her preferences, allowing the customer to have a more satisfactory experience. From the firm viewpoint, customizing the assortment for each customer may facilitate shifting the demand away from the products with scarce inventories, allowing the firm to utilize its inventories more efficiently. Keeping these two viewpoints in mind, making assortment customization decisions requires keeping a balance between offering an assortment that will satisfy the current customer and reserving the products with scarce inventories for the customers that will arrive in the future. Thus, while the assortment customization decisions should depend on the current inventories of the products, the stocking decisions should anticipate how the customized assortments will deplete the inventory, thereby creating a natural interaction between inventory stocking and assortment customization. The challenge of coordinating assortment customization and inventory stocking appears in numerous online retail settings. Online grocers, such as Amazon Fresh, operate out of urban warehouses to be close to their customers. Such urban warehouses tend to be capacity-constrained. Thus, online grocers face the problem of how to periodically stock their capacitated warehouses and how to use the stocked inventory to serve the customers arriving at their platforms. In particular, when a customer arrives at their platform, online grocers have access to a variety of information about the type of a customer, such as zip code, age, gender and purchase history. Using this information, along with the remaining inventories of the products, they have the ability to offer a customized assortment to each customer type. Even if the platform does not attempt to use the information about an arriving customer to customize the assortment, aligning the offered assortment with the remaining inventories is still a challenge. Similar tradeoffs occur for online retailers with same-day delivery promises, as they also operate tight urban warehouses.

In this paper, we study a joint inventory stocking and assortment customization problem. We have a set of products to stock a storage facility with limited capacity. At the beginning of the selling horizon, we decide how many units of each product to stock. Customers of different types with type-dependent product preferences arrive over the selling horizon. Each customer type encodes all information available about the customer, such as her zip code, age, gender and purchase history. Depending on the remaining inventories and type of the customer, we offer a product assortment to the arriving customer. The customer makes a choice within the assortment according to a choice model based on her type. Our goal is to pick the stocking quantities at the beginning of the selling horizon and to find a policy to offer a customized assortment to the customer type arriving at each time period to maximize the total expected revenue over the selling horizon. Our work is motivated by online retailers making same-day delivery promises or selling groceries, as both operate out of capacity-constrained urban warehouses. Computing the optimal assortment customization policy requires solving a dynamic program that keeps track of the remaining inventories. The value function of the dynamic program gives the optimal total expected revenue over the selling horizon as a function of the stocking quantities. Making the stocking decisions requires solving an optimization problem to choose the state in the initial value function.

Technical Contributions: Our main technical contributions include algorithms to obtain constant-factor and asymptotically optimal solutions to the joint inventory stocking and assortment customization problem. To construct these algorithms, we make use of a configurable approximation framework that we develop in our paper.

Approximation Framework. We develop an approximation framework for our joint inventory stocking and assortment customization problem. The framework has three steps. In Step 1, letting n be the number of products, we construct a function $f : \mathbb{Z}_+^n \to \mathbb{R}_+$ that upper bounds the optimal total expected revenue from the assortment customization decisions when viewed as a function of the initial stocking quantities. We refer to this function as our surrogate. In Step 2, we make the stocking decisions. Letting c_i be the stocking quantity of product i and K be limit on the total number of units stocked, using the vector $\boldsymbol{c} = (c_1, \ldots, c_n)$, we choose the stocking quantities as an α-approximate solution to the problem $\max_{\boldsymbol{c} \in \mathbb{Z}_+^n} \{f(\boldsymbol{c}) : \sum_{i=1}^n c_i \leq K\}$ for some $\alpha \in (0, 1]$. In Step 3, we make the assortment customization decisions. Letting $\widehat{\boldsymbol{c}}$ be our stocking quantities, we construct an assortment customization policy such that the total expected revenue of the policy starting with the stocking quantities $\widehat{\boldsymbol{c}}$ is at least $\beta\, f(\widehat{\boldsymbol{c}})$ for some $\beta \in (0, 1]$. In this case, letting opt be the optimal total expected revenue in the joint stocking and assortment customization problem, we show that using the stocking quantities computed in Step 2 and subsequently following the assortment customization policy constructed in Step 3 yields a total expected revenue of at least $\alpha\,\beta\,\mathsf{opt}$. Therefore, our approximation framework provides an $\alpha\beta$-approximate solution.

Performance Guarantees. Using our approximation framework in practice requires answering three questions. How do we construct a surrogate in Step 1 that upper bounds the optimal total expected revenue from the assortment customization decisions when viewed as a function of the initial stocking quantities? How do we choose the stocking quantities in Step 2 by approximately solving a problem that involves the surrogate in the objective function? How do we construct an assortment customization policy in Step 3 such that the total expected revenue of the policy is lower bounded by a certain fraction of the surrogate? We show that we can answer all of these questions. In this way, using our approximation framework, if the customers choose under the multinomial logit model, then we get a $\frac{1}{4}(1 - \frac{1}{e})$-approximate solution to the joint stocking and assortment customization problem, whereas if the customers choose under a general choice model, then we get a $(1 - (\sqrt{2} + 1)\, \sqrt[3]{\frac{n}{K}})$-approximate solution. To our knowledge, these are the first guarantees for our problem class. The last guarantee becomes near optimal when the storage capacity is large. This result is different from those in the revenue management literature that give asymptotically optimal policies when the capacities of the resources get large, because even if the storage capacity in our problem is large, it is not clear that the stocking quantity of each product will be large.

Surrogate Construction. In Step 1 of our approximation framework, we construct our surrogate by using a linear program to approximate the optimal total expected revenue from the assortment customization decisions. This linear program is the so-called choice-based linear program in the revenue management literature. The optimal objective value of this linear program at some fixed stocking quantities is an upper bound on the optimal total expected revenue from the assortment customization decisions starting from the fixed stocking quantities, as needed in Step 1. Irrespective of whether the customers choose under the multinomial logit model or a general choice model, throughout the paper, we always use this linear programming-based surrogate.

Inventory Stocking Decisions. In Step 2 of our approximation framework, we choose the stocking quantities by solving the problem $\max_{\boldsymbol{c}\in\mathbb{Z}_+^n}\{f(\boldsymbol{c}):\sum_{i=1}^n c_i\le K\}$, where $f(\boldsymbol{c})$ is the linear programming-based surrogate. We show that this stocking problem is APX-hard even when the customers choose according to the multinomial logit model, so we turn to approximation strategies. Under the multinomial logit model, we develop a monotone and submodular approximation to the linear programming-based surrogate. Using the monotone and submodular approximation to the surrogate in the stocking problem, we cast the stocking problem as maximizing a monotone and submodular function under a cardinality constraint, which admits a constant-factor approximation; see Soma and Yoshida (2018). Under a general choice model, we solve a continuous approximation to the stocking problem and round an optimal solution while bounding the loss. Thus, we can obtain an approximate solution to the stocking problem, as needed in Step 2.

Assortment Customization Decisions. In Step 3 of our approximation framework, given stocking quantities $\widehat{\boldsymbol{c}}$ for the products, using $\widehat{c}_{\min}=\min_{i\in\mathcal{N}}\{\widehat{c}_i:\widehat{c}_i\ge 1\}$ to capture the smallest non-zero stocking quantity, we construct an assortment customization policy that provides a total expected revenue of at least $\max\{\frac{1}{2},1-\frac{1}{\sqrt{\widehat{c}_{\min}}}\}\,f(\widehat{\boldsymbol{c}})$. Thus, we can construct an assortment customization policy such that the total expected revenue of the policy is lower bounded by a certain fraction of the linear programming-based surrogate, as needed in Step 3. Our assortment customization policy is extremely simple. We solve a linear program to come up with the probability of offering each assortment to a customer of each type. When a customer of a particular type arrives, we sample an assortment from the distribution corresponding to the customer type. The sampled assortment may include products without any remaining inventories. We drop the products without remaining inventories from the sampled assortment and offer the remaining products.

Positioning Our Work. Under the multinomial logit model, we give a $\frac{1}{4}(1-\frac{1}{e})$-approximate solution to the joint stocking and assortment customization problem. This result is the first to give a constant-factor approximate solution for a joint stocking and assortment customization problem. Under a general choice model, we give a $(1-(\sqrt{2}+1)\,\sqrt[3]{\frac{n}{K}})$-approximate solution, which is near optimal as the storage capacity gets large. Note that there are no hidden constants in the last performance guarantee. The performance guarantee does not depend on any problem parameters other than n and K, so the expected number of arrivals, choice probabilities and product revenues can be arbitrary and we still obtain the same performance guarantee. When working with a general choice model, we only require that if we know the type of a customer, then we can efficiently find an assortment that maximizes the expected revenue from the customer. Considering our assortment customization policy, there is existing work that constructs policies for similar assortment customization problems. However, existing work assumes that the stocking quantities are fixed, given as the problem data. Even under fixed stocking quantities, the policies in the existing work have some form of a post-processing step, obscuring the intuitive nature of the policy. Our assortment customization policy directly follows the distribution that we obtain from a linear program without post-processing this distribution in any way. Through a novel analysis, the performance guarantee for our policy matches the best ones in the literature, but implementing our policy is much simpler. While the assortment customization decisions under fixed stocking quantities are not our sole focus, we make a useful contribution in that domain.

Computational Experiments and Extensions. We work with synthetically generated datasets, as well as datasets based on purchases in a real-world supermarket. We compare the performance of our approximation framework with an efficiently computable upper bound on the optimal total expected revenue and heuristics. On average, we obtain solutions within 6.01% of the upper bound and improve the performance of the heuristics by 4.16%. Using our approximation framework, we quantify the benefit from customizing the assortment for each customer based on her type. Lastly, we give useful extensions of our approximation framework. In particular, instead of limiting the total number of units stocked, we can limit the total inventory investment in our approximation framework. To capture end of selling horizon effects, we can incorporate salvage values for the units left at the end of the selling horizon. Building on the fact that we can limit the total inventory investment in our approximation framework, we can use our approach to develop a heuristic to maximize the total expected profit, where the total expected profit is given by the difference between the total expected revenue from the sales and the total cost of the stocked units.

Related Literature: There is work on making inventory stocking decisions at the beginning of the selling horizon when the customers arriving over the selling horizon make choices among all products with remaining inventories. Honhon et al. (2010) use the non-parametric choice model, where each customer arrives with a ranked list of products in mind and purchases the highest ranked available product. The authors give a fluid approximation where the customers can make fractional purchases. Goyal et al. (2016) give a polynomial time approximation scheme without resorting to a fluid approximation. Under the multinomial logit model, Aouad et al. (2018) give a randomized algorithm that provides a constant-factor performance guarantee with high probability. Aouad et al. (2019) use the non-parametric choice model to give a guarantee that depends logarithmically on the gap between the unit product revenues. Under the Markov chain choice model, El Housni et al. (2021) give an algorithm with an additive optimality gap that grows sublinearly with the number of time periods in the selling horizon and the number of products. Liang et al. (2021) give a similar sublinear additive performance guarantee under the multinomial logit model.

In the papers discussed above, the customers choose among all products with remaining inventories. In contrast, we adjust the assortment to be offered to each customer. Chen et al. (2022) develop a model where they choose the stocking quantities for the products and match each arriving customer to a product. The authors solve an integer program to make their stocking decisions, so their algorithm does not run in polynomial time. Zhang et al. (2022) choose the stocking quantities of the products and a ranking of the products to be displayed to the customers in search results. Even if the ranking can change over the selling horizon, they show that fixing the ranking can be near-optimal in the asymptotic regime they consider. There are papers that make approximations by assuming that we can offer products without remaining inventories but if a customer chooses a product without remaining inventory, then she leaves without a purchase, possibly resulting in a goodwill cost; see van Ryzin and Mahajan (1999), Gaur and Honhon (2006) and Topaloglu (2013). Such an approximation can be reasonable when the probability of stocking out is low.

Our linear programming-based surrogate uses a linear program to approximate the optimal total expected revenue from the assortment customization decisions. This linear program is known as the choice-based linear program in the literature; see Gallego et al. (2004) and Liu and van Ryzin (2008). In this linear program, we have one decision variable for each possible assortment, so it is customary to solve the linear program through column generation. The column generation subproblem has been studied under a variety of choice models, including multinomial logit, nested logit, paired combinatorial logit, non-parametric and Markov chain choice models; see Talluri and van Ryzin (2004), Davis et al. (2014), Blanchet et al. (2016), Zhang et al. (2020) and Aouad et al. (2021). Under certain choice models, we can solve the choice-based linear program directly without resorting to column generation; see Gallego et al. (2015), Feldman and Topaloglu (2017), Cao et al. (2023) and Cao et al. (2022). Under these choice models, we can reformulate the choice-based linear program by using the expected sales of each product as the decision variable.

There is work on assortment customization policies under fixed stocking quantities. Golrezaei et al. (2014) give a $\frac{1}{2}$-approximate policy by adjusting the unit revenue of each product as a function of its remaining inventory. Rusmevichientong et al. (2020) give a $\frac{1}{2}$-approximate policy by using linear value function approximations. For network revenue management problems, Ma et al. (2020) use nonlinear value function approximations to give a $\frac{1}{1+L}$-approximate policy when each product uses at most L resources. Letting $\widehat{c}_{\min}$ be the smallest product inventory, Feng et al. (2020) give a $1-\frac{1}{\sqrt{\widehat{c}_{\min}}}$-approximate policy. Ma et al. (2021) give a $1-\sqrt{\frac{\log \widehat{c}_{\min}}{\widehat{c}_{\min}}}$-approximate policy, but they may offer a product with no remaining inventory. Baek and Ma (2022) give improved performance guarantees for network revenue management problems under special network structures. Feng et al. (2023) consider variants of a two-stage matching problem with rewards depending both on the properties of the matched demand and supply, while allowing pricing decisions in the second stage. Their strongest performance guarantee gives a 0.767-approximate policy.

We note that all of the papers discussed in the previous paragraph, including Feng et al. (2020) and Ma et al. (2021), assume that the initial stocking quantity for each product is fixed, given as the problem data. Similarly, Feng et al. (2023) assume that the initial supply at each supply vertex is fixed. In contrast, we have to make product stocking decisions at the beginning of the selling horizon, which requires us to allocate a total of K units of storage capacity among the products. Thus, the performance guarantees in those papers do not apply to our setting. Product stocking decisions have a combinatorial nature and they bring challenges that have not been tackled before. In particular, to obtain a constant-factor approximate solution, we construct a monotone and submodular approximation to our linear programming-based surrogate, whereas to obtain an asymptotically optimal solution, we carefully round the solution to a continuous relaxation. To our knowledge, our approach gives the first efficient solution methods with performance guarantees when we consider the interplay between stocking decisions at the beginning of the selling horizon and assortment customization decisions over the course of the selling horizon.

In our model, we customize the assortment offered to each customer based on her type. Customers have been segmented into different customer types and the customers of a particular type choose among the products according to a choice model based on their type. We know the choice model governing the choice process of the customers of a particular type and observe the type of a customer before offering an assortment. This setup is sensible for many online platforms. Our model is different from those that learn the preferences of a specific customer from multiple interactions with the customer and personalize the assortment offerings for an individual.

Organization: In Section 2, we formulate our joint inventory stocking and assortment customization problem. In Section 3, we give our approximation framework. In Section 4, we describe our linear programming-based surrogate and the performance guarantees that we obtain by using the surrogate in our approximation framework. In Section 5, we focus on making stocking decisions under the multinomial logit model. In Section 6, we focus on making stocking decisions under a general choice model. In Section 7, we give our assortment customization policy. In Section 8, we test the performance of our approximation framework. In Section 9, we conclude.

2. Problem Formulation

We have n products indexed by $\mathcal{N} = \{1, \ldots, n\}$. The revenue associated with product i is r_i. We have m customer types indexed by $\mathcal{M} = \{1, \ldots, m\}$. We divide the selling horizon into a number of time periods, where each time period corresponds to small enough duration of time that there is at most one customer arrival at each time period. We have T time periods in the selling horizon indexed by $\mathcal{T} = \{1, \ldots, T\}$. At time period t, a customer of type j arrives into the system with probability λ_{jt}. We do not have a customer arrival at time period t with probability $1 - \sum_{j \in \mathcal{M}} \lambda_{jt}$. If we offer the assortment of products $S \subseteq \mathcal{N}$ to a customer of type j, then she chooses product i with probability $\phi_{ij}(S)$. We have $\phi_{ij}(S) = 0$ for $i \notin S$. We have K units of storage capacity for the products. Each unit of product that we stock at the beginning of the selling horizon consumes one unit of storage capacity. We want to decide which products to stock in which quantities and which assortment to offer to each arriving customer as a function of the remaining inventories and type of the customer so as to maximize the total expected revenue over the selling horizon.

We give a dynamic program to find the optimal policy to choose an assortment to offer to each arriving customer. Using the value functions of the dynamic program, we will decide which products to stock in which quantities. Letting x_i be the remaining inventory of product i, we use $\boldsymbol{x} = (x_i : i \in \mathcal{N})$ as the state variable at the beginning of a generic time period. At each time period, we observe the type of the arriving customer and offer an assortment of products. The offered assortment has to be a subset of the products that have remaining inventory. Given that the state of the system at the beginning of a generic time period is $\boldsymbol{x}$, we use $\mathcal{N}(\boldsymbol{x}) = \{i \in \mathcal{N} : x_i \geq 1\}$ to denote the set of products that have remaining inventory. Thus, if the state of the system at the beginning of a time period is $\boldsymbol{x}$, then the offered assortment must be a subset of $\mathcal{N}(\boldsymbol{x})$. Let $J_t(\boldsymbol{x})$ be the maximum total expected revenue over time periods $t, \ldots, T$ given that the state of the system at the beginning of time period t is $\boldsymbol{x}$. Using $\boldsymbol{e}_i \in \mathbb{R}^n_+$ to denote the i-th unit vector, we can compute the value functions $\{J_t : t \in \mathcal{T}\}$ through the dynamic program

$$\begin{aligned} J_t(\boldsymbol{x}) &= \sum_{j \in \mathcal{M}} \lambda_{jt} \max_{S \subseteq \mathcal{N}(\boldsymbol{x})} \left\{ \sum_{i \in \mathcal{N}} \phi_{ij}(S) \Big[r_i + J_{t+1}(\boldsymbol{x} - \boldsymbol{e}_i) \Big] + \left(1 - \sum_{i \in \mathcal{N}} \phi_{ij}(S) \right) J_{t+1}(\boldsymbol{x}) \right\} + \left(1 - \sum_{j \in \mathcal{M}} \lambda_{jt} \right) J_{t+1}(\boldsymbol{x}) \\ &= \sum_{j \in \mathcal{M}} \lambda_{jt} \max_{S \subseteq \mathcal{N}(\boldsymbol{x})} \left\{ \sum_{i \in \mathcal{N}} \phi_{ij}(S) \Big[r_i + J_{t+1}(\boldsymbol{x} - \boldsymbol{e}_i) - J_{t+1}(\boldsymbol{x}) \Big] \right\} + J_{t+1}(\boldsymbol{x}), \end{aligned} \tag{1}$$

with the boundary condition that $J_{T+1} = 0$. If the state variable at the beginning of the selling horizon is $\boldsymbol{x}$, then the optimal total expected revenue is $J_1(\boldsymbol{x})$.

On the right side of (1), a customer of type j arrives at time period t with probability λ_{jt}. If we offer the assortment S to this customer, then she chooses product i with probability $\phi_{ij}(S)$, in which case, we generate a revenue of r_i and consume one unit of inventory for product i. The customer does not make a purchase with probability $1 - \sum_{i \in \mathcal{N}} \phi_{ij}(S)$ and there is no customer arrival at time period t with probability $1 - \sum_{j \in \mathcal{M}} \lambda_{jt}$. In either case, we do not consume the inventory of a product. The second equality in (1) follows by arranging the terms. Throughout the paper, we assume that the choices of the customers are governed by a choice model such that if we add a product to an assortment, then the choice probabilities of all products in the assortment decrease. That is, we have $\phi_{ij}(S \cup \{k\}) \leq \phi_{ij}(S)$ for all $S \subseteq \mathcal{N}$, $k \in \mathcal{N} \setminus S$, $i \in S$ and $j \in \mathcal{M}$. All choice processes that are based on random utility maximization principle yield choice probabilities that satisfy this substitutability property. Letting c_i be the number of units of product i that we stock at the beginning of the selling horizon, we use $\boldsymbol{c} = (c_i : i \in \mathcal{N})$ to capture our stocking decisions. We can find the optimal stocking decisions by solving the problem

$$\mathsf{opt} \;=\; \max_{\boldsymbol{c} \in \mathbb{Z}_+^n} \left\{ J_1(\boldsymbol{c}) \;:\; \sum_{i \in \mathcal{N}} c_i \leq K \right\}, \tag{2}$$

where we maximize the total expected revenue over the selling horizon by choosing the initial stocking quantities for the products, while adhering to the storage space constraint.

We comment on several aspects of our problem formulation. In problem (2), we impose a cardinality constraint on the total number of stocked units. Instead, we may impose a knapsack constraint on the total space consumption of stocked units. Letting σ_i be the space consumption of a unit of product i, this constraint takes the form $\sum_{i \in \mathcal{N}} \sigma_i \, c_i \leq K$. If σ_i is the unit procurement cost of product i, then using the last constraint corresponds to maximizing the total expected revenue from the sales subject to a constraint on the total inventory investment. One of our algorithmic approaches for problem (2) is to approximate its objective function using a submodular function, so that we need to maximize a submodular function subject to a cardinality constraint, which is a well-studied problem. If we use the same algorithmic approach under a knapsack constraint, then we need to maximize a submodular function subject to a knapsack constraint, which is also well-studied. We discuss knapsack constraints after we give our submodular approximation.

In a variant of problem (2), we may try to maximize the total expected profit with no constraint on the stocking quantities, where the total expected profit is the difference between the total expected revenue from the sales and the total cost of the stocked units. Thus, letting σ_i be the unit procurement cost of product i, this variant is $\max_{\boldsymbol{c} \in \mathbb{Z}_+^n} \big\{ J_1(\boldsymbol{c}) - \sum_{i \in \mathcal{N}} \sigma_i \, c_i \big\}$. Obtaining a constant-factor approximation to the last problem can be difficult, because its objective function involves both positive and negative components, so its optimal objective value can be arbitrarily close to zero. In contrast, to construct algorithms with performance guarantees, our formulation in (2) maximizes the total expected revenue from the sales subject to a constraint on the stocking quantities. Such formulations that maximize the total expected revenue from the sales subject to constraints on the stocking quantities appear in the literature; see Goyal et al. (2016). Nevertheless, we can use our approach to construct heuristics for maximizing the total expected profit with no constraint on the stocking quantities. In particular, letting $\{B^1, \ldots, B^q\}$ be a collection of possible values for the total inventory investment that we are willing to make, we can focus on the problem $\max_{\boldsymbol{c} \in \mathbb{Z}_+^n} \big\{ J_1(\boldsymbol{c}) : \sum_{i \in \mathcal{N}} \sigma_i \, c_i \le B^\ell \big\}$ for each $\ell = 1, \ldots, q$. This problem maximizes the total expected revenue subject to a constraint on the total inventory investment. By the discussion in the previous paragraph, we can obtain an approximate solution to the last problem. Letting $\widehat{\boldsymbol{c}}^\ell$ be an approximate solution as a function of ℓ, we can use $\arg\max_{\ell = 1, \ldots, q} \big\{ J_1(\widehat{\boldsymbol{c}}^\ell) - \sum_{i \in \mathcal{N}} \sigma_i \, \widehat{c}_i^\ell \big\}$ as a heuristic solution to the problem $\max_{\boldsymbol{c} \in \mathbb{Z}_+^n} \big\{ J_1(\boldsymbol{c}) - \sum_{i \in \mathcal{N}} \sigma_i \, c_i \big\}$. This approach does not have a performance guarantee, but it performs well. We visit this approach in our computational experiments.

The selling horizon corresponds to a canonical interval of time, such as a week, for replenishing the inventory in a periodic review system. A natural question is whether we can use our model when we have multiple selling horizons with a stocking decision at the beginning of each. Note that we do not explicitly incorporate the total cost of the stocked units into the objective function, but we have a constraint on the total number of stocked units for a selling horizon. Because we do not explicitly incorporate the total cost of the stocked units into the objective function, even if we consider multiple selling horizons with a stocking decision at the beginning of each, the stocking decisions for different selling horizons do not interact, so we can focus on each selling horizon separately. That said, it also turns out that our approach can include salvage values for the units left at the end of the selling horizon, which may be useful for incorporating end of selling horizon effects. We explain how to incorporate salvage values after giving our solution strategy.

A critical component of our problem formulation is that we customize the assortment offered to each customer type. By customizing the assortment offered to each customer depending on the type of the customer, we generate a larger total expected revenue when compared with the case where we offer the same assortment to all customers arriving at a particular time period, irrespective of their types. We can give a tight characterization of the value provided by the ability to customize the assortment offered to each customer based on her type. In Appendix A, we show that the ability to customize the assortments increases the total expected revenue by a factor of at most m, where m is the number of customer types. While this result is intuitive, its proof uses the substitutability property of the choice model. Furthermore, we show that this bound is tight, so there are problem instances where the ability to customize the assortments increases the total expected revenue by a factor arbitrarily close to m. Thus, the value of customization can be significant.

3. Approximation Framework

In problem (2), opt corresponds to the maximum total expected revenue that we can obtain by jointly choosing the stocking quantities and making the customized assortment offer decisions. Simply computing the objective value of problem (2) at a particular solution requires having access to the value functions $\{J_t : t \in \mathcal{T}\}$, which, in turn, requires solving a dynamic program with a high-dimensional state variable. Thus, we focus on obtaining an approximate solution to problem (2). To be able to have an implementable solution, we need to obtain an approximate solution to problem (2) telling us which products to stock in which quantities, as well as an approximate policy telling us which customized assortment to offer at each time period as a function of the remaining inventories and type of the arriving customer.

We give an approximation framework that will allow us to reach both goals. In our approximation framework, we start with a surrogate function $f : \mathbb{Z}_+^n \to \mathbb{R}_+$ such that $f(\boldsymbol{c})$ will approximate $J_1(\boldsymbol{c})$. We will choose the surrogate such that $f(\boldsymbol{c})$ is an upper bound on $J_1(\boldsymbol{c})$. In this case, we will make our stocking decisions by solving problem (2) after replacing $J_1(\boldsymbol{c})$ in the objective function of this problem with $f(\boldsymbol{c})$. Under our surrogate, we will be able to obtain an approximate solution to (2) when we replace $J_1(\boldsymbol{c})$ with $f(\boldsymbol{c})$. Finally, we will construct an approximate policy to decide which assortment of products to offer to each customer so that the total expected revenue of the approximate policy can be lower bounded by a constant fraction of the surrogate evaluated at our stocking decisions. Below is the description of our approximation framework.

Approximation Framework:

Step 1: (Surrogate Function Construction) Construct a surrogate function $f : \mathbb{Z}_+^n \to \mathbb{R}_+$ to approximate J_1 such that we have $f(\boldsymbol{c}) \geq J_1(\boldsymbol{c})$ for all $\boldsymbol{c} \in \mathbb{Z}_+^n$.

Step 2: (Stocking) For $\alpha \in (0,1]$, compute the approximate stocking quantities $\widehat{\boldsymbol{c}} = (\widehat{c}_1, \ldots, \widehat{c}_n)$ as an α-approximate solution to the problem

$$\mathsf{app} \;=\; \max_{\boldsymbol{c} \in \mathbb{Z}_+^n} \left\{ f(\boldsymbol{c}) \;:\; \sum_{i \in \mathcal{N}} c_i \leq K \right\}. \tag{3}$$

Step 3: (Customization) For $\beta \in (0,1]$, construct a policy to offer customized assortments such that the total expected revenue of the policy with initial inventories $\widehat{\boldsymbol{c}}$ is at least $\beta f(\widehat{\boldsymbol{c}})$.

Our approximation framework is a blueprint with gaps. In Step 1, we need to choose a surrogate to upper bound the value function. In Step 2, we need to obtain an approximate solution to problem (3) with the chosen surrogate. In Step 3, we need to construct an assortment customization policy whose total expected revenue is lower bounded by a fraction of the chosen surrogate. We will show that we can fill all of these gaps for our joint stocking and assortment customization problem. In the next theorem, we give a performance guarantee for our approximation framework.

Theorem 3.1 (Approximation Framework) *If we use the stocking decisions $\widehat{\boldsymbol{c}}$ from Step 2 at the beginning of the selling horizon, followed by the assortment customization policy from Step 3, then we obtain a total expected revenue of at least $\alpha\beta\,\mathsf{opt}$.*

Proof: Let $\mathsf{Rev}(\widehat{\boldsymbol{c}})$ be the total expected revenue of the assortment customization policy in Step 3 starting with initial inventories $\widehat{\boldsymbol{c}}$. By Step 3, we have $\mathsf{Rev}(\widehat{\boldsymbol{c}}) \geq \beta\, f(\widehat{\boldsymbol{c}})$. Letting $\boldsymbol{c}^*$ be an optimal solution to problem (2), we have $\mathsf{opt} = J_1(\boldsymbol{c}^*)$. The solution $\boldsymbol{c}^*$ is feasible but not necessarily optimal to problem (3). By Step 2, since $\widehat{\boldsymbol{c}}$ is an α-approximate solution to problem (3), we have $f(\widehat{\boldsymbol{c}}) \geq \alpha f(\boldsymbol{c}^*)$. By Step 1, we have $f(\boldsymbol{c}^*) \geq J_1(\boldsymbol{c}^*)$. Collecting the preceding three inequalities in the proof, we obtain $\mathsf{Rev}(\widehat{\boldsymbol{c}}) \geq \beta\, f(\widehat{\boldsymbol{c}}) \geq \alpha\beta f(\boldsymbol{c}^*) \geq \alpha\beta J_1(\boldsymbol{c}^*) = \alpha\beta\,\mathsf{opt}$. ■

Thus, our approximation framework gets an $\alpha\beta$-approximate solution. In the next section, we give our surrogate and describe the performance guarantees attained by using this surrogate.

4. Surrogate Function and Main Results

In Step 1 of our approximation framework, we need to construct a surrogate function that approximates the value function at the first time period from above. To construct our surrogate, we use a linear programming approximation to the assortment customization problem that we formulated in the dynamic program in (1). In our linear programming approximation, we assume that the arrivals and choices of the customers take on their expected values and the customers can make purchases for fractional numbers of units of products. In particular, we use the decision variable $w_j(S)$ to capture the total expected number of times that we offer assortment S to a customer of type j over the whole selling horizon. For notational brevity, we use $\tau_j = \sum_{t \in \mathcal{T}} \lambda_{jt}$ to denote the total expected number of customer arrivals of type j over the selling horizon. In this case, using the vector of decision variables $\boldsymbol{w} = (w_j(S) : j \in \mathcal{M},\ S \subseteq \mathcal{N})$, our surrogate $f : \mathbb{Z}_+^n \to \mathbb{R}_+$ is given by the optimal objective value of the linear program

$$f(\boldsymbol{c}) \;=\; \max_{\boldsymbol{w} \in \mathbb{R}_+^{m\,2^n}} \left\{ \sum_{j \in \mathcal{M}} \sum_{S \subseteq \mathcal{N}} \sum_{i \in \mathcal{N}} r_i\, \phi_{ij}(S)\, w_j(S) \;:\; \sum_{j \in \mathcal{M}} \sum_{S \subseteq \mathcal{N}} \phi_{ij}(S)\, w_j(S) \leq c_i \quad \forall\, i \in \mathcal{N}, \quad \sum_{S \subseteq \mathcal{N}} w_j(S) \leq \tau_j \quad \forall\, j \in \mathcal{M} \right\}. \tag{4}$$

In the linear program above, if a customer of type j arrives at a time period and we offer assortment S, then we obtain an expected revenue of $\sum_{i \in \mathcal{N}} r_i\, \phi_{ij}(S)$. Thus, the objective function accounts for the total expected revenue over the selling horizon. If a customer of type j arrives at a time period and we offer assortment S, then the expected consumption of the inventory of product i is $\phi_{ij}(S)$, so the first constraint ensures that the total expected inventory consumption of product i does not exceed its stocking quantity. The second constraint ensures that the total expected number of customers of type j that are offered some assortment is at most the total expected number of arrivals. Similar linear programming approximations have been used by Gallego et al. (2004), Liu and van Ryzin (2008), Golrezaei et al. (2014) and Ma et al. (2021). We refer to the surrogate given by the optimal objective value of problem (4) as the *linear programming-based surrogate*.

Throughout the paper, we will use the linear programming-based surrogate and $f(\boldsymbol{c})$ will always denote the surrogate given by the optimal objective value of problem (4) as a function of $\boldsymbol{c}$. Problem (4) is a fluid approximation for the dynamic program in (1). It is a standard result that the optimal objective value of such a fluid approximation is an upper bound on the optimal total expected revenue. Proposition 2 in Gallego et al. (2004), for example, gives a proof of this result with a single customer type and we can extend their result to multiple customer types. Thus, we have $f(\boldsymbol{c}) \geq J_1(\boldsymbol{c})$, as needed from the surrogate in Step 1 of our approximation framework. The number of decision variables in (4) increases exponentially with the number of products, so we can solve this problem by using column generation. The column generation subproblem is of the form $\max_{S\subseteq\mathcal{N}}\{\sum_{i\in\mathcal{N}}\phi_{ij}(S)\,(r_i-\mu_i)\}$ for fixed dual multipliers $(\mu_i : i\in\mathcal{N})$ obtained through the master problem. We can solve this column generation subproblem efficiently under various choice models, such as the multinomial logit, nested logit, generalized attraction, Markov chain and a mixture of independent demand and multinomial logit models; see Talluri and van Ryzin (2004), Davis et al. (2014), Gallego et al. (2015), Blanchet et al. (2016) and Cao et al. (2023). We proceed with the understanding that we can solve problem (4) efficiently. Using the linear programming-based surrogate, we will execute Steps 2 and 3 of our approximation framework to get guarantees for our joint stocking and assortment customization problem, as explained next.

Outline and Main Results:

Using our approximation framework, we give two performance guarantees. To get either of the two performance guarantees, we use the linear programing-based surrogate (4) in Step 1, so $f(\boldsymbol{c})$ is always given by problem (4). Next, we consider Step 2. First, we assume that the customers choose under the multinomial logit model. In Section 5, we show that we can obtain a $\frac{1}{2}(1-\frac{1}{e})$-approximate solution $\widehat{\boldsymbol{c}}$ to problem (3) in polynomial time. Thus, we can execute Step 2 with $\alpha = \frac{1}{2}(1-\frac{1}{e})$ with the linear programming-based surrogate under the multinomial logit model. Second, we assume that the customers choose under a general choice model. We focus on the case where the storage capacity is large so that $K \geq n$. In Section 6, we show that we can obtain a $(1-\sqrt[3]{\frac{n}{K}})$-approximate solution $\widehat{\boldsymbol{c}}$ to problem (3) in polynomial time in such a way that the approximate solution satisfies $\widehat{c}_i \geq \frac{1}{2}\,(\frac{K}{n})^{2/3}$ for all $i \in \mathcal{N}$. Thus, we can execute Step 2 with $\alpha = 1 - \sqrt[3]{\frac{n}{K}}$ with the linear programming-based surrogate under a general choice model. Lastly, we consider Step 3 of our approximation framework. In Section 7, letting $\widehat{c}_{\min} = \min_{i\in\mathcal{N}}\{\widehat{c}_i : \widehat{c}_i \geq 1\}$ to capture the smallest non-zero stocking quantity in the initial inventory vector $\widehat{\boldsymbol{c}}$, we give an assortment customization policy such that the total expected revenue of the policy with initial inventories $\widehat{\boldsymbol{c}}$ is at least $\max\{\frac{1}{2}, 1 - \frac{1}{\sqrt{\widehat{c}_{\min}}}\}\, f(\widehat{\boldsymbol{c}})$, when the customers choose under a general choice model. Thus, we can execute Step 3 with $\beta = \max\{\frac{1}{2}, 1 - \frac{1}{\sqrt{\widehat{c}_{\min}}}\}$. Using these results in Theorem 3.1, we get the following two performance guarantees for our joint stocking and assortment customization problem.

Theorem 4.1 (Performance Guarantees) *We have the following two performance guarantees for the joint stocking and assortment customization problem.*

(*a*) *Under the multinomial logit model, we can compute a* $\frac{1}{4}\,(1 - \frac{1}{e})$*-approximate solution in polynomial time.*

(*b*) *Under a general choice model, considering the case with large storage capacity so that* $K \geq n$*, we can compute a* $(1 - (\sqrt{2}+1)\,\sqrt[3]{\frac{n}{K}})$*-approximate solution in polynomial time.*

Proof: By the discussion right before the theorem, under the multinomial logit model, using the linear programming-based surrogate, we can execute Step 2 with $\alpha = \frac{1}{2}\,(1 - \frac{1}{e})$ and Step 3 with $\beta = \frac{1}{2}$. Thus, by Theorem 3.1, we get a performance guarantee of $\frac{1}{4}\,(1 - \frac{1}{e})$, so the first part of the theorem follows. Under a general choice model, we can execute Step 2 with $\alpha = 1 - \sqrt[3]{\frac{n}{K}}$ to obtain an approximate solution $\widehat{\boldsymbol{c}}$ to problem (3) that satisfies $\widehat{c}_i \geq \frac{1}{2}\,(\frac{K}{n})^{2/3}$ for all $i \in \mathcal{N}$. Furthermore, letting $\widehat{c}_{\min} = \min_{i\in\mathcal{N}}\{\widehat{c}_i : \widehat{c}_i \geq 1\}$, we can execute Step 3 with $\beta = 1 - \frac{1}{\sqrt{\widehat{c}_{\min}}}$. Thus, by Theorem 3.1, we get a performance guarantee of $(1 - \sqrt[3]{\frac{n}{K}})\,(1 - \frac{1}{\sqrt{\widehat{c}_{\min}}})$. Noting that $\widehat{c}_i \geq \frac{1}{2}(\frac{K}{n})^{2/3}$ for all $i \in \mathcal{N}$, we have $\widehat{c}_{\min} \geq \frac{1}{2}(\frac{K}{n})^{2/3}$. In this case, the last performance guarantee satisfies $(1 - \sqrt[3]{\frac{n}{K}})\,(1 - \frac{1}{\sqrt{\widehat{c}_{\min}}}) \geq (1 - \sqrt[3]{\frac{n}{K}})\,(1 - \sqrt{2}\,\sqrt[3]{\frac{n}{K}}) \geq 1 - (\sqrt{2}+1)\,\sqrt[3]{\frac{n}{K}}$, so the second part of the theorem follows. ■

By the first part of the theorem, if the customers choose according to the multinomial logit model, then we can obtain a solution with a constant-factor performance guarantee. By the second part of the theorem, even when the customers choose according to a general choice model, as the storage capacity gets arbitrarily large, we obtain an asymptotically optimal solution. The performance guarantee in the second part depends only on n and K. By the discussion in the proof of the theorem, we actually obtain a performance guarantee of $(1 - \sqrt[3]{\frac{n}{K}})\,(1 - \sqrt{2}\,\sqrt[3]{\frac{n}{K}})$ under a general choice model, which we lower bounded by $1 - (\sqrt{2}+1)\,\sqrt[3]{\frac{n}{K}}$. In Figure 1, we plot $(1 - \sqrt[3]{\frac{n}{K}})\,(1 - \sqrt{2}\,\sqrt[3]{\frac{n}{K}})$ as a function of K/n. We interpret K/n as the average storage capacity available for a product. When $K/n = 70$, for example, we get a performance guarantee of about $1/2$,

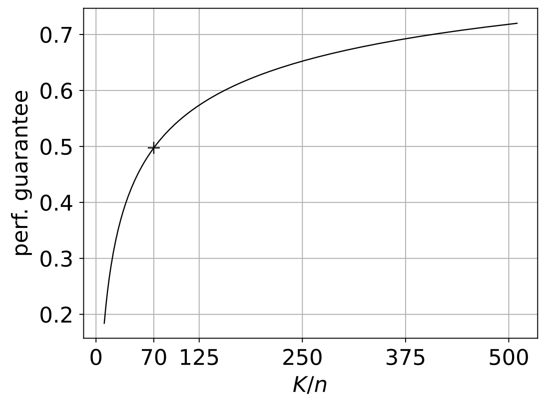

Figure 1 Performance guarantee provided by our approximation framework as a function of K/n.

which is marked by the cross. In our computational experiments, the practical performance of our approximation framework is substantially better than its theoretical performance guarantee. Note that our asymptotic optimality result is different from other asymptotic optimality results in the revenue management literature, which show that if the initial inventories of all products grow arbitrarily large, then we can design asymptotically optimal policies for various dynamic pricing and capacity control problems; see Gallego and van Ryzin (1994), Jasin and Kumar (2012), Feng et al. (2020) and Ma et al. (2021). In our result, the storage capacity grows arbitrarily large, which does not necessarily imply that the initial inventories for all products grow large.

5. Stocking Decisions under the Multinomial Logit Model

We consider finding an approximate solution to problem (3) in Step 2 of our approximation framework when the customers choose according to the multinomial logit model. Under the multinomial logit model, customers of type j associate the preference weight v_{ij} with product i. We normalize the preference weight of the no-purchase option to one. If we offer the assortment S, then a customer of type j chooses product $i \in S$ with probability $\phi_{ij}(S) = v_{ij}/(1 + \sum_{k \in S} v_{kj})$; see McFadden (1974) and Train (2003). In the next theorem, we show that if we use the linear programming-based surrogate in problem (3), then it is NP-hard to approximate problem (3) within a factor of $1 - \frac{1}{e}$ even when the customers choose under the multinomial logit model.

Theorem 5.1 (Complexity) *Even when the customers choose under the multinomial logit model and all product revenues are equal to each other, there exists no polynomial-time algorithm to approximate problem* (3) *within a factor of* $1 - \frac{1}{e} - \epsilon$ *for all* $\epsilon > 0$*, unless* $P = NP$.

The proof of the theorem is in Appendix B and it uses a reduction from the maximum coverage problem. Thus, we focus on obtaining approximate solutions to problem (3). We will use submodular function maximization tools to obtain an approximate solution to problem (3). The function $g:\mathbb{Z}_+^n \to \mathbb{R}_+$ is said to be submodular if it satisfies $g(\boldsymbol{c}+\boldsymbol{e}_i)-g(\boldsymbol{c}) \le g(\boldsymbol{b}+\boldsymbol{e}_i)-g(\boldsymbol{b})$ for all $i \in \mathcal{N}$ and $\boldsymbol{c},\boldsymbol{b} \in \mathbb{Z}_+^n$ with $\boldsymbol{c} \ge \boldsymbol{b}$. On the other hand, the function $g:\mathbb{Z}_+^n \to \mathbb{R}_+$ is said to be monotone if it satisfies $g(\boldsymbol{c}) \ge g(\boldsymbol{b})$ for all $\boldsymbol{c},\boldsymbol{b} \in \mathbb{Z}_+^n$ with $\boldsymbol{c} \ge \boldsymbol{b}$. If the linear programming-based surrogate $f(\boldsymbol{c})$ were monotone and submodular in $\boldsymbol{c}$, then problem (3) would be maximizing a monotone and submodular function over the integer lattice subject to a cardinality constraint. It is known that such a monotone and submodular function maximization problem admits a $(1-\frac{1}{e})$-approximation; see Soma and Yoshida (2018). Unfortunately, we can come up with examples to show that the linear programming-based surrogate is not submodular even when the customers choose under the multinomial logit model. We give one example in Appendix C.

To get around the fact that the linear programming-based surrogate is not submodular, we will construct a monotone and submodular approximation to the linear programming-based surrogate and leverage this approximation. Our starting point for constructing the monotone and submodular approximation is an alternative formulation of the linear programming-based surrogate under the multinomial logit model, which we borrow from the earlier literature. In the alternative formulation, we use the decision variable y_{ij} to capture the total expected number of customers of type j making a purchase for product i, whereas we use the decision variable y_{0j} to capture the total expected number of customers of type j leaving without a purchase. In this case, using the vectors of decision variables $\boldsymbol{y}=(y_{ij}: i \in \mathcal{N},\ j \in \mathcal{M})$ and $\boldsymbol{y}_0=(y_{0j}: j \in \mathcal{M})$, we consider the linear program

$$f(\boldsymbol{c}) = \max_{(\boldsymbol{y},\boldsymbol{y}_0)\in\mathbb{R}_+^{nm+m}} \Bigg\{ \sum_{j\in\mathcal{M}}\sum_{i\in\mathcal{N}} r_i\, y_{ij} \ :\ \sum_{j\in\mathcal{M}} y_{ij} \le c_i \quad \forall\, i \in \mathcal{N}, \qquad (5)$$

$$\sum_{i\in\mathcal{N}} y_{ij} + y_{0j} \le \tau_j \quad \forall\, j \in \mathcal{M}, \qquad \frac{y_{ij}}{v_{ij}} \le y_{0j} \quad \forall\, i \in \mathcal{N},\ j \in \mathcal{M} \Bigg\}.$$

We can show that if the choice probabilities are of the form $\phi_{ij}(S)=v_{ij}/(1+\sum_{k\in S} v_{kj})$, then problems (4) and (5) have the same optimal objective value; see Gallego et al. (2015). Thus, we continue using $f(\boldsymbol{c})$ to denote the optimal objective value of problem (5) as a function of $\boldsymbol{c}$. To interpret problem (5), noting the definition of the decision variable y_{ij}, the objective function is the total expected revenue. The first constraint ensures that the total expected purchases for product i by all customer types does not exceed the stocking quantity of the product. The second constraint ensures that the total expected number of customers of type j either purchasing some product or leaving without a purchase is at most the total expected number of arrivals. The third constraint, roughly speaking, aligns the product purchases with the multinomial logit model.

We can compute the linear programming-based surrogate by solving problem (5) instead of problem (4), but whether we use problem (4) or (5), as discussed earlier in this section, $f(\boldsymbol{c})$ is not submodular in $\boldsymbol{c}$. We consider an approximation to $f(\boldsymbol{c})$ that is obtained by fixing the value of the decision variable y_{0j} in (5) at $\tau_j/2$. So, consider the linear program

$$f_{\mathsf{app}}(\boldsymbol{c}) \;=\; \max_{\boldsymbol{y}\in\mathbb{R}_+^{nm}} \left\{ \sum_{j\in\mathcal{M}}\sum_{i\in\mathcal{N}} r_i\, y_{ij} \;:\; \sum_{j\in\mathcal{M}} y_{ij} \le c_i \quad \forall\, i\in\mathcal{N}, \right. \tag{6}$$
$$\left. \sum_{i\in\mathcal{N}} y_{ij} \le \frac{1}{2}\tau_j \quad \forall\, j\in\mathcal{M}, \qquad y_{ij} \le \frac{1}{2} v_{ij}\,\tau_j \quad \forall\, i\in\mathcal{N},\; j\in\mathcal{M} \right\}.$$

To motivate fixing $y_{0j} = \tau_j/2$, note that we obtain problem (6) by fixing the values of the decision variables $(y_{0j} : j \in \mathcal{M})$. Thus, the optimal objective value of problem (6) is a lower bound on that of problem (5). On the other hand, writing the second constraint in problem (5) as $\sum_{i\in\mathcal{N}} y_{ij} \le \tau_j - y_{0j}$ for all $j\in\mathcal{M}$, the right side of the second constraint in (5) is decreasing y_{0j}, whereas the right side of the third constraint in (5) is increasing in y_{0j}. Thus, if we fix the value of y_{0j} too large, then the lower bound provided by problem (6) becomes too loose because we tighten the second constraint in (5) too much, whereas if we fix the value of y_{0j} too small, then the lower bound provided by (6) becomes too loose because we tighten the third constraint in (5) too much. Fixing $y_{0j} = \tau_j/2$ allows us to tradeoff between these two conflicting goals.

In the next lemma, we show that the optimal objective value of problem (6) approximates the linear programming-based surrogate within a factor of two.

Lemma 5.2 (Surrogate Approximation) *Noting that the optimal objective value of problem (6) as a function of* $\boldsymbol{c}$ *is* $f_{\mathsf{app}}(\boldsymbol{c})$, *we have* $\frac{1}{2}f(\boldsymbol{c}) \le f_{\mathsf{app}}(\boldsymbol{c}) \le f(\boldsymbol{c})$ *for all* $\boldsymbol{c}\in\mathbb{Z}_+^n$.

Proof: We obtain problem (6) by fixing the values of the decision variables $(y_{0j} : j\in\mathcal{M})$ in problem (5), so we immediately have $f_{\mathsf{app}}(\boldsymbol{c}) \le f(\boldsymbol{c})$. Let $(\boldsymbol{y}^*, \boldsymbol{y}_0^*)$ be an optimal solution to problem (5). We claim that $\frac{1}{2}\boldsymbol{y}^*$ is a feasible solution to problem (6). Because the solution $(\boldsymbol{y}^*, \boldsymbol{y}_0^*)$ is feasible to problem (5), we have $\sum_{j\in\mathcal{M}} y^*_{ij} \le c_i$ for all $i\in\mathcal{N}$, $\sum_{i\in\mathcal{N}} y^*_{ij} \le \tau_j$ and $y^*_{0j} \le \tau_j$ for all $j\in\mathcal{M}$ and $y^*_{ij} \le v_{ij}\, y^*_{0j}$ for all $i\in\mathcal{N}$ and $j\in\mathcal{M}$, where the second and third inequalities use the second constraint in problem (5). In this case, these four inequalities imply that we have $\sum_{j\in\mathcal{M}} \frac{1}{2} y^*_{ij} \le c_i$, $\sum_{i\in\mathcal{N}} \frac{1}{2} y^*_{ij} \le \frac{1}{2}\tau_j$ and $\frac{1}{2} y^*_{ij} \le \frac{1}{2} v_{ij}\, y^*_{0j} \le \frac{1}{2} v_{ij}\,\tau_j$. Thus, it follows that $\frac{1}{2}\boldsymbol{y}^*$ is a feasible solution to problem (6). In this case, noting that $\frac{1}{2}\boldsymbol{y}^*$ is a feasible but not necessarily an optimal solution to problem (6), we obtain $f_{\mathsf{app}}(\boldsymbol{c}) \ge \frac{1}{2}\sum_{j\in\mathcal{M}}\sum_{i\in\mathcal{N}} r_i\, y^*_{ij} = \frac{1}{2} f(\boldsymbol{c})$. ∎

We refer to the surrogate given by the optimal objective value of (6) as the *approximate surrogate*. By Lemma 5.2, $f_{\mathsf{app}}(\boldsymbol{c})$ approximates $f(\boldsymbol{c})$ within a factor of two. Thus, if we replace $f(\boldsymbol{c})$ in problem (3) with $f_{\mathsf{app}}(\boldsymbol{c})$ and obtain an α-approximate solution, then this solution is a $\frac{1}{2}\alpha$-approximate solution to problem (3). The approximate surrogate is monotone because if we increase the right side of the first constraint in (6), then the optimal objective value of this problem does not decrease. In the remainder of this section, we show that the approximate surrogate is submodular.

Submodularity of the Approximate Surrogate:

We proceed to showing that $f_{\mathsf{app}}(\boldsymbol{c})$ is submodular in $\boldsymbol{c}$. Using the dual variables $\boldsymbol{\mu} = (\mu_i : i \in \mathcal{N})$, $\boldsymbol{\sigma} = (\sigma_j : j \in \mathcal{M})$ and $\boldsymbol{\theta} = (\theta_{ij} : i \in \mathcal{N},\ j \in \mathcal{M})$ for the three constraints, the dual of (6) is

$$\begin{aligned} f_{\mathsf{app}}(\boldsymbol{c}) &= \min_{(\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{\theta}) \in \mathbb{R}_+^{n+m+nm}} \left\{ \sum_{i \in \mathcal{N}} c_i\, \mu_i + \sum_{j \in \mathcal{M}} \frac{\tau_j}{2}\, \sigma_j + \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} \frac{v_{ij}\, \tau_j}{2}\, \theta_{ij} \ :\ \mu_i + \sigma_j + \theta_{ij} \geq r_i \ \forall\, i \in \mathcal{N},\ j \in \mathcal{M} \right\} \\ &= \min_{\boldsymbol{\mu} \in \mathbb{R}_+^n} \left\{ \sum_{i \in \mathcal{N}} c_i\, \mu_i + \sum_{j \in \mathcal{M}} \frac{\tau_j}{2} \min_{(\sigma_j, \boldsymbol{\theta}_j) \in \mathbb{R}_+^{1+n}} \left\{ \sigma_j + \sum_{i \in \mathcal{N}} v_{ij}\, \theta_{ij} \ :\ \sigma_j + \theta_{ij} \geq r_i - \mu_i \ \ \forall\, i \in \mathcal{N} \right\} \right\}, \end{aligned}$$

where the second equality follows by minimizing over the decision variables $\boldsymbol{\mu}$ in the outer problem and over the decision variables $(\boldsymbol{\sigma}, \boldsymbol{\theta})$ in the inner problem, as well as noting that the inner problem decomposes by the customer types. On the right side of the chain of equalities above, we use the vector of decision variables $\boldsymbol{\theta}_j = (\theta_{ij} : i \in \mathcal{N})$. In this case, letting $L(\boldsymbol{c}, \boldsymbol{\mu}) = \sum_{i \in \mathcal{N}} c_i\, \mu_i$ and using $G_j(\boldsymbol{\mu})$ to denote the optimal objective value of the inner minimization problem on the right side of the chain of equalities above, we have $f_{\mathsf{app}}(\boldsymbol{c}) = \min_{\boldsymbol{\mu} \in \mathbb{R}_+^n} \big\{ L(\boldsymbol{c}, \boldsymbol{\mu}) + \sum_{j \in \mathcal{M}} \frac{\tau_j}{2}\, G_j(\boldsymbol{\mu}) \big\}$. By the definition of $L(\boldsymbol{c}, \boldsymbol{\mu})$, we have $L(\boldsymbol{c} + \boldsymbol{e}_i, \boldsymbol{\mu}) = L(\boldsymbol{c}, \boldsymbol{\mu}) + \mu_i$. Furthermore, consider solving the inner minimization problem on the right side of the chain of inequalities above by using its dual. Associating the dual variables $\boldsymbol{z}_j = (z_{ij} : i \in \mathcal{N})$ with the constraints in the inner minimization problem and noting that we use $G_j(\boldsymbol{\mu})$ to denote the optimal objective value of this problem, we have $G_j(\boldsymbol{\mu}) = \max_{\boldsymbol{z}_j \in \mathbb{R}_+^n} \big\{ \sum_{i \in \mathcal{N}} (r_i - \mu_i)\, z_{ij} : \sum_{i \in \mathcal{N}} z_{ij} \leq 1,\ z_{ij} \leq v_{ij}\ \forall\, i \in \mathcal{N} \big\}$, which is a knapsack problem. Therefore, $G_j(\boldsymbol{\mu})$ corresponds to the optimal objective value of a knapsack problem when viewed as a function of its objective function coefficients, establishing a relationship between our approximate surrogate and the knapsack problem. In the next theorem, we build on these observations to show that the approximate surrogate is submodular.

Theorem 5.3 (Submodularity of Approximate Surrogate) *For each $i \in \mathcal{N}$ and $\boldsymbol{c}, \boldsymbol{b} \in \mathbb{Z}_+^n$ that satisfies $\boldsymbol{c} \geq \boldsymbol{b}$, we have $f_{\mathsf{app}}(\boldsymbol{c} + \boldsymbol{e}_i) - f_{\mathsf{app}}(\boldsymbol{c}) \leq f_{\mathsf{app}}(\boldsymbol{b} + \boldsymbol{e}_i) - f_{\mathsf{app}}(\boldsymbol{b})$.*

Proof: We can compute the approximate surrogate as $f_{\mathsf{app}}(\boldsymbol{c}) = \min_{\boldsymbol{\mu} \in \mathbb{R}_+^n} \big\{ L(\boldsymbol{c}, \boldsymbol{\mu}) + \sum_{j \in \mathcal{M}} \frac{\tau_j}{2}\, G_j(\boldsymbol{\mu}) \big\}$. We define the notation $\boldsymbol{\mu} \vee \boldsymbol{\eta} = (\mu_i \vee \eta_i : i \in \mathcal{N})$ and $\boldsymbol{\mu} \wedge \boldsymbol{\eta} = (\mu_i \wedge \eta_i : i \in \mathcal{N})$ for $\boldsymbol{\mu}, \boldsymbol{\eta} \in \mathbb{R}_+^n$. An auxiliary lemma, given as Lemma D.1 in Appendix D, shows that the function $L(\boldsymbol{c}, \boldsymbol{\mu})$ satisfies $L(\boldsymbol{c}, \boldsymbol{\mu}) + L(\boldsymbol{b}, \boldsymbol{\eta}) \geq L(\boldsymbol{c}, \boldsymbol{\mu} \wedge \boldsymbol{\eta}) + L(\boldsymbol{b}, \boldsymbol{\mu} \vee \boldsymbol{\eta})$ for all $\boldsymbol{c}, \boldsymbol{b} \in \mathbb{R}_+^n$ with $\boldsymbol{c} \geq \boldsymbol{b}$ and $\boldsymbol{\mu}, \boldsymbol{\eta} \in \mathbb{R}_+^n$, whereas the function $G_j(\boldsymbol{\mu})$ satisfies $G_j(\boldsymbol{\mu}) + G_j(\boldsymbol{\eta}) \geq G_j(\boldsymbol{\mu} \wedge \boldsymbol{\eta}) + G_j(\boldsymbol{\mu} \vee \boldsymbol{\eta})$ for all $\boldsymbol{\mu}, \boldsymbol{\eta} \in \mathbb{R}_+^n$. The proof of the first inequality uses the fact that $L(\boldsymbol{c}, \boldsymbol{\mu})$ is a bilinear function of the form $L(\boldsymbol{c}, \boldsymbol{\mu}) = \sum_{i \in \mathcal{N}} c_i\, \mu_i$, whereas the proof of the second inequality uses the fact that $G_j(\boldsymbol{\mu})$

is the optimal objective value of a knapsack problem when viewed as a function of the objective function coefficients. Thus, defining $F(\boldsymbol{c},\boldsymbol{\mu}) = L(\boldsymbol{c},\boldsymbol{\mu}) + \sum_{j\in\mathcal{M}} \frac{\tau_j}{2}\, G_j(\boldsymbol{\mu})$ for notational brevity, the function $F(\boldsymbol{c},\boldsymbol{\mu})$ also satisfies $F(\boldsymbol{c},\boldsymbol{\mu}) + F(\boldsymbol{b},\boldsymbol{\eta}) \geq F(\boldsymbol{c},\boldsymbol{\mu}\wedge\boldsymbol{\eta}) + F(\boldsymbol{b},\boldsymbol{\mu}\vee\boldsymbol{\eta})$ for all $\boldsymbol{c},\boldsymbol{b}\in\mathbb{R}^n_+$ with $\boldsymbol{c}\geq\boldsymbol{b}$ and $\boldsymbol{\mu},\boldsymbol{\eta}\in\mathbb{R}^n_+$. Furthermore, by the definition of $F(\boldsymbol{c},\boldsymbol{\mu})$, we have $f_{\mathsf{app}}(\boldsymbol{c}) = \min_{\boldsymbol{\mu}\in\mathbb{R}^n_+} F(\boldsymbol{c},\boldsymbol{\mu})$. We make three useful observations. First, by the discussion right before the theorem, we have $L(\boldsymbol{c}+\boldsymbol{e}_i,\boldsymbol{\mu}) = L(\boldsymbol{c},\boldsymbol{\mu}) + \mu_i$, so we have $F(\boldsymbol{c}+\boldsymbol{e}_i,\boldsymbol{\mu}) = F(\boldsymbol{c},\boldsymbol{\mu}) + \mu_i$ as well. Second, we define $\widehat{\boldsymbol{\mu}}_{\boldsymbol{c}} = \arg\min_{\boldsymbol{\mu}\in\mathbb{R}^n_+} F(\boldsymbol{c},\boldsymbol{\mu})$, in which case, we have $f_{\mathsf{app}}(\boldsymbol{c}) = F(\boldsymbol{c},\widehat{\boldsymbol{\mu}}_{\boldsymbol{c}})$. Similarly, we define $\widehat{\boldsymbol{\mu}}^+_{\boldsymbol{c}} = \arg\min_{\boldsymbol{\mu}\in\mathbb{R}^n_+} F(\boldsymbol{c}+\boldsymbol{e}_i,\boldsymbol{\mu})$, $\widehat{\boldsymbol{\mu}}_{\boldsymbol{b}} = \arg\min_{\boldsymbol{\mu}\in\mathbb{R}^n_+} F(\boldsymbol{b},\boldsymbol{\mu})$ and $\widehat{\boldsymbol{\mu}}^+_{\boldsymbol{b}} = \arg\min_{\boldsymbol{\mu}\in\mathbb{R}^n_+} F(\boldsymbol{b}+\boldsymbol{e}_i,\boldsymbol{\mu})$. Third, noting that $\widehat{\boldsymbol{\mu}}_{\boldsymbol{b}} = \arg\min_{\boldsymbol{\mu}\in\mathbb{R}^n_+} F(\boldsymbol{b},\boldsymbol{\mu})$ and $\widehat{\boldsymbol{\mu}}^+_{\boldsymbol{c}} = \arg\min_{\boldsymbol{\mu}\in\mathbb{R}^n_+} F(\boldsymbol{c}+\boldsymbol{e}_i,\boldsymbol{\mu})$, we have $F(\boldsymbol{b},\widehat{\boldsymbol{\mu}}_{\boldsymbol{b}}) \leq F(\boldsymbol{b},\boldsymbol{\mu})$ and $F(\boldsymbol{c}+\boldsymbol{e}_i,\widehat{\boldsymbol{\mu}}^+_{\boldsymbol{c}}) \leq F(\boldsymbol{c}+\boldsymbol{e}_i,\boldsymbol{\mu})$ for all $\boldsymbol{\mu}\in\mathbb{R}^n_+$. In this case, using $\widehat{\mu}^+_{i,\boldsymbol{b}}$ to denote the i-th component of the vector $\widehat{\boldsymbol{\mu}}^+_{\boldsymbol{b}}$, we obtain

$$
\begin{aligned}
&f_{\mathsf{app}}(\boldsymbol{c}) + f_{\mathsf{app}}(\boldsymbol{b}+\boldsymbol{e}_i) \overset{(a)}{=} F(\boldsymbol{c},\widehat{\boldsymbol{\mu}}_{\boldsymbol{c}}) + F(\boldsymbol{b}+\boldsymbol{e}_i,\widehat{\boldsymbol{\mu}}^+_{\boldsymbol{b}}) \\
&\quad\overset{(b)}{=} F(\boldsymbol{c},\widehat{\boldsymbol{\mu}}_{\boldsymbol{c}}) + F(\boldsymbol{b},\widehat{\boldsymbol{\mu}}^+_{\boldsymbol{b}}) + \widehat{\mu}^+_{i,\boldsymbol{b}} \overset{(c)}{\geq} F(\boldsymbol{c},\widehat{\boldsymbol{\mu}}_{\boldsymbol{c}}\wedge\widehat{\boldsymbol{\mu}}^+_{\boldsymbol{b}}) + F(\boldsymbol{b},\widehat{\boldsymbol{\mu}}_{\boldsymbol{c}}\vee\widehat{\boldsymbol{\mu}}^+_{\boldsymbol{b}}) + \widehat{\mu}^+_{i,\boldsymbol{b}} \\
&\quad\geq F(\boldsymbol{c},\widehat{\boldsymbol{\mu}}_{\boldsymbol{c}}\wedge\widehat{\boldsymbol{\mu}}^+_{\boldsymbol{b}}) + F(\boldsymbol{b},\widehat{\boldsymbol{\mu}}_{\boldsymbol{c}}\vee\widehat{\boldsymbol{\mu}}^+_{\boldsymbol{b}}) + (\widehat{\mu}_{i,\boldsymbol{c}}\wedge\widehat{\mu}^+_{i,\boldsymbol{b}}) \overset{(d)}{=} F(\boldsymbol{c}+\boldsymbol{e}_i,\widehat{\boldsymbol{\mu}}_{\boldsymbol{c}}\wedge\widehat{\boldsymbol{\mu}}^+_{\boldsymbol{b}}) + F(\boldsymbol{b},\widehat{\boldsymbol{\mu}}_{\boldsymbol{c}}\vee\widehat{\boldsymbol{\mu}}^+_{\boldsymbol{b}}) \\
&\quad\overset{(e)}{\geq} F(\boldsymbol{c}+\boldsymbol{e}_i,\widehat{\boldsymbol{\mu}}^+_{\boldsymbol{c}}) + F(\boldsymbol{b},\widehat{\boldsymbol{\mu}}_{\boldsymbol{b}}) \overset{(f)}{=} f_{\mathsf{app}}(\boldsymbol{c}+\boldsymbol{e}_i) + f_{\mathsf{app}}(\boldsymbol{b}),
\end{aligned}
$$

where (a) and (f) use the second observation, (b) and (d) use the first observation, (c) holds because $F(\boldsymbol{c},\boldsymbol{\mu}) + F(\boldsymbol{b},\boldsymbol{\eta}) \geq F(\boldsymbol{c},\boldsymbol{\mu}\wedge\boldsymbol{\eta}) + F(\boldsymbol{b},\boldsymbol{\mu}\vee\boldsymbol{\eta})$ and (e) uses the third observation. ■

Problem (6) is the bipartite transportation problem, where the flow from node i to node j is upper bounded by $\frac{1}{2}\, v_{ij}\, \tau_j$. If the flows do not have upper bounds, then the optimal objective value of the bipartite transportation problem is known to be submodular in the availability at the supply nodes; see Theorem 3 in Nemhauser et al. (1978). The analogue of this result in Theorem 5.3, allowing upper bounds, is new. Since $f_{\mathsf{app}}(\boldsymbol{c})$ is monotone and submodular in $\boldsymbol{c}$, if we replace the objective function of problem (3) with $f_{\mathsf{app}}(\boldsymbol{c})$, then we can obtain a $(1-\frac{1}{e})$-approximate solution to this problem in polynomial time; see Soma and Yoshida (2018). Nemhauser et al. (1978) show that a greedy algorithm also yields a $(1-\frac{1}{e})$-approximate solution to the same problem. In the greedy algorithm, we start with the solution $\widehat{\boldsymbol{c}}^1 = \boldsymbol{0}\in\mathbb{Z}^n_+$. For each $k = 1,\ldots,K$, we set $\widehat{i}^k = \arg\max_{i\in\mathcal{N}} f_{\mathsf{app}}(\widehat{\boldsymbol{c}}^k + \boldsymbol{e}_i)$ and $\widehat{\boldsymbol{c}}^{k+1} = \widehat{\boldsymbol{c}}^k + e_{\widehat{i}^k}$. Since $f_{\mathsf{app}}(\boldsymbol{c})$ approximates $f(\boldsymbol{c})$ within a factor of two, the resulting solution is a $\frac{1}{2}(1-\frac{1}{e})$-approximate solution to problem (3).

We can efficiently get a $(1-\frac{1}{e})$-approximation to the problem of maximizing a monotone and submodular function subject to a knapsack constraint; see Sviridenko (2004) and Soma and Yoshida (2018). So, letting σ_i be the space consumption of a unit of product i, we can use our approximation framework under a knapsack constraint of the form $\sum_{i\in\mathcal{N}} \sigma_i\, c_i \leq K$. In Appendix E, we show that we can also incorporate salvage values for the units left at the end of the selling horizon by changing our accounting of the total expected revenue and modifying our surrogate accordingly.

6. Stocking Decisions under a General Choice Model

We consider obtaining an approximate solution to problem (3) in Step 2 of our approximation framework when the customers choose according to a general choice model. We focus on the case with large storage capacity, so that we have $K \geq n$. Our approach is based on formulating a continuous relaxation of problem (3) and rounding the optimal solution to the continuous relaxation. In particular, we use the decision variable $w_j(S)$ to capture the total expected number of times that we offer the assortment S to customers of type j, whereas we use the decision variable c_i to capture the number of units of product i that we stock. We used the analogue of the decision variable $w_j(S)$ when formulating our linear programming-based surrogate in (4). Using the vectors of decision variables $\boldsymbol{w} = (w_j(S) : j \in \mathcal{M},\ S \subseteq \mathcal{N})$ and $\boldsymbol{c} = (c_i : i \in \mathcal{N})$, we consider a continuous relaxation of problem (3) given by the linear program

$$\mathsf{relax} \;=\; \max_{(\boldsymbol{w},\boldsymbol{c}) \in \mathbb{R}_+^{m\,2^n+n}} \left\{ \sum_{j\in\mathcal{M}} \sum_{S\subseteq\mathcal{N}} \sum_{i\in\mathcal{N}} r_i\, \phi_{ij}(S)\, w_j(S) \;:\; \sum_{j\in\mathcal{M}} \sum_{S\subseteq\mathcal{N}} \phi_{ij}(S)\, w_j(S) \leq c_i \quad \forall\, i \in \mathcal{N}, \right. \tag{7}$$
$$\left. \sum_{S\subseteq\mathcal{N}} w_j(S) \leq \tau_j \quad \forall\, j \in \mathcal{M}, \quad \sum_{i\in\mathcal{N}} c_i \leq K \right\}.$$

The number of decision variables above increases exponentially with the number of products, but the column generation subproblem for (7) has the same structure as the column generation subproblem for (4). Thus, we can solve problem (7) efficiently under a variety of choice models. To see that problem (7) is a continuous relaxation of problem (3), letting $\boldsymbol{c}^*$ be an optimal solution to problem (3) and noting that we use app to denote the optimal objective value of problem (3), we have $f(\boldsymbol{c}^*) = \mathsf{app}$. Let $\boldsymbol{w}^*$ be an optimal solution to problem (4) when we solve this problem with $\boldsymbol{c} = \boldsymbol{c}^*$, so we have $f(\boldsymbol{c}^*) = \sum_{j\in\mathcal{M}} \sum_{S\subseteq\mathcal{N}} \sum_{i\in\mathcal{N}} r_i\, \phi_{ij}(S)\, w_j^*(S)$ as well. By the two constraints in problem (4) with $\boldsymbol{c} = \boldsymbol{c}^*$ and the constraint in problem (3), the solution $(\boldsymbol{w}^*, \boldsymbol{c}^*)$ is feasible to problem (7) and provides an objective value of $\sum_{j\in\mathcal{M}} \sum_{S\subseteq\mathcal{N}} \sum_{i\in\mathcal{N}} r_i\, \phi_{ij}(S)\, w_j^*(S) = f(\boldsymbol{c}^*) = \mathsf{app}$. Thus, the optimal objective value of problem (7) is at least as large as that of problem (3).

In the next theorem, we show that we can perform rounding on an optimal solution to problem (7) to obtain a solution to problem (3) with a performance guarantee.

Theorem 6.1 (Stocking under General Choice) *Letting $(\boldsymbol{w}^*, \boldsymbol{c}^*)$ be an optimal solution to problem (7), for any integer $\gamma \in [1, \frac{K}{n}]$, let $\widehat{c}_i = \lfloor (1 - \gamma \frac{n}{K})\, c_i^* \rfloor + \gamma$ for all $i \in \mathcal{N}$. In this case, $\widehat{\boldsymbol{c}} = (\widehat{c}_i : i \in \mathcal{N})$ is a $(1 - \gamma \frac{n}{K})$-approximate solution to problem (3).*

Proof: Using the optimal solution $(\boldsymbol{w}^*, \boldsymbol{c}^*)$ to problem (7), let $\widehat{w}_j(S) = (1 - \gamma \frac{n}{K})\, w_j^*(S)$ for all $j \in \mathcal{M}$ and $S \subseteq \mathcal{N}$. Letting $\widehat{\boldsymbol{c}}$ be as in the theorem, using the vector $\widehat{\boldsymbol{w}} = (\widehat{w}_j(S) : j \in \mathcal{M},\ S \subseteq \mathcal{N})$ as we just defined, we claim that the solution $\widehat{\boldsymbol{w}}$ is feasible to problem (4) when we solve problem (4) with $\boldsymbol{c}=\widehat{\boldsymbol{c}}$. In particular, we have the chain of inequalities

$$\sum_{j\in\mathcal{M}}\sum_{S\subseteq\mathcal{N}}\phi_{ij}(S)\,\widehat{w}_j(S) \stackrel{(a)}{=} \left(1-\gamma\frac{n}{K}\right)\sum_{j\in\mathcal{M}}\sum_{S\subseteq\mathcal{N}}\phi_{ij}(S)\,w_j^*(S) \stackrel{(b)}{\le} \left(1-\gamma\frac{n}{K}\right)c_i^* \stackrel{(c)}{\le} \left\lfloor\left(1-\gamma\frac{n}{K}\right)c_i^*\right\rfloor+\gamma \stackrel{(d)}{=} \widehat{c}_i,$$

where (a) uses the definition of $\widehat{w}_j(S)$, (b) holds because $(\boldsymbol{w}^*,\boldsymbol{c}^*)$ is a feasible solution to problem (7), (c) holds since $x\le\lfloor x\rfloor+1\le\lfloor x\rfloor+\gamma$ and (d) uses the definition of $\widehat{c}_i$. Thus, the solution $\widehat{\boldsymbol{w}}$ satisfies the first constraint in problem (4) when we solve this problem with $\boldsymbol{c}=\widehat{\boldsymbol{c}}$. Furthermore, we have $\sum_{S\subseteq\mathcal{N}}\widehat{w}_j(S)=(1-\gamma\frac{n}{K})\sum_{S\subseteq\mathcal{N}}w_j^*(S)\le\tau_j$, where the inequality holds because $(\boldsymbol{w}^*,\boldsymbol{c}^*)$ is a feasible solution to problem (7). Thus, the solution $\widehat{\boldsymbol{w}}$ also satisfies the second constraint in problem (4) when we solve this problem with $\boldsymbol{c}=\widehat{\boldsymbol{c}}$, so the claim follows. Since $\widehat{\boldsymbol{w}}$ is a feasible but not necessarily an optimal solution to problem (4) when we solve this problem with $\boldsymbol{c}=\widehat{\boldsymbol{c}}$ and the optimal objective value of the latter problem is $f(\widehat{\boldsymbol{c}})$, we get $f(\widehat{\boldsymbol{c}})\ge\sum_{j\in\mathcal{M}}\sum_{S\subseteq\mathcal{N}}\sum_{i\in\mathcal{N}}r_i\,\phi_{ij}(S)\,\widehat{w}_j(S)=(1-\gamma\frac{n}{K})\sum_{j\in\mathcal{M}}\sum_{S\subseteq\mathcal{N}}\sum_{i\in\mathcal{N}}r_i\,\phi_{ij}(S)\,w_j^*(S)$, where the equality uses the definition of $\widehat{w}_j(S)$. Recalling that app and relax are, respectively, the optimal objective values of problems (3) and (7), by the discussion right before the theorem, we have $\mathsf{relax}\ge\mathsf{app}$, so noting that $(\boldsymbol{w}^*,\boldsymbol{c}^*)$ is an optimal solution to problem (7), we get $\sum_{j\in\mathcal{M}}\sum_{S\subseteq\mathcal{N}}\sum_{i\in\mathcal{N}}r_i\,\phi_{ij}(S)\,w_j^*(S)=\mathsf{relax}\ge\mathsf{app}$. In this case, the last two chains of inequalities establishes that $f(\widehat{\boldsymbol{c}})\ge(1-\gamma\frac{n}{K})\,\mathsf{app}$. Lastly, we have $\sum_{i\in\mathcal{N}}\widehat{c}_i=n\,\gamma+\sum_{i\in\mathcal{N}}\lfloor(1-\gamma\frac{n}{K})\,c_i^*\rfloor\le n\,\gamma+\sum_{i\in\mathcal{N}}(1-\gamma\frac{n}{K})\,c_i^*\le n\gamma+(1-\gamma\frac{n}{K})\,K=K$, where the last inequality holds because $(\boldsymbol{w}^*,\boldsymbol{c}^*)$ is a feasible solution to problem (7). Since γ is an integer, $\widehat{c}_i$ is an integer as well for all $i\in\mathcal{N}$. Thus, having $f(\widehat{\boldsymbol{c}})\ge(1-\gamma\frac{n}{K})\,\mathsf{app}$ and $\sum_{i\in\mathcal{N}}\widehat{c}_i\le K$ implies that $\widehat{\boldsymbol{c}}$ is a $(1-\gamma\,\frac{n}{K})$-approximate solution to problem (3). ■

Noting that we focus on the case $K\ge n$, $\lfloor(\frac{K}{n})^{2/3}\rfloor$ is an integer in the interval $[1,\frac{K}{n}]$. Thus, setting $\gamma=\lfloor(\frac{K}{n})^{2/3}\rfloor$ in the theorem above, we obtain a $(1-\lfloor(\frac{K}{n})^{2/3}\rfloor\frac{n}{K})$-approximate solution to problem (3). This performance guarantee satisfies $(1-\lfloor(\frac{K}{n})^{2/3}\rfloor\frac{n}{K})\ge(1-(\frac{K}{n})^{2/3}\frac{n}{K})=1-\sqrt[3]{\frac{n}{K}}$. Therefore, we can use the theorem above to obtain a $(1-\sqrt[3]{\frac{n}{K}})$-approximate solution to problem (3). For $x\ge1$, we have $\lfloor x\rfloor\ge\frac{1}{2}\,x$, so the approximate solution $\widehat{\boldsymbol{c}}$ that we obtain by setting $\gamma=\lfloor(\frac{K}{n})^{2/3}\rfloor$ satisfies $\widehat{c}_i\ge\lfloor\left(\frac{K}{n}\right)^{2/3}\rfloor\ge\frac{1}{2}\left(\frac{K}{n}\right)^{2/3}$ for all $i\in\mathcal{N}$. There is an inherent tradeoff in the choice of the parameter γ. The stocking quantity of each product in the approximate solution $\widehat{\boldsymbol{c}}$ is at least γ. If we choose γ large, then we obtain an approximate solution with large stocking quantities for each product. When the stocking quantities are thicker, we will be able to come up with assortment customization policies with better performance guarantees, allowing us to use larger values for β in Step 3 of our approximation framework. On the other hand, if we choose γ smaller, then the performance guarantee of $1-\gamma\frac{n}{K}$ for the approximate solution $\widehat{\boldsymbol{c}}$ gets better, allowing us to use larger values for α in Step 2 of our approximation framework.

7. Assortment Customization Policy

We consider constructing an assortment customization policy that we can use in Step 3 of our approximation framework. In particular, for any vector of stocking quantities $\widehat{\boldsymbol{c}} = (\widehat{c}_i : i \in \mathcal{N})$ for the products, we construct an assortment customization policy such that the total expected revenue of the policy starting with the stocking quantities $\widehat{\boldsymbol{c}}$ is lower bounded by a fraction of the linear programming-based surrogate evaluated at $\widehat{\boldsymbol{c}}$. Throughout our discussion in this section, we fix the stocking quantities for the products at $\widehat{\boldsymbol{c}}$. In our assortment customization policy, we solve the linear program in (4) with $\boldsymbol{c} = \widehat{\boldsymbol{c}}$ once at the beginning of the selling horizon. We use $\widehat{\boldsymbol{w}}$ to denote an optimal solution to problem (4) when we solve this problem with $\boldsymbol{c} = \widehat{\boldsymbol{c}}$. Without loss of generality, we assume that the second constraint in problem (4) is tight at the optimal solution, so $\sum_{S \subseteq \mathcal{N}} \widehat{w}_j(S) = \tau_j$ for all $j \in \mathcal{M}$. In particular, since the empty assortment is one possible assortment, if $\sum_{S \subseteq \mathcal{N}} \widehat{w}_j(S) < \tau_j$, then we can increase the value of the decision variable $\widehat{w}_j(\varnothing)$ until the inequality is satisfied as equality. Noting that $\phi_{ij}(\varnothing) = 0$ for all $i \in \mathcal{N}$, we do not change the value of the objective function or the left side of the first constraint by doing so. Thus, since $\widehat{\boldsymbol{w}}$ satisfies $\sum_{S \subseteq \mathcal{N}} \widehat{w}_j(S) = \tau_j$, we use $\{\widehat{w}_j(S)/\tau_j : S \subseteq \mathcal{N}\}$ to characterize a probability distribution over the set of assortments. At any time period, if a customer of type j arrives into the system, then our assortment customization policy samples an assortment $\widehat{S}$ from the probability distribution characterized by $\{\widehat{w}_j(S)/\tau_j : S \subseteq \mathcal{N}\}$, removes all products that do not have remaining inventories from the assortment $\widehat{S}$ and offers the remaining products. Below is a description of our policy. Recall that we use $\mathcal{N}(\boldsymbol{x}) = \{i \in \mathcal{N} : x_i \geq 1\}$ to denote the set of products with remaining inventories when the current inventories of the products are given by the vector $\boldsymbol{x} = (x_i : i \in \mathcal{N})$.

Assortment Customization Policy:

- **(Initialization)** The input is the vector of initial stocking quantities $\widehat{\boldsymbol{c}}$ for the products. Solve the linear program in (4) once at the beginning of the selling horizon with $\boldsymbol{c} = \widehat{\boldsymbol{c}}$ and let $\widehat{\boldsymbol{w}}$ be the corresponding optimal solution.
- **(Decision)** At time period t, if a customer of type j arrives and the current inventories of the products are given by the vector $\boldsymbol{x}$, then sample an assortment $\widehat{S}$ from the probability distribution characterized by $\{\widehat{w}_j(S)/\tau_j : S \subseteq \mathcal{N}\}$ and offer the assortment $\widehat{S} \cap \mathcal{N}(\boldsymbol{x})$.

In the decision step of the policy, we sample the assortment $\widehat{S}$ that we offer to customers of type j such that $\mathbb{P}\{\widehat{S} = S\} = \widehat{w}_j(S)/\tau_j$. We can use a dynamic program similar to the one in (1) to compute the total expected revenue of the policy. We continue using $\boldsymbol{x} = (x_i : i \in \mathcal{N})$ as the state variable at the beginning of a generic time period, where x_i is the remaining inventory of product i. The assortment offer decision that we make at each time period is fixed by the policy. Let $V_t(\boldsymbol{x})$ be the total expected revenue obtained by our assortment customization policy over time periods $t, \ldots, T$ given that the state of the system at the beginning of time period t is $\boldsymbol{x}$. We can compute the value functions $\{V_t : t \in \mathcal{T}\}$ through the dynamic program

$$V_t(\boldsymbol{x}) = \sum_{j \in \mathcal{M}} \lambda_{jt} \sum_{S \subseteq \mathcal{N}} \frac{\widehat{w}_j(S)}{\tau_j} \left\{ \sum_{i \in \mathcal{N}} \phi_{ij}(S \cap \mathcal{N}(\boldsymbol{x})) \Big[r_i + V_{t+1}(\boldsymbol{x} - \boldsymbol{e}_i) - V_{t+1}(\boldsymbol{x}) \Big] \right\} + V_{t+1}(\boldsymbol{x}), \tag{8}$$

with the boundary condition that $V_{T+1} = 0$. The dynamic program above follows from an argument similar to the one in (1), but on the right side of (8), if a customer of type j arrives, then we sample assortment S with probability $\frac{\widehat{w}_j(S)}{\tau_j}$, in which case, we offer the assortment $S \cap \mathcal{N}(\boldsymbol{x})$ to the customer. If we offer the assortment $S \cap \mathcal{N}(\boldsymbol{x})$, then a customer of type j chooses product i with probability $\phi_{ij}(S \cap \mathcal{N}(\boldsymbol{x}))$. Note that the decisions of our policy depends on the initial stocking quantities $\widehat{\boldsymbol{c}}$ that we fixed at the beginning of this section, because $\widehat{\boldsymbol{w}}$ is the optimal solution to problem (4) when we solve this problem with $\boldsymbol{c} = \widehat{\boldsymbol{c}}$. The total expected revenue of our assortment customization policy with the initial stocking quantities $\widehat{\boldsymbol{c}}$ is $V_1(\widehat{\boldsymbol{c}})$. In the next theorem, letting $\widehat{c}_{\min} = \min_{i \in \mathcal{N}} \{\widehat{c}_i : \widehat{c}_i \geq 1\}$ to denote the smallest non-zero stocking quantity for a product, we show that we can lower bound the total expected revenue of our assortment customization policy with $\max\{\frac{1}{2}, 1 - \frac{1}{\sqrt{\widehat{c}_{\min}}}\} f(\widehat{\boldsymbol{c}})$, which implies that we can use our assortment customization policy in Step 3 of our approximation framework with $\beta = \max\{\frac{1}{2}, 1 - \frac{1}{\sqrt{\widehat{c}_{\min}}}\}$.

Theorem 7.1 (Policy Performance) *The total expected revenue obtained by the assortment customization policy with the initial stocking quantities $\widehat{\boldsymbol{c}}$ satisfies $V_1(\widehat{\boldsymbol{c}}) \geq \max\{\frac{1}{2}, 1 - \frac{1}{\sqrt{\widehat{c}_{\min}}}\} f(\widehat{\boldsymbol{c}})$.*

We give the proof in Appendix F. In the proof, we consider an inventory-agnostic assortment customization policy that can offer a product with no remaining inventory, but if the customer chooses such a product, then she leaves without a purchase. We show that the total expected revenue of the inventory-agnostic policy is a lower bound on that of our assortment customization policy, so lower bounding the former is enough to lower bound the latter. The inventory-agnostic policy does not pay attention to the remaining inventories, so we can express the total expected demand for a product under this policy as a sum of independent Bernoullis. To lower bound the total expected revenue of the inventory-agnostic policy, we derive an inequality in Appendix G to show that if Z is a sum of independent Bernoullis and $c \geq \mathbb{E}\{Z\}$, then $\mathbb{E}\{[Z-c]^+\} \leq \min\{\frac{1}{2}, \frac{1}{\sqrt{c}}\} \mathbb{E}\{Z\}$. One related inequality shows that if $c \geq \mathbb{E}\{Z\}$, then $\mathbb{E}\{[Z-c]^+\} \leq \frac{1}{2}\sqrt{\mathbb{E}\{Z\}}$; see Lemma 1 in Gallego and Moon (1993). When c is large, this inequality is significantly looser than ours. Our inequality ends up being critical to give a strong performance guarantee for our policy.

There are two important features of our assortment customization policy. First, our policy is extremely simple. It picks an assortment directly sampled according to an optimal solution to a linear program and offers the sampled assortment after filtering out the products without remaining inventories. Second, our policy obtains a unified performance guarantee of $\max\{\frac{1}{2}, 1-\frac{1}{\sqrt{\widehat{c}_{\min}}}\}$. In contrast, Ma et al. (2021), for example, propose two different policies, each with performance guarantees of $\frac{1}{2}$ and $1-\sqrt{\frac{\log \widehat{c}_{\min}}{\widehat{c}_{\min}}}$, none of which dominates our performance guarantee. Similarly Feng et al. (2020) propose two different policies, each with performance guarantees of $\frac{1}{2}$ and $1-\frac{1}{\sqrt{\widehat{c}_{\min}}}$. Furthermore, as discussed in the introduction, these policies require either solving small-dimensional dynamic programs or non-trivial distortions of the assortment after sampling it according to an optimal solution to a linear program. Using an extremely simple and unified policy, we match or outperform the best performance guarantees in the literature.

8. Computational Experiments

We give computational experiments on synthetically generated datasets. We describe our experimental setup, followed by the benchmark strategies and computational results.

8.1 Experimental Setup

In all of our test problems, the number of products is $n = 100$ and the number of customer types is $m = 50$. There are T time periods in the selling horizon. We vary T. We sample the revenue r_i of each product i from the uniform distribution over $[0, 10]$. We reindex the products such that $r_1 \geq r_2 \geq \ldots \geq r_n$, so the products with smaller indices have larger revenues. The choices of customers of different types are governed by the multinomial logit model with different parameters. To introduce heterogeneity into the customer types, letting L_j be the size of the consideration set for customer type j, we sample L_j uniformly over $\{10, \ldots, 40\}$. Using $\mathcal{C}_j \subseteq \mathcal{N}$ to denote the consideration set of customer type j, we sample $\mathcal{C}_j$ uniformly over all subsets of $\mathcal{N}$ with size L_j. Customers of type j are only interested in the products in the consideration set $\mathcal{C}_j$.

In the multinomial logit model that governs the choice process of customers of type j, using v_{ij} to denote the preference weight that a customer of type j attaches to product i, if $i \in \mathcal{C}_j$ so that customers of type j are interested in purchasing product i, then we sample v_{ij} from the uniform distribution over $[1, 10]$. If $i \notin \mathcal{C}_j$, then we set $v_{ij} = 0$. After we generate all of the preference weights, for half of the customer types, we reorder their preference weights for the products in their consideration sets so that the preference weights follow the reverse order of the product revenues. In this way, these customer types associate smaller preference weights with more expensive products. We set the preference weight of the no-purchase option for customer type j as $v_{0j} = \frac{P_0}{1-P_0} \sum_{i \in \mathcal{C}_j} v_{ij}$. In this case, if we offer all products, then a customer leaves without a purchase with probability

$v_{0j}/(v_{0j}+\sum_{i\in\mathcal{C}_j} v_{ij}) = \frac{P_0}{1-P_0}/(\frac{P_0}{1-P_0}+1) = P_0$. Thus, the parameter P_0 controls the likelihood of the customers to leave without a purchase. We vary this parameter.

We generate the arrival probabilities of different customer types in such a way that customer types with smaller consideration sets tend to arrive later. We view the customer types with smaller consideration sets as the picky ones. When the picky customer types tend to arrive later, it becomes important to reserve the inventory for them. In particular, the probability that we have a customer arrival of type j at time period t is proportional to $\exp(-\gamma\, L_j\,(t-T/2))$ for some $\gamma>0$. Thus, the arrival probability for customer type j at time period t is $\lambda_{jt} = \frac{\exp(-\gamma\, L_j\,(t-T/2))}{\sum_{k\in\mathcal{M}}\exp(-\gamma\, L_k\,(t-T/2))}$. Irrespective of our choice of γ, we have $\sum_{j\in\mathcal{M}}\lambda_{jt}=1$ for all $t\in\mathcal{T}$. Using $\Lambda_{jt}(\gamma)$ to denote the last arrival probability as a function of γ, the market share of customer type j is $\sum_{t\in\mathcal{T}}\Lambda_{jt}(\gamma)/T$. We choose γ such that the customer type with the smallest market share still has a market share of $\theta_{\min}$. We vary $\theta_{\min}$. Lastly, to generate the storage capacity, we compute the myopic assortment that maximizes the expected revenue from a customer of type j as $\widetilde{S}_j = \arg\max_{S\subseteq\mathcal{N}}\sum_{i\in\mathcal{N}} r_i\,\phi_{ij}(S)$. If we always offer the myopic assortments, then the total expected demand for all products over the selling horizon is $\mathsf{Demand} = \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{M}}\lambda_{jt}\,\phi_{ij}(\widetilde{S}_j)$. We set the storage capacity as $K=\lceil \eta\,\mathsf{Demand}\rceil$, where the parameter η controls the tightness of the capacity. We also vary this parameter. The discussion so far describes our approach to generate a test problem.

Varying $T\in\{4000,8000,16000\}$, $P_0\in\{0.1,0.3\}$, $\theta_{\min}\in\{\frac{1}{200},\frac{1}{100}\}$ and $\eta\in\{\frac{1}{4},\frac{1}{2}\}$. we get 24 parameter configurations. For each parameter configuration, we generate 10 test problems.

8.2 Benchmark Policies

We use four benchmarks based on our approximation framework and two heuristics based on dynamic programming decomposition and newsvendor approximation.

Greedy Stocking and Randomized Customization (GRa): For stocking decisions, we replace the objective function of (3) with the approximate surrogate $f_{\mathsf{app}}(\boldsymbol{c})$ in (6) and use the greedy algorithm to get a $(1-\frac{1}{e})$-approximate solution to this problem. Thus, by the discussion in Section 5, we execute Step 2 in our approximation framework with $\alpha=\frac{1}{2}(1-\frac{1}{e})$. For assortment customization decisions, we use the policy in Section 7, so by Theorem 7.1, we have $\beta=\frac{1}{2}$ in Step 3.

Greedy Stocking and Rollout Customization (GRo): We make the stocking decisions as in GRa. Considering assortment customization decisions, by Lemma F.1, we can compute the value functions of the inventory-agnostic policy in closed form. The value functions of the inventory-agnostic policy are $\{\widetilde{V}_t : t\in\mathcal{T}\}$. In this case, if we are at time period t with the remaining inventories $\boldsymbol{x}$, then we offer the assortment $\arg\max_{S\subseteq\mathcal{N}(\boldsymbol{x})}\sum_{i\in\mathcal{N}}\phi_{ij}(S)\,[r_i+\widetilde{V}_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i)-\widetilde{V}_{t+1}(\boldsymbol{x})]$

to a customer of type j. This assortment customization policy corresponds to performing rollout on the inventory-agnostic policy; see, Section 6.1.3 in Bertsekas and Tsitsiklis (1996). Therefore, the total expected revenue of this policy starting with the stocking quantities $\widehat{\boldsymbol{c}}$ is at least as large as the total expected revenue of the inventory-agnostic policy, which is given by $\widetilde{V}_1(\widehat{\boldsymbol{c}})$. In the proof of Theorem 7.1, we establish that $\widetilde{V}_1(\widehat{\boldsymbol{c}}) \geq \frac{1}{2} f(\widehat{\boldsymbol{c}})$, which implies that if we use this assortment customization policy, then we can execute Step 3 in our approximation framework with $\beta = \frac{1}{2}$.

Stocking with Multiple Estimates and Randomized Customization (MRa): In (6), we fix the value of the decision variable y_{0j} in problem (5) at $\tau_j/2$. In this benchmark, we try other values. For $\kappa \in (0,1)$, we define the surrogate $f^{\kappa}_{\mathsf{app}}(\boldsymbol{c})$ as the optimal objective value of problem (5) after we fix the value of the decision variable y_{0j} at $\kappa\,\tau_j$ for all $j \in \mathcal{M}$. To make the stocking decisions, for each $\kappa \in \{0.1, 0.2, \ldots, 0.9\}$, we replace the objective function of problem (3) with the surrogate $f^{\kappa}_{\mathsf{app}}(\boldsymbol{c})$ in (6) and use the greedy algorithm to obtain a $(1-\frac{1}{e})$-approximate solution $\widehat{\boldsymbol{c}}^{\kappa}$ to this problem. In this case, our stocking decisions are given by $\widehat{\boldsymbol{c}} = \arg\max\{f(\boldsymbol{c}) : \boldsymbol{c} = \widehat{\boldsymbol{c}}^{0.1}, \widehat{\boldsymbol{c}}^{0.2} \ldots, \widehat{\boldsymbol{c}}^{0.9}\}$. Since the approximate surrogate corresponds to the case $\kappa = 0.5$, by the discussion in Section 5, the solution $\widehat{\boldsymbol{c}}^{0.5}$ is a $\frac{1}{2}(1-\frac{1}{e})$-approximation to problem (3), so we execute Step 2 in our approximation framework with $\alpha = \frac{1}{2}\,(1-\frac{1}{e})$. We make the assortment customization decisions as in GRa.

Stocking with Multiple Estimates and Rollout Customization (MRo): In this benchmark, we make the stocking decisions as in MRa and assortment customization decisions as in GRo. The benchmarks GRa, GRo, MRa and MRo all yield the same performance guarantee, as they execute Steps 2 and 3 of our approximation framework with $\alpha = \frac{1}{2}\,(1-\frac{1}{e})$ and $\beta = \frac{1}{2}$, but MRa and MRo are more sophisticated in stocking as they try multiple values for the decision variable y_{0j}, whereas GRo and MRo are more sophisticated in assortment customization as they use rollout.

Dynamic Programming Decomposition (Dec): In this benchmark, we adopt the standard dynamic programming decomposition method for network revenue management problems. We construct an approximation to the value function $J_t(\boldsymbol{x})$ in (1) that has a separable form $\sum_{i\in\mathcal{N}} \nu_{it}(x_i)$. For stocking decisions, we solve the problem $\max_{\boldsymbol{c}\in\mathbb{Z}^n_+}\big\{\sum_{i\in\mathcal{N}} \nu_{i1}(c_i) \,:\, \sum_{i\in\mathcal{N}} c_i \leq K\big\}$. For assortment customization decisions, we use the greedy policy with respect to the value function approximation $\sum_{i\in\mathcal{N}} \nu_{it}(x_i)$. We give the details of this benchmark in Appendix H.

Newsvendor Heuristic (NVH): We precompute the assortments to offer to each customer type and solve a newsvendor model to find the stocking quantities under the demand distributions driven by the precomputed assortments. In particular, we compute the myopic assortment that maximizes the expected revenue from a customer of type j as $\widetilde{S}_j = \arg\max_{S\subseteq\mathcal{N}} \sum_{i\in\mathcal{N}} r_i\,\phi_{ij}(S)$. If we always offer the myopic assortment $\widetilde{S}_j$ to a customer of type j, then the demand for product i at time period t is a Bernoulli random variable with parameter $\gamma_{it} = \sum_{j\in\mathcal{M}} \lambda_{jt}\,\phi_{ij}(\widetilde{S}_j)$. Letting $\{X_{it} : t \in \mathcal{T}\}$

be independent Bernoullis with X_{it} having parameter γ_{it}, the total demand for product i over the selling horizon is given by the random variable $X_i = \sum_{t\in\mathcal{T}} X_{it}$. For stocking decisions, we solve the problem $\max_{\boldsymbol{c}\in\mathbb{Z}_+^n}\big\{\sum_{i\in\mathcal{N}} r_i\,\mathbb{E}\{\min\{c_i, X_i\}\} : \sum_{i\in\mathcal{N}} c_i \le K\big\}$. In this problem, if we stock c_i units for product i, then the sales for this product is $\min\{c_i, X_i\}$. Noting that $\mathbb{E}\{\min\{c_i, X_i\}\}$ is concave in c_i, we can solve this problem efficiently. For assortment customization decisions, if we are at time period t with the remaining inventories $\boldsymbol{x}$, then we offer the assortment $\widetilde{S}_j \cap \mathcal{N}(\boldsymbol{x})$ to a customer of type j, where we recall that $\mathcal{N}(\boldsymbol{x})$ is the set of products with positive inventories when the inventory vector is given by $\boldsymbol{x}$. This benchmark has no performance guarantee.

8.3 Computational Results

We give our computational results in Table 1. We normalize the total expected revenues obtained by our benchmarks by using an upper bound on the optimal total expected revenue. Our linear programming-based surrogate $f(\boldsymbol{c})$ satisfies $f(\boldsymbol{c}) \ge J_1(\boldsymbol{c})$ for all $\boldsymbol{c} \in \mathbb{Z}_+^m$, so the optimal objective value of problem (3) is an upper bound on the optimal total expected revenue in (2). To obtain an upper bound efficiently, we solve the linear programming relaxation of problem (3), which is equivalent to problem (7). In the table, the first column gives the parameter configurations for our test problems using the tuple $(T, P_0, \theta_{\min}, \eta)$. We generate 10 test problems for each parameter configuration. The second column gives the average value of the storage capacity K, where the average is computed over the 10 test problems. The next six columns give the average total expected revenues obtained by each benchmark as a percentage of the upper bound, where, once again, the average is computed over the 10 test problems. Letting Rev^k and UB^k, respectively, be the total expected revenue obtained by a benchmark and upper bound for problem instance k, these columns give $\frac{1}{10}\sum_{k=1}^{10} 100 \times \frac{\mathsf{Rev}^k}{\mathsf{UB}^k}$. The last six columns give the average rankings of the total expected revenues of each benchmark. For the first parameter configuration, for example, over the 10 test problems, the average ranking of the total expected revenues of GRA was 4.6. We estimate all total expected revenues by simulating the decisions of the benchmarks for 1000 sample paths.

Our results indicate that MRO is consistently the strongest benchmark, evidenced by its low average rankings. When making the stocking decisions, MRO works with nine different surrogates $\{f^{\kappa}_{\mathsf{app}}(\boldsymbol{c}) : \kappa = 0.1, 0.2, \ldots, 0.9\}$ and tries to pick the best stocking decision found by using each of these surrogates, whereas when making the assortment customization decisions, MRO performs rollout. In contrast, when making the stocking decisions, GRA uses only one surrogate $f_{\mathsf{app}}(\boldsymbol{c})$, whereas when making the assortment customization decisions, GRA directly uses the assortment customization policy in Section 7. Thus, MRO goes one step beyond GRA in both stocking and assortment customization decisions. This effort pays off and MRO improves the performance of

Params. $(T, P_0, \theta_{\min}, \eta)$	K	Total Exp. Rev. GRa	GRo	MRa	MRo	Dec	NVH	Ranking of Total Exp. Rev. GRa	GRo	MRa	MRo	Dec	NVH
$(4000, 0.1, 0.005, 0.25)$	749.4	95.60	99.14	95.84	99.25	97.17	94.44	4.6	1.8	4.3	1.0	3.4	5.5
$(4000, 0.1, 0.005, 0.50)$	1499.5	94.96	98.01	94.99	97.72	93.96	91.95	3.5	1.0	3.6	1.3	4.5	5.7
$(4000, 0.1, 0.010, 0.25)$	748.5	95.55	99.17	95.84	99.30	97.56	94.37	5.0	1.9	4.3	1.1	3.2	5.5
$(4000, 0.1, 0.010, 0.50)$	1497.6	94.92	98.13	94.98	98.00	94.23	91.51	3.3	1.0	3.2	1.1	4.6	6.0
$(4000, 0.3, 0.005, 0.25)$	566.0	90.51	96.41	93.40	97.20	96.65	93.13	6.0	2.5	4.4	1.4	2.1	4.6
$(4000, 0.3, 0.005, 0.50)$	1132.1	90.22	94.99	92.61	95.81	92.74	90.49	5.2	2.6	3.6	1.4	3.2	5.0
$(4000, 0.3, 0.010, 0.25)$	564.5	90.20	96.28	93.25	97.30	97.06	92.77	6.0	2.7	4.4	1.4	1.9	4.6
$(4000, 0.3, 0.010, 0.50)$	1129.5	89.95	94.97	92.52	95.82	90.96	90.34	5.2	2.3	3.4	1.1	4.3	4.7
$(8000, 0.1, 0.005, 0.25)$	1499.5	96.64	99.17	96.89	99.37	97.24	94.81	4.5	1.8	3.8	1.0	3.6	5.9
$(8000, 0.1, 0.005, 0.50)$	2999.6	96.01	98.35	96.08	98.18	93.95	92.37	3.4	1.0	3.3	1.3	5.0	5.6
$(8000, 0.1, 0.010, 0.25)$	1497.6	96.64	99.18	96.87	99.40	97.71	94.76	4.7	2.0	4.1	1.0	3.2	6.0
$(8000, 0.1, 0.010, 0.50)$	2995.7	95.96	98.42	96.01	98.32	94.42	91.97	3.1	1.0	3.0	1.1	5.0	6.0
$(8000, 0.3, 0.005, 0.25)$	1132.1	92.28	96.50	95.09	98.11	97.01	93.63	6.0	2.7	3.9	1.2	2.3	4.9
$(8000, 0.3, 0.005, 0.50)$	2264.7	91.81	95.44	94.24	96.84	92.93	91.06	4.8	2.7	3.5	1.3	3.3	5.4
$(8000, 0.3, 0.010, 0.25)$	1129.5	92.05	96.36	94.94	98.21	97.59	93.35	6.0	3.0	4.0	1.2	1.8	5.0
$(8000, 0.3, 0.010, 0.50)$	2259.5	91.61	95.33	94.16	96.86	90.65	90.93	4.5	2.5	3.3	1.1	4.3	5.3
$(16000, 0.1, 0.005, 0.25)$	2999.6	97.43	99.18	97.66	99.41	97.50	95.09	4.3	1.8	3.5	1.0	4.0	6.0
$(16000, 0.1, 0.005, 0.50)$	5999.8	96.74	98.47	96.80	98.37	93.78	92.66	3.2	1.0	3.1	1.3	5.4	5.6
$(16000, 0.1, 0.010, 0.25)$	2995.7	97.38	99.19	97.63	99.41	97.78	95.04	4.6	2.0	3.7	1.0	3.7	6.0
$(16000, 0.1, 0.010, 0.50)$	5992.1	96.73	98.47	96.76	98.41	94.63	92.28	3.1	1.0	3.0	1.1	5.0	6.0
$(16000, 0.3, 0.005, 0.25)$	2264.7	93.56	96.52	96.26	98.65	97.17	94.01	5.8	2.9	3.4	1.1	2.6	5.2
$(16000, 0.3, 0.005, 0.50)$	4530.0	92.93	95.55	95.35	97.43	92.88	91.49	4.6	2.9	3.3	1.3	3.3	5.6
$(16000, 0.3, 0.010, 0.25)$	2259.5	93.33	96.39	96.14	98.70	97.90	93.76	5.8	3.3	3.7	1.1	1.9	5.2
$(16000, 0.3, 0.010, 0.50)$	4519.5	92.75	95.42	95.28	97.46	90.17	91.32	4.4	2.7	3.1	1.1	4.3	5.4
Avg.		93.99	97.29	95.40	98.06	95.15	92.81	4.7	2.1	3.6	1.2	3.6	5.4

Table 1 Total expected revenues obtained by the benchmarks for the synthetic datasets.

GRa by 4.15% on average. In addition to being the strongest benchmark, on average, MRo obtains total expected revenues within 1.94% of the upper bound on the optimal total expected revenue. The second strongest benchmark is GRo, which uses only one surrogate $f_{\mathsf{app}}(\boldsymbol{c})$ when making the stocking decisions, but performs rollout for the assortment customization decisions.

The total expected revenues of Dec lag behind those of MRo and GRo. Note that Dec attempts to coordinate the stocking and assortment customization decisions, as it constructs value function approximations for the dynamic program that characterizes the optimal assortment customization policy and uses these value function approximations to make the stocking decisions, but Dec considers one product at a time when constructing the value function approximations. The average gap between the total expected revenues of MRo and Dec is 2.97%. The total expected revenues of NVH are significantly smaller than those of all other benchmarks. On average, MRo improves the total expected revenues of NVH by 5.35%. Thus, there is significant value in carefully coordinating the stocking and assortment customization decisions through our approximation framework.

To understand the workings of our benchmarks, we compare the average number of products in the assortments offered by MRo, Dec and NVH as a function of time. We focus on MRo, Dec and NVH because MRo is the strongest benchmark from our approximation framework, whereas Dec and NVH are heuristics motivated by other work in the revenue management literature. Overall, NVH does not try to protect inventory for the customers arriving later and offers larger assortments earlier in the selling horizon. As the inventory is depleted, NVH switches to smaller assortments later. Somewhat reverse trends holds for Dec. This benchmark tends to offer smaller assortments earlier in the selling horizon, but keeps the assortments on the larger side later. In contrast, MRo sits between the two benchmarks. These observations consistently applied to our test problems, but they are based on numerical results and it is difficult to prove technical results to verify that these observations would hold in full generality. In Appendix I, we give detailed computational results that compare the assortments offered by the different benchmarks.

We implemented two variants of each of GRa, GRo, MRa and MRo. In the first variant, we solve problem (3) with the linear programming-based surrogate $f(\boldsymbol{c})$ directly as an integer program. Over all our test problems, the average running times of both GRa and GRo are less than two seconds, whereas the average running times of both MRa and MRo are less than 20 seconds. All running times include the time to compute the stocking quantities, as well as the assortment customization policy. Noting (5), the integer programs have $n+m+nm+1$ constraints and $n+m+nm$ decision variables, n of which take integer values. Relatively compact size of the integer programs may be helping us solve them quickly, but solving an integer program does not yield a polynomial time algorithm. In the second variant, we use the greedy algorithm to get $(1-\frac{1}{e})$-approximate solution to (3) with the approximate surrogate $f_{\mathsf{app}}(\boldsymbol{c})$. Average running times of both GRa and GRo are about 83 minutes. Average running times of both MRa and MRo are about 761 minutes. Greedy algorithm solves $O(Kn)$ linear programs and the value of Kn is on the order of 5×10^5 for our test problems. Average running times for Dec and NVH are, respectively, 488 and 40 minutes.

Additional Sensitivity Analyses and Computational Results: To generate our datasets, we made some choices about the intervals over which we sample the preference weights of the products, the arrival order of the customers with different sizes for their consideration sets and the fraction of customer types for which we reorder the preference weights for the products. It is difficult to argue that one way of generating our datasets is better than the other. In Appendix J, we carry out sensitivity analyses to check the robustness of our benchmarks to changes in our choices. The benchmarks GRa, GRo, MRa and MRo make the stocking decisions by using the greedy algorithm to obtain an approximate solution to problem (3). In Section 6, we describe another approach to make the stocking decisions by using a continuous relaxation of problem (3). In Appendix K, we test the performance of this approach. In Section 2, we explain that we can use our approximation framework to construct a heuristic when we consider the total cost of the stocked units, so that we maximize the total expected profit. In Appendix L, we test such a heuristic. Our model assumes that we can identify the type of each customer. In Appendix M, we give computational experiments when we cannot identify the types of some customers. In this way, we quantify the benefit from customizing the assortment offered to each customer based on her type. Lastly, we generated the test problems that we used in this section synthetically. In Appendix N, we generate our test problems by using a Nielsen dataset on supermarket purchases; see Nielsen (2021). Our findings qualitatively remain the same. The benchmarks that are based on our approximation framework, especially the ones that use rollout, provide significant improvements over Dec and NVH.

9. Conclusions

Motivated by online retail settings for selling fresh groceries and making same-day delivery promises, both of which requiring operating out of an urban warehouse, we studied a joint inventory stocking and assortment customization problem. We gave a $\frac{1}{4}(1-\frac{1}{e})$-approximate solution under the multinomial logit model, whereas we gave a $(1-(\sqrt{2}+1)\sqrt[3]{\frac{n}{K}})$-approximate solution under a general choice model. To our knowledge, these results provide the first approximation guarantees for the joint stocking and assortment customization setting.

There are several directions to pursue. First, the performance guarantee that we gave under a general choice model is $1-(\sqrt{2}+1)\sqrt[3]{\frac{n}{K}}$, which depends on n and K. One can study constant-factor approximation guarantees under a general choice model. Second, under the multinomial logit model, we gave a constant-factor guarantee of $\frac{1}{4}(1-\frac{1}{e})$. One can study constant-factor performance guarantees under other structured choice models, such as the generalized attraction or Markov chain choice model. Third, using results on maximizing submodular functions under a knapsack constraint, letting σ_i be the space or capital consumption of one unit of product i, our approximation framework easily extends to a knapsack constraint of the form $\sum_{i\in\mathcal{N}}\sigma_i\, c_i \leq K$; see Sviridenko (2004). One can study other constraints, beside knapsack constraints. Fourth, we assumed that we can identify the type of a customer precisely before picking an assortment to offer. In Appendix M, we also gave computational experiments when we either identify the type of a customer precisely or not identify the type of a customer at all. In the latter case, we know that we are not able to identify the type of the customer and do not attempt to customize the assortment based on the type. As another possibility, we may misidentify the type of a customer and customize the offered assortment based on the misidentified type. When the type of a customer is dictated by attributes such as zip code, age and average spending on the platform, which are available in the login information, it may be reasonable to hope that we can identify the type of a customer precisely. In contrast, when the type of a customer is dictated by attributes such as brand loyalty and budget, which are not available in the login information, we may misidentify the type of a customer. In Appendix O, we give a brief computational study to check the performance of our approximation framework when we misidentify the customer types. In this computational study, we compute an assortment customization policy under the assumption that we can identify the types of the customers precisely, but check the performance of this policy when the customer types are misidentified. Our model does not explicitly capture the possibility of misidentified customer types. It would be interesting to develop a model that uses a robust or adversarial approach to explicitly model misidentified customer types, where the adversary picks the types of the customers and we offer assortments that protect against the worst-case customer types that could have been picked. Naturally, we expect a wholly different set of tools and results in the adversarial and robust approaches. Fourth, throughout the paper, we use the linear programming-based surrogate. One can study other surrogates to obtain potentially stronger performance guarantees.

Acknowledgements: We thank the area editor, associate editor and two anonymous referees whose comments improved both the technical results and exposition in the paper.

References

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

Aouad, A., R. Levi, D. Segev. 2018. Greedy-like algorithms for dynamic assortment planning under multinomial logit preferences. *Operations Research* **66**(5) 1321–1345.

Aouad, A., R. Levi, D. Segev. 2019. Approximation algorithms for dynamic assortment optimization models. *Mathematics of Operations Research* **44**(2) 487–511.

Baek, J., W. Ma. 2022. Technical note – Bifurcating constraints to improve approximation ratios for network revenue management with reusable resources. *Operations Research* **70**(4) 2226–2236.

Berbeglia, G., A. Garassino, G. Vulcano. 2022. A comparative empirical study of discrete choice models in retail operations. *Management Science* **68**(6) 4005–4023.

Bertsekas, D. P., J. N. Tsitsiklis. 1996. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA.

Bian, A. A., B. Mirzasoleiman, J. Buhmann, Andreas A. Krause. 2017. Guaranteed Non-Convex Optimization: Submodular Maximization over Continuous Domains. A. Singh, Jerry J. Zhu, eds., *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, *Proceedings of Machine Learning Research*, vol. 54. PMLR, 111–120.

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

Cao, Y., A. J. Kleywegt, H. Wang. 2022. Network revenue management under a spiked multinomial logit choice model. *Operations Research* **70**(4) 2237–2253.

Cao, Y., P. Rusmevichientong, H. Topaloglu. 2023. Revenue management under a mixture of independent demand and multinomial logit models. *Operations Research* **71**(2) 603–625.

Chen, X., J. Feldman, S. H. Jung, P. Kouvelis. 2022. Approximation schemes for the joint inventory selection and online resource allocation problem. *Production and Operations Management* **31**(8) 3143–3159.

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

Desir, A., V. Goyal, J. Zhang. 2022. Technical note – Capacitated assortment optimization: Hardness and approximation. *Operations Research* **70**(2) 893–904.

El Housni, O., O. Mouchtaki, G. Gallego, V. Goyal, S. Humair, S. Kim, A. Sadighian, J. Wu. 2021. Joint assortment and inventory planning for heavy tailed demand. Tech. rep., Cornell Tech, New York, NY.

Feige, U. 1998. A threshold of ln n for approximating set cover. *Journal of the Association for Computing Machinery* **45**(4) 634–652.

Feldman, J. B., H. Topaloglu. 2017. Revenue management under the Markov chain choice model. *Operations Research* **65**(5) 1322–1342.

Feng, Y., R. Niazadeh, A. Saberi. 2020. Near-optimal Bayesian online assortment of reusable resources. Tech. rep., University of Chicago, Chicago, IL.

Feng, Y., R. Niazadeh, A. Saberi. 2023. Two-stage stochastic matching and pricing with applications to ride hailing. *Operations Research* (forthcoming).

Gallego, G., G. Iyengar, R. Phillips, A. Dubey. 2004. Managing flexible products on a network. CORC Technical Report TR-2004-01.

Gallego, G., I. K. Moon. 1993. The distribution-free newsboy problem - Review and extensions. *Journal of the Operational Research Society* **44**(8) 825–834.

Gallego, G., R. Ratliff, S. Shebalov. 2015. A general attraction model and sales-based linear programming formulation for network revenue management under customer choice. *Operations Research* **63**(1) 212–232.

Gallego, G., G. van Ryzin. 1994. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management Science* **40**(8) 999–1020.

Gaur, V., D. Honhon. 2006. Assortment planning and inventory decisions under a locational choice model. *Management Science* **52**(10) 1528–1543.

Golrezaei, N., H. Nazerzadeh, P. Rusmevichientong. 2014. Real-time optimization of personalized assortments. *Management Science* **60**(6) 1532–1551.

Goyal, V., R. Levi, Segev D. 2016. Near-optimal algorithms for the assortment planning problem under dynamic substitution and stochastic demand. *Operations Research* **64**(1) 219–235.

Honhon, D., V. Gaur, S. Seshadri. 2010. Assortment planning and inventory decisions under stock-out based substitution. *Operations Research* **58**(5) 1364–1379.

Jasin, S., S. Kumar. 2012. A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. *Mathematics of Operations Research* **37**(2) 313–345.

Liang, A. J., S. Jasin, J. Uichanco. 2021. Assortment and inventory planning under dynamic substitution with MNL model: An LP approach and an asymptotically optimal policy. Tech. rep., University of Michigan, Ann Arbor, MI.

Liu, Q., G. J. van Ryzin. 2008. On the choice-based linear programming model for network revenue management. *Manufacturing & Service Operations Management* **10**(2) 288–310.

Ma, W., D. Simchi-Levi, J. Zhao. 2021. Dynamic pricing (and assortment) under a static calendar. *Management Science* **67**(4) 2292–2313.

Ma, Y., P. Rusmevichientong, M. Sumida, H. Topaloglu. 2020. An approximation algorithm for network revenue management under nonstationary arrivals. *Operations Research* **68**(3) 834–855.

McFadden, D. 1974. Conditional logit analysis of qualitative choice behavior. P. Zarembka, ed., Frontiers in Economics. Academic Press, New York, NY, 105–142.

Nemhauser, G., L. Wolsey, M. Fisher. 1978. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming* **14** 265–294.

Nielsen. 2021. Nielsen and NielsenIQ marketing data. URL `https://www.chicagobooth.edu/research/kilts/datasets/nielsenIQ-nielsen`. Last access date: November 20, 2022.

Pelekis, C. 2016. Lower bounds on binomial and Poisson tails: An approach via tail conditional expectations. URL `https://arxiv.org/abs/1609.06651`. Last access date: November 20, 2022.

Pollard, D. 2021. Empirical processes. URL `http://www.stat.yale.edu/~pollard/Books/Mini/BinFriends.pdf`. Last access date: November 20, 2022.

Rusmevichientong, P., M. Sumida, H. Topaloglu. 2020. Dynamic assortment optimization for reusable products with random usage durations. *Management Science* **66**(7) 2820–2844.

Soma, T., Y. Yoshida. 2018. Maximizing monotone submodular functions over the integer lattice. *Mathematical Programming* **172** 539–563.

Sviridenko, M. 2004. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters* **32**(1) 41–43.

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

Talluri, K. T., G. J. van Ryzin. 2005. *The Theory and Practice of Revenue Management*. Kluwer Academic Publishers, Boston, MA.

Topaloglu, H. 2013. Joint stocking and product offer decisions under the multinomial logit model. *Production and Operations Management* **22**(5) 1182–1199.

Train, K. 2003. *Discrete Choice Methods with Simulation*. Cambridge University Press, Cambridge, UK.

van Ryzin, G., S. Mahajan. 1999. On the relationship between inventory costs and variety benefits in retail assortments. *Management Science* **45**(11) 1496–1509.

Zhang, D., D. Adelman. 2009. An approximate dynamic programming approach to network revenue management with customer choice. *Transportation Science* **42**(3) 381–394.

Zhang, H., P. Rusmevichientong, H. Topaloglu. 2020. Assortment optimization under the paired combinatorial logit model. *Operations Research* **68**(3) 741–761.

Zhang, Z., H.-S. Ahn, L. Baardman. 2022. Ordering and ranking products for an online retailer. Tech. rep., University of Michigan, Ann Arbor, MI.

Electronic Supplement: Coordinated Inventory Stocking and Assortment Customization

Yicheng Bai, Omar El Housni, Paat Rusmevichientong, Huseyin Topaloglu

January 1, 2025

Appendix A: Tight Upper Bound on the Value of Customization

We consider a variant of our joint stocking and assortment customization problem, where we offer the same assortment to the customer arriving at a particular time period, irrespective of the type of the customer. In other words, we do not customize the assortment offered to each customer in this variant. Naturally, the total expected revenue that we obtain by customizing the assortment offered to the customer arriving at a particular time period is larger than the total expected revenue that we obtain by not customizing. We show that the total expected revenue obtained by customizing the assortments improves the total expected revenue obtained by not customizing by at most a factor of m, where m is the number of customer types. Furthermore, we show that this bound is tight. In particular, we show that there exists a problem instance such that the total expected revenue obtained by customizing the assortments improves the total expected revenue obtained by not customizing by at least a factor of $(1-\epsilon)\,m$ for any $\epsilon > 0$. In this way, we give a tight characterization of the value of customizing the assortment offered to the customer arriving at each time period. We start by showing that the total expected revenue obtained by customizing the assortments improves the total expected revenue obtained by not customizing by at most a factor of m. While this result is intuitive, its proof is not straightforward and uses the properties of the choice model. We continue using the notation that we introduced in Section 2. When we do not customize the assortment offered to the customer arriving at a particular time period, let $V_t(\boldsymbol{x})$ be the maximum total expected revenue over time periods $t,\ldots,T$ given that the state of the system at the beginning of time period t is $\boldsymbol{x}$. When we do not customize, we can compute the value functions $\{V_t : t \in \mathcal{T}\}$ through the dynamic program

$$
\begin{aligned}
V_t(\boldsymbol{x}) &= \max_{S\subseteq\mathcal{N}(\boldsymbol{x})}\left\{\sum_{j\in\mathcal{M}}\lambda_{jt}\left\{\sum_{i\in\mathcal{N}}\phi_{ij}(S)\Big[r_i+V_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i)\Big]+\left(1-\sum_{i\in\mathcal{N}}\phi_{ij}(S)\right)V_{t+1}(\boldsymbol{x})\right\}\right\}+\left(1-\sum_{j\in\mathcal{M}}\lambda_{jt}\right)V_{t+1}(\boldsymbol{x})\\
&= \max_{S\subseteq\mathcal{N}(\boldsymbol{x})}\left\{\sum_{i\in\mathcal{N}}\left(\sum_{j\in\mathcal{M}}\lambda_{jt}\,\phi_{ij}(S)\right)\Big[r_i+V_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i)-V_{t+1}(\boldsymbol{x})\Big]\right\}+V_{t+1}(\boldsymbol{x}), \qquad (9)
\end{aligned}
$$

with the boundary condition that $V_{T+1}=0$. The second equality in the chain of equalities above follows by arranging the terms.

If we do not customize the assortment offered to the customers arriving at each time period, then the optimal total expected revenue is $\nu^* = \max_{\boldsymbol{c}\in\mathbb{Z}_+^n}\big\{V_1(\boldsymbol{c}) : \sum_{i\in\mathcal{N}} c_i \le K\big\}$. Noting that opt is the optimal objective value of problem (2), we will show that $\mathsf{opt} \leq m\,\nu^*$, so customizing the assortment improves the total expected revenue obtained by not customizing by at most a factor of m. Following this result, we will show that there exists a problem instance such that $\mathsf{opt} \geq (1-\epsilon)\,m\,\nu^*$ for any $\epsilon > 0$, so there exists a problem instance for which customizing the assortment can improve the total expected revenue obtained by not customizing by a factor arbitrarily close to m. We proceed to showing that $\mathsf{opt} \leq m\,\nu^*$. To show this result, we use a variant of the dynamic program in (1) formulated under the assumption that we obtain revenue only from customers of type j. We refer to this dynamic program as the auxiliary dynamic program for customer type j. The auxiliary dynamic program for customer type j is given by

$$\begin{aligned} Q_{jt}(\boldsymbol{x}) &= \lambda_{jt} \max_{S \subseteq \mathcal{N}(\boldsymbol{x})} \left\{ \sum_{i \in \mathcal{N}} \phi_{ij}(S) \Big[r_i + Q_{j,t+1}(\boldsymbol{x} - \boldsymbol{e}_i) \Big] + \left(1 - \sum_{i \in \mathcal{N}} \phi_{ij}(S) \right) Q_{j,t+1}(\boldsymbol{x}) \right\} + (1 - \lambda_{jt})\, Q_{j,t+1}(\boldsymbol{x}) \\ &= \lambda_{jt} \max_{S \subseteq \mathcal{N}(\boldsymbol{x})} \left\{ \sum_{i \in \mathcal{N}} \phi_{ij}(S) \Big[r_i + Q_{j,t+1}(\boldsymbol{x} - \boldsymbol{e}_i) - Q_{j,t+1}(\boldsymbol{x}) \Big] \right\} + Q_{j,t+1}(\boldsymbol{x}), \end{aligned} \tag{10}$$

with the boundary condition that $Q_{j,T+1} = 0$. We need a sequence of lemmas. In the next lemma, we show that $Q_{jt}(\boldsymbol{x})$ is monotone in $\boldsymbol{x}$.

Lemma A.1 *For any $t \in \mathcal{T}$ and $\boldsymbol{x} \in \mathbb{Z}_+^n$ with $x_i \geq 1$, we have $Q_{jt}(\boldsymbol{x}) \geq Q_{jt}(\boldsymbol{x} - \boldsymbol{e}_i)$.*

Proof: We show the result by using induction over the time periods. We have $Q_{j,T+1} = 0$ at time period $T+1$, so the result holds at time period $T+1$. Assuming that the result holds at time period $t+1$, we show that the result holds at time period t as well. The coefficients of $Q_{j,t+1}(\boldsymbol{x} - \boldsymbol{e}_i)$ and $Q_{j,t+1}(\boldsymbol{x})$ on the right side of the first equality in (10) are non-negative. Thus, if we replace $Q_{j,t+1}(\boldsymbol{x} - \boldsymbol{e}_i)$ and $Q_{j,t+1}(\boldsymbol{x})$ with larger quantities, then the right side of this equality gets larger. The expressions on the right side of the first and second equalities in (10) are equal to each other, so the same observation also applies to the right side of the second equality in (10). Consider $\boldsymbol{x} \in \mathbb{Z}_+^n$ with $x_\ell \geq 1$. By the induction assumption, we have $Q_{j,t+1}(\boldsymbol{x} - \boldsymbol{e}_\ell) \leq Q_{j,t+1}(\boldsymbol{x})$, as well as $Q_{j,t+1}(\boldsymbol{x} - \boldsymbol{e}_\ell - \boldsymbol{e}_i) \leq Q_{j,t+1}(\boldsymbol{x} - \boldsymbol{e}_\ell)$ for all $i \in \mathcal{N}(\boldsymbol{x} - \boldsymbol{e}_\ell)$. By (10), we get

$$\begin{aligned} Q_{jt}(\boldsymbol{x} - \boldsymbol{e}_\ell) &= \lambda_{jt} \max_{S \subseteq \mathcal{N}(\boldsymbol{x} - \boldsymbol{e}_\ell)} \left\{ \sum_{i \in \mathcal{N}} \phi_{ij}(S) \Big[r_i + Q_{j,t+1}(\boldsymbol{x} - \boldsymbol{e}_\ell - \boldsymbol{e}_i) - Q_{j,t+1}(\boldsymbol{x} - \boldsymbol{e}_\ell) \Big] \right\} + Q_{j,t+1}(\boldsymbol{x} - \boldsymbol{e}_\ell) \\ &\leq \lambda_{jt} \max_{S \subseteq \mathcal{N}(\boldsymbol{x})} \left\{ \sum_{i \in \mathcal{N}} \phi_{ij}(S) \Big[r_i + Q_{j,t+1}(\boldsymbol{x} - \boldsymbol{e}_i) - Q_{j,t+1}(\boldsymbol{x}) \Big] \right\} + Q_{j,t+1}(\boldsymbol{x}) \;=\; Q_{jt}(\boldsymbol{x}), \end{aligned}$$

where the inequality follows from the discussion right before the chain of inequalities above and the fact that we have $\mathcal{N}(\boldsymbol{x} - \boldsymbol{e}_\ell) \subseteq \mathcal{N}(\boldsymbol{x})$. ∎

In the next lemma, we relate the value functions $\{Q_{jt} : t \in \mathcal{T}\}$ computed through (10) to the value functions $\{J_t : t \in \mathcal{T}\}$ computed through (1).

Lemma A.2 *For any $t \in \mathcal{T}$ and $\boldsymbol{x} \in \mathbb{Z}_+^n$, we have $\sum_{j\in\mathcal{M}} Q_{jt}(\boldsymbol{x}) \geq J_t(\boldsymbol{x})$.*

Proof: We show the result by using induction over the time periods. We have $Q_{j,T+1} = 0 = J_{T+1}$ at time period $T+1$, so the result holds at time period $T+1$. Assuming that the result holds at time period $t+1$, we show that the result holds at time period t as well. We make two observations. First, considering any $j \in \mathcal{M}$, by Lemma A.1, we have $Q_{k,t+1}(\boldsymbol{x}-\boldsymbol{e}_i) - Q_{k,t+1}(\boldsymbol{x}) \leq 0$ for all $k \in \mathcal{M}\setminus\{j\}$, in which case, we get $Q_{j,t+1}(\boldsymbol{x}-\boldsymbol{e}_i) - Q_{j,t+1}(\boldsymbol{x}) \geq \sum_{k\in\mathcal{M}} Q_{k,t+1}(\boldsymbol{x}-\boldsymbol{e}_i) - \sum_{k\in\mathcal{M}} Q_{k,t+1}(\boldsymbol{x})$ for any $\boldsymbol{x} \in \mathbb{Z}_+^n$ with $x_i \geq 1$. Second, by the induction assumption, $\sum_{k\in\mathcal{M}} Q_{k,t+1}(\boldsymbol{x}) \geq J_{t+1}(\boldsymbol{x})$ and $\sum_{k\in\mathcal{M}} Q_{k,t+1}(\boldsymbol{x}-\boldsymbol{e}_i) \geq J_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i)$ for any $\boldsymbol{x} \in \mathbb{Z}_+^n$ with $x_i \geq 1$. Having $i \in \mathcal{N}(\boldsymbol{x})$ is equivalent to having $x_i \geq 1$, so adding (10) over all $j \in \mathcal{M}$, we get the chain of inequalities

$$\begin{aligned}
\sum_{j\in\mathcal{M}} Q_{jt}(\boldsymbol{x}) &= \sum_{j\in\mathcal{M}} \lambda_{jt} \max_{S\subseteq\mathcal{N}(\boldsymbol{x})} \left\{ \sum_{i\in\mathcal{N}} \phi_{ij}(S) \Big[r_i + Q_{j,t+1}(\boldsymbol{x}-\boldsymbol{e}_i) - Q_{j,t+1}(\boldsymbol{x}) \Big] \right\} + \sum_{j\in\mathcal{M}} Q_{j,t+1}(\boldsymbol{x}) \\
&\overset{(a)}{\geq} \sum_{j\in\mathcal{M}} \lambda_{jt} \max_{S\subseteq\mathcal{N}(\boldsymbol{x})} \left\{ \sum_{i\in\mathcal{N}} \phi_{ij}(S) \Big[r_i + \sum_{k\in\mathcal{M}} Q_{k,t+1}(\boldsymbol{x}-\boldsymbol{e}_i) - \sum_{k\in\mathcal{M}} Q_{k,t+1}(\boldsymbol{x}) \Big] \right\} + \sum_{j\in\mathcal{M}} Q_{j,t+1}(\boldsymbol{x}) \\
&\overset{(b)}{\geq} \sum_{j\in\mathcal{M}} \lambda_{jt} \max_{S\subseteq\mathcal{N}(\boldsymbol{x})} \left\{ \sum_{i\in\mathcal{N}} \phi_{ij}(S) \Big[r_i + J_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i) - J_{t+1}(\boldsymbol{x}) \Big] \right\} + J_{t+1}(\boldsymbol{x}) \;=\; J_t(\boldsymbol{x}),
\end{aligned}$$

where (a) uses the first observation and (b) uses the second observation, as well as the fact the coefficient of $\sum_{k\in\mathcal{M}} Q_{k,t+1}(\boldsymbol{x})$ on the left side of this inequality is $1 - \sum_{j\in\mathcal{M}} \sum_{i\in\mathcal{N}} \lambda_{jt}\, \phi_{ij}(S) \geq 0$. ■

There exists an optimal solution to the maximization problem on the right side of (9) such that if $r_i + V_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i) - V_{t+1}(\boldsymbol{x}) \leq 0$, then product i is not included in the optimal solution. In particular, letting S^* be an optimal solution, using $\Phi_{it}(S) = \sum_{j\in\mathcal{M}} \lambda_{jt}\, \phi_{ij}(S)$ for notational brevity, the contribution of product i to the optimal objective value is $\Phi_{it}(S^*)\,(r_i + V_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i) - V_{t+1}(\boldsymbol{x}))$. If we drop all products in S^* that make a non-positive contribution, then we get the solution $\widehat{S} = \{i \in S^* : r_i + V_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i) - V_{t+1}(\boldsymbol{x}) > 0\}$. By the substitutability property, $\Phi_{it}(\widehat{S}) \geq \Phi_{it}(S^*)$ for all $i \in \widehat{S}$. Thus, we have $\Phi_{it}(\widehat{S})\,(r_i + V_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i) - V_{t+1}(\boldsymbol{x})) \geq \Phi_{it}(S^*)\,(r_i + V_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i) - V_{t+1}(\boldsymbol{x}))$ for all $i \in \widehat{S}$ and $\Phi_{it}(S^*)\,(r_i + V_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i) - V_{t+1}(\boldsymbol{x})) \leq 0$ for all $i \in S^* \setminus \widehat{S}$, so $\widehat{S}$ is also optimal.

In the next lemma, we use the observation in the previous paragraph to relate the value functions $\{Q_{jt} : t \in \mathcal{T}\}$ computed through (10) to the value functions $\{V_t : t \in \mathcal{T}\}$ computed through (9).

Lemma A.3 *For any $t \in \mathcal{T}$, $j \in \mathcal{M}$ and $\boldsymbol{x} \in \mathbb{Z}_+^n$, we have $Q_{jt}(\boldsymbol{x}) \leq V_t(\boldsymbol{x})$.*

Proof: We show the result by using induction over the time periods. We have $Q_{j,T+1} = 0 = V_{T+1}$ at time period $T+1$, so the result holds at time period $T+1$. Assuming that the result holds at time period $t+1$, we show that the result holds at time period t as well. We make three observations. First, by the discussion right before the lemma, if $r_i + V_{t+1}(\boldsymbol{x} - \boldsymbol{e}_i) - V_{t+1}(\boldsymbol{x}) \leq 0$, then we can drop product i from consideration in the maximization problem on the right side of (9). Thus, letting $\overline{\mathcal{N}}_t(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}) \cap \{i \in \mathcal{N} : r_i + V_{t+1}(\boldsymbol{x} - \boldsymbol{e}_i) - V_{t+1}(\boldsymbol{x}) > 0\}$, we can replace $\mathcal{N}(\boldsymbol{x})$ in the maximization problem on the right side of (9) with $\overline{\mathcal{N}}_t(\boldsymbol{x})$ without changing the optimal objective value. Second, we have $\sum_{j \in \mathcal{M}} \lambda_{jt}\, \phi_{ij}(S) \geq \lambda_{kt}\, \phi_{ik}(S)$ for any $k \in \mathcal{M}$, so noting the definition of $\overline{\mathcal{N}}_t(\boldsymbol{x})$, we get $\sum_{j \in \mathcal{M}} \lambda_{jt}\, \phi_{ij}(S)\, (r_i + V_{t+1}(\boldsymbol{x} - \boldsymbol{e}_i) - V_{t+1}(\boldsymbol{x})) \geq \lambda_{kt}\, \phi_{ik}(S)\, (r_i + V_{t+1}(\boldsymbol{x} - \boldsymbol{e}_i) - V_{t+1}(\boldsymbol{x}))$ for all $i \in \overline{\mathcal{N}}_t(\boldsymbol{x})$. Third, if we replace the probability $\sum_{j \in \mathcal{M}} \lambda_{jt}\, \phi_{ij}(S)$ with $\lambda_{jt}\, \phi_{ij}(S)$ in the maximization problem on the right side of (9), then we can follow precisely the same reasoning right before the lemma to argue that we can replace $\mathcal{N}(\boldsymbol{x})$ with $\overline{\mathcal{N}}_t(\boldsymbol{x})$ without changing the optimal objective value of this problem. In this case, for any $k \in \mathcal{M}$, we get the chain of inequalities

$$\begin{aligned}
V_t(\boldsymbol{x}) &\overset{(a)}{=} \max_{S \subseteq \overline{\mathcal{N}}_t(\boldsymbol{x})} \left\{ \sum_{i \in \mathcal{N}} \left(\sum_{j \in \mathcal{M}} \lambda_{jt}\, \phi_{ij}(S) \right) \Big[r_i + V_{t+1}(\boldsymbol{x} - \boldsymbol{e}_i) - V_{t+1}(\boldsymbol{x}) \Big] \right\} + V_{t+1}(\boldsymbol{x}) \\
&\overset{(b)}{\geq} \max_{S \subseteq \overline{\mathcal{N}}_t(\boldsymbol{x})} \left\{ \sum_{i \in \mathcal{N}} \lambda_{kt}\, \phi_{ik}(S) \Big[r_i + V_{t+1}(\boldsymbol{x} - \boldsymbol{e}_i) - V_{t+1}(\boldsymbol{x}) \Big] \right\} + V_{t+1}(\boldsymbol{x}) \\
&\overset{(c)}{=} \max_{S \subseteq \mathcal{N}(\boldsymbol{x})} \left\{ \sum_{i \in \mathcal{N}} \lambda_{kt}\, \phi_{ik}(S) \Big[r_i + V_{t+1}(\boldsymbol{x} - \boldsymbol{e}_i) - V_{t+1}(\boldsymbol{x}) \Big] \right\} + V_{t+1}(\boldsymbol{x}) \\
&\overset{(d)}{\geq} \max_{S \subseteq \mathcal{N}(\boldsymbol{x})} \left\{ \sum_{i \in \mathcal{N}} \lambda_{kt}\, \phi_{ik}(S) \Big[r_i + Q_{k,t+1}(\boldsymbol{x} - \boldsymbol{e}_i) - Q_{k,t+1}(\boldsymbol{x}) \Big] \right\} + Q_{k,t+1}(\boldsymbol{x}) \;=\; Q_{kt}(\boldsymbol{x}),
\end{aligned}$$

where (a), (b) and (c) use the first, second and third observations, (d) is by the induction assumption and the coefficient of $V_{t+1}(\boldsymbol{x})$ on the left side of this inequality is $1 - \sum_{i \in \mathcal{N}} \lambda_{kt}\, \phi_{ik}(S) \geq 0$. ∎

Recall that the optimal total expected revenue is $\nu^* = \max_{\boldsymbol{c} \in \mathbb{Z}_+^n} \big\{ V_1(\boldsymbol{c}) : \sum_{i \in \mathcal{N}} c_i \leq K \big\}$ when we do not customize. In the next proposition, we upper bound the value of customization.

Proposition A.4 *We have* $\mathsf{opt} \leq m\, \nu^*$.

Proof: Let $\boldsymbol{c}^*$ be an optimal solution to problem (2) and $\widehat{j} = \arg\max_{j \in \mathcal{M}} Q_{j1}(\boldsymbol{c}^*)$. By the definition of $\widehat{j}$, we have $\sum_{j \in \mathcal{M}} Q_{j1}(\boldsymbol{c}^*) \leq m\, Q_{\widehat{j}1}(\boldsymbol{c}^*)$. By Lemma A.2, we have $\sum_{j \in \mathcal{M}} Q_{j1}(\boldsymbol{c}^*) \geq J_1(\boldsymbol{c}^*)$. By Lemma A.3, we have $Q_{\widehat{j}1}(\boldsymbol{c}^*) \leq V_1(\boldsymbol{c}^*)$. Putting the preceding three inequalities together, we have $\mathsf{opt} = J_1(\boldsymbol{c}^*) \leq \sum_{j \in \mathcal{M}} Q_{j1}(\boldsymbol{c}^*) \leq m\, Q_{\widehat{j}1}(\boldsymbol{c}^*) \leq m\, V_1(\boldsymbol{c}^*) \leq m\, \nu^*$, where the last inequality holds because c^* is a feasible solution to the problem $\nu^* = \max_{\boldsymbol{c} \in \mathbb{Z}_+^n} \big\{ V_1(\boldsymbol{c}) : \sum_{i \in \mathcal{N}} c_i \leq K \big\}$. ∎

Next, we lower bound the value of customization.

Lower Bounding the Value of Customization:

For any $\epsilon > 0$, we show that we can come up with a problem instance such that $\mathsf{opt} \geq (1 - \epsilon)\, m\, \nu^*$, which establishes that there exists a problem instance, where customizing the assortment can improve the total expected revenue obtained by not customizing by a factor arbitrarily close to m. We consider a sequence of problem instances indexed by the number of customer types m. In each problem instance, there is one time period in the selling horizon, so we drop the time index in the arrival probability λ_{jt} for a customer of type j at time period t. In this case, the dynamic programs in (1) and (9), respectively, take the form $J_1(\boldsymbol{x}) = \sum_{j\in\mathcal{M}} \lambda_j \max_{S\subseteq\mathcal{N}(\boldsymbol{x})} \sum_{i\in\mathcal{N}} \phi_{ij}(S)\, r_i$ and $V_1(\boldsymbol{x}) = \max_{S\subseteq\mathcal{N}(\boldsymbol{x})} \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{M}} \lambda_j\, \phi_{ij}(S)\, r_i$. In each problem instance, the storage capacity is infinite. Because we have one time period in the selling horizon, there is no need to stock more than one unit of each product. Furthermore, because the storage capacity is infinite, we only need to find an assortment of products that maximizes the expected revenue obtained from the customer arriving at the single time period. In this case, problems $\mathsf{opt} = \max_{\boldsymbol{c}\in\mathbb{Z}_+^n} \{J_1(\boldsymbol{c}) : \sum_{i\in\mathcal{N}} c_i \le K\}$ and $\nu^* = \max_{\boldsymbol{c}\in\mathbb{Z}_+^n} \{V_1(\boldsymbol{c}) : \sum_{i\in\mathcal{N}} c_i \le K\}$ are, respectively, given by

$$\mathsf{opt} = \sum_{j\in\mathcal{M}} \lambda_j \max_{S\subseteq\mathcal{N}} \left\{ \sum_{i\in\mathcal{N}} \phi_{ij}(S)\, r_i \right\} \quad \text{and} \quad \nu^* = \max_{S\subseteq\mathcal{N}} \left\{ \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{M}} \lambda_j\, \phi_{ij}(S)\, r_i \right\}. \tag{11}$$

In our problem instance, the number of products is equal to the number of customer types, so $n = m$. We index the sets of products and customer types both by $\mathcal{M} = \{1, \ldots, m\}$. To specify the revenues of the products, arrival probabilities of the customer types and choice probabilities, we adopt a problem instance in Desir et al. (2022), where the authors use this problem instance to characterize the complexity of the assortment optimization problem under a mixture of multinomial logit models. In particular, the revenue of product i is $r_i = m^{3\,(i-1)}$. The probability that a customer of type j arrives into the system is $\lambda_j = \frac{1}{2}\, m^{3\,(1-j)}$. With probability $1 - \frac{1}{2}\sum_{j=1}^m m^{3\,(1-j)}$, there is no customer arrival. Choices of the customers of different types are governed by the multinomial logit model. The preference weight that a customer of type j associates with product i is given by $v_{ij} = m^2$ for $i < j$, $v_{ij} = 1$ for $i = j$ and $v_{ij} = 0$ for $i > j$. If we offer the assortment S to a customer of type j, then she chooses product $i \in S$ with probability $\phi_{ij}(S) = v_{ij}/(1 + \sum_{k\in S} v_{kj})$.

As a function of m, we refer to the problem instance given in the previous two paragraphs as $\mathcal{P}^m$. In the next lemma, we give a simple lower bound on opt for problem instance $\mathcal{P}^m$.

Lemma A.5 *For problem instance $\mathcal{P}^m$, we have $\mathsf{opt} \ge \frac{m}{4}$.*

Proof: We have $\phi_{jj}(\{j\}) = \frac{1}{2}$ and $\phi_{ij}(\{j\}) = 0$ for $i \ne j$. The solution $\{j\}$ is feasible to the first maximization problem in (11), so $\mathsf{opt} \ge \sum_{j\in\mathcal{M}} \lambda_j\, \phi_{jj}(\{j\})\, r_j = \sum_{j\in\mathcal{M}} \frac{1}{2} m^{3(1-j)}\, \frac{1}{2}\, m^{3(j-1)} = \frac{m}{4}$. ■

In the next lemma, we give an upper bound on ν^*. The proof of the upper bound on ν^* in this lemma is more involved than the proof of the lower bound on opt in the previous lemma.

Lemma A.6 *For problem instance $\mathcal{P}^m$, we have $\nu^* \leq \frac{1}{4} + \frac{1}{m}$.*

Proof: Let S^* be an optimal solution to the second maximization problem in (11). We will upper bound $\sum_{i\in\mathcal{N}} \phi_{ij}(S^*)\, r_i$. Letting $N_j^* = \{i \in S^* : i < j\}$, we consider three cases. First, assume that $N_j^* = \varnothing$ and $j \notin S^*$. We have $v_{ij} = 0$ for all $i > j$, so $\phi_{ij}(S^*) = 0$ for all $i > j$. Because $N_j^* = \varnothing$, we have $i \notin S^*$ for all $i < j$. Thus, we have $\phi_{ij}(S^*) = 0$ for all $i < j$. Finally, $\phi_{jj}(S^*) = 0$ since $j \notin S^*$. Thus, we have $\sum_{i\in\mathcal{N}} \phi_{ij}(S^*)\, r_i = 0$. Second, assume that $N_j^* = \varnothing$ and $j \in S^*$. Because $N_j^* = \varnothing$, by the same argument in the first case, we have $\phi_{ij}(S) = 0$ for all $i > j$ or $i < j$. Noting that $j \in S^*$, we have $\phi_{jj}(S^*) = \frac{v_{jj}}{1+v_{jj}} = \frac{1}{2}$. Thus, we have $\sum_{i\in\mathcal{N}} \phi_{ij}(S^*)\, r_i = \frac{1}{2} m^{3\,(j-1)}$. Third, assume that $N_j^* \neq \varnothing$. Noting that $v_{ij} = 0$ for $i > j$, letting $\mathbf{1}(\cdot)$ be the indicator function, we have

$$
\begin{aligned}
\sum_{i\in\mathcal{N}} \phi_{ij}(S^*)\, r_i &= \frac{\sum_{i\in S^*} \mathbf{1}(i \leq j)\, v_{ij}\, r_i}{1 + \sum_{i\in S^*} \mathbf{1}(i \leq j)\, v_{ij}} \leq \frac{v_{jj}\, r_j + \sum_{i\in S^*} \mathbf{1}(i < j)\, v_{ij}\, r_i}{\sum_{i\in S^*} \mathbf{1}(i < j)\, v_{ij}} \\
&= \frac{m^{3\,(j-1)} + m^2 \sum_{i\in S^*} \mathbf{1}(i<j)\, m^{3\,(i-1)}}{m^2 \sum_{i\in S^*} \mathbf{1}(i<j)} \leq \frac{m^{3\,(j-1)} + m^2\, m^{3\,(j-2)} \sum_{i\in S^*} \mathbf{1}(i<j)}{m^2 \sum_{i\in S^*} \mathbf{1}(i<j)} \\
&= \frac{m^{3\,(j-1)} + m^{3j-4}\, |N_j^*|}{m^2\, |N_j^*|} \leq \frac{m^{3\,(j-1)} + m^{3j-4}\, m}{m^2} = 2\, \frac{m^{3\,(j-1)}}{m^2}.
\end{aligned}
$$

The discussion in the previous paragraph gives an upper bound on $\sum_{i\in\mathcal{N}} \phi_{ij}(S^*)\, r_i$ in three cases. Noting that $\nu^* = \sum_{j\in\mathcal{M}} \lambda_j \sum_{i\in\mathcal{N}} \phi_{ij}(S^*)\, r_i$, we will use these upper bounds to ultimately upper bound ν^*. We need one claim. We claim that there can be at most one customer type j that satisfies $N_j^* = \varnothing$ and $j \in S^*$. To get a contradiction, assume that customer types j_1 and j_2 with $j_1 < j_2$ satisfy $N_{j_1}^* = \varnothing$, $j_1 \in S^*$, $N_{j_2}^* = \varnothing$ and $j_2 \in S^*$. Since $N_{j_2}^* = \varnothing$, by the definition of N_j^*, we have $i \geq j_2$ for all $i \in S^*$. However, we have $j_1 \in S^*$ and $j_1 < j_2$, which is a contradiction, so the claim holds. By the claim, we have $\sum_{j\in\mathcal{M}} \mathbf{1}(N_j^* = \varnothing,\ j \in S^*) \leq 1$. Therefore, collecting the three cases in the previous paragraph, using the fact that $\lambda_j = \frac{1}{2} m^{3\,(1-j)}$, we upper bound ν^* as

$$
\begin{aligned}
\nu^* &= \sum_{j\in\mathcal{M}} \lambda_j \sum_{i\in\mathcal{N}} \phi_{ij}(S^*)\, r_i \leq \sum_{j\in\mathcal{M}} \lambda_j\, \mathbf{1}(N_j^* = \varnothing,\ j \in S^*) \frac{1}{2} m^{3\,(j-1)} + \sum_{j\in\mathcal{M}} \lambda_j\, \mathbf{1}(N_j^* \neq \varnothing)\, 2\, \frac{m^{3\,(j-1)}}{m^2} \\
&= \frac{1}{4} \sum_{j\in\mathcal{M}} \mathbf{1}(N_j^* = \varnothing,\ j \in S^*) + \frac{1}{m^2} \sum_{j\in\mathcal{M}} \mathbf{1}(N_j^* \neq \varnothing) \leq \frac{1}{4} + \frac{1}{m}.
\end{aligned}
$$

■

In the next proposition, we show that there are problem instances, where customization can improve the total expected revenue by a factor arbitrarily close to m.

Proposition A.7 *For any $\epsilon > 0$, there exists a problem instance such that $\mathsf{opt} \geq (1-\epsilon)\, m\, \nu^*$.*

Proof: Consider problem instance $\mathcal{P}^m$ with $m \geq \frac{4}{\epsilon}$. For this problem instance, we have $\lambda_1 = \frac{1}{2}$, $\phi_{11}(\{1\}) = \frac{1}{2}$ and $r_1 = 1$, so by the second maximization problem in (11), $\nu^* \geq \lambda_1\, \phi_{11}(\{1\})\, r_1 = \frac{1}{4}$.

Since $\epsilon m \geq 4$, we get $\epsilon m\nu^* \geq 1$. By Lemma A.5, we have $\mathsf{opt} \geq \frac{m}{4}$. By Lemma A.6, we have $m\nu^* - 1 \leq \frac{m}{4}$. Thus, $\mathsf{opt} \geq \frac{m}{4} \geq m\nu^* - 1 \geq m\nu^* - \epsilon m\nu^* = (1-\epsilon)\, m\nu^*$. ■

Proposition A.4 gives an upper bound on the benefit from customization, whereas Proposition A.7 shows that there are problem instances for which this upper bound can be achieved.

Appendix B: Computational Complexity

In this section, we give a proof for Theorem 5.1. If the customers choose according to the multinomial logit model, then we can compute the linear programming-based surrogate $f(\boldsymbol{c})$ by using the optimal objective value of problem (5). Considering the case where all product revenues are equal to each other, without loss of generality, we assume that $r_i = 1$ for all $i \in \mathcal{N}$. Therefore, if the customers choose according to the multinomial logit model and all product revenues are equal to each other, then problem (3) is equivalent to

$$\max_{(\boldsymbol{c},\boldsymbol{y},\boldsymbol{y}_0)\in\mathbb{Z}_+^n\times\mathbb{R}_+^{nm+m}} \left\{ \sum_{j\in\mathcal{M}}\sum_{i\in\mathcal{N}} y_{ij} \;:\; \sum_{i\in\mathcal{N}} c_i \leq K, \quad \sum_{j\in\mathcal{M}} y_{ij} \leq c_i \quad \forall\, i\in\mathcal{N}, \right. \tag{12}$$

$$\left. \sum_{i\in\mathcal{N}} y_{ij} + y_{0j} \leq \tau_j \quad \forall\, j\in\mathcal{M}, \qquad y_{ij} \leq v_{ij}\, y_{0j} \quad \forall\, i\in\mathcal{N},\; j\in\mathcal{M} \right\}.$$

Our proof of Theorem 5.1 uses a reduction from the maximum coverage problem, which is stated as follows. We are given a set of elements $\mathcal{M} = \{1,\ldots,m\}$, a collection of subsets of elements $\{S_i : i\in\mathcal{N}\}$ with $S_i \subseteq \mathcal{M}$ for all $i\in\mathcal{N}$ and a maximum number of subsets K that we can use. We say that the subset S_i covers element j if $j\in S_i$. In maximum coverage problem, we find at most K subsets to maximize the total number of covered elements. This problem is NP-hard to approximate within a factor better than $1-\frac{1}{e}$ unless $P = NP$; see Feige (1998).

Proof of Theorem 5.1:

Consider an instance of the maximum coverage problem with set of items $\mathcal{M} = \{1,\ldots,m\}$, collection of subsets $\{S_i : i\in\mathcal{N}\}$ and maximum number of subsets to use K. We construct an instance of problem (12) as follows. The set of products corresponds to the collection of subsets $\mathcal{N}$. The set of customer types corresponds to the set of items $\mathcal{M}$. The storage capacity corresponds to the maximum number of subsets to use K. The expected number of arrivals of customers of type j is $\tau_j = 1/m$. Fix any $\epsilon\in(0,1)$. Letting $\gamma = \frac{1}{\epsilon} - 1$, the preference weights are

$$v_{ij} = \begin{cases} \gamma & \text{if } j\in S_i \\ 0 & \text{otherwise.} \end{cases}$$

We use $\boldsymbol{x} = (x_i : i\in\mathcal{N}) \in \{0,1\}^{|\mathcal{N}|}$ to denote a solution to the maximum coverage problem, where $x_i = 1$ if and only if we use subset S_i. First, assuming that there exists a feasible maximum coverage solution $\boldsymbol{x}^*$ with an objective value of Z^*, we construct a feasible solution $(\widehat{\boldsymbol{c}}, \widehat{\boldsymbol{y}}, \widehat{\boldsymbol{y}}_0)$ to (12) with an objective value of at least $\frac{1}{m}(1-\epsilon)\,Z^*$. In particular, we set $\widehat{c}_i = x_i^*$,

$$\widehat{y}_{ij} = \frac{v_{ij}\,x_i^*}{m\,(1+\sum_{k\in\mathcal{N}} v_{kj}\,x_k^*)}, \qquad \widehat{y}_{0j} = \frac{1}{m\,(1+\sum_{k\in\mathcal{N}} v_{kj}\,x_k^*)}. \tag{13}$$

Since $\boldsymbol{x}^*$ is a solution to the maximum coverage problem, we have $\sum_{i\in\mathcal{N}} x_i^* \le K$, so $\sum_{i\in\mathcal{N}} \widehat{c}_i \le K$, which implies that the solution $(\widehat{\boldsymbol{c}}, \widehat{\boldsymbol{y}}, \widehat{\boldsymbol{y}}_0)$ satisfies the first constraint in (12). Note that $\widehat{y}_{ij} \le x_i^*/m$ by (13), in which case, we get $\sum_{j\in\mathcal{M}} \widehat{y}_{ij} \le x_i^* = \widehat{c}_i$, so the solution $(\widehat{\boldsymbol{c}}, \widehat{\boldsymbol{y}}, \widehat{\boldsymbol{y}}_0)$ satisfies the second constraint in (12). By the definition of $\widehat{y}_{ij}$ and $\widehat{y}_{0j}$ in (13), we have $\sum_{i\in\mathcal{N}} \widehat{y}_{ij} + \widehat{y}_{0j} = 1/m = \tau_j$, so the solution $(\widehat{\boldsymbol{c}}, \widehat{\boldsymbol{y}}, \widehat{\boldsymbol{y}}_0)$ satisfies the third constraint in (12). Lastly, once again, by the definition of $\widehat{y}_{ij}$ and $\widehat{y}_{0j}$ in (13), we have $\widehat{y}_{ij}/\widehat{y}_{0j} = v_{ij}\,x_i^* \le v_{ij}$, so the solution $(\widehat{\boldsymbol{c}}, \widehat{\boldsymbol{y}}, \widehat{\boldsymbol{y}}_0)$ satisfies the fourth constraint in (12) as well. Thus, the solution $(\widehat{\boldsymbol{c}}, \widehat{\boldsymbol{y}}, \widehat{\boldsymbol{y}}_0)$ is feasible to problem (12). Using $\mathbf{1}(\cdot)$ to denote the indicator function, for the maximum coverage problem, the solution $\boldsymbol{x}^*$ provides an objective value of $Z^* = \sum_{j\in\mathcal{M}} \max_{i\in\mathcal{N}}\{\mathbf{1}(j\in S_i)\,x_i^*\}$, where we use the fact that $\max_{i\in\mathcal{N}}\{\mathbf{1}(j\in S_i)\,x_i^*\} = 1$ if and only if we have a subset in the solution that covers element j. In this case, for problem (12), the solution $(\widehat{\boldsymbol{c}}, \widehat{\boldsymbol{y}}, \widehat{\boldsymbol{y}}_0)$ provides an objective value of

$$\begin{aligned}
\sum_{j\in\mathcal{M}}\sum_{i\in\mathcal{N}} \widehat{y}_{ij} &= \frac{1}{m}\sum_{j\in\mathcal{M}} \frac{\sum_{i\in\mathcal{N}} v_{ij}\,x_i^*}{1+\sum_{i\in\mathcal{N}} v_{ij}\,x_i^*} \\
&\overset{(a)}{=} \frac{1}{m}\sum_{j\in\mathcal{M}} \frac{\gamma\sum_{i\in\mathcal{N}} \mathbf{1}(j\in S_i)\,x_i^*}{1+\gamma\sum_{i\in\mathcal{N}} \mathbf{1}(j\in S_i)\,x_i^*} \overset{(b)}{\ge} \frac{1}{m}\sum_{j\in\mathcal{M}} \frac{\gamma\,\max_{i\in\mathcal{N}}\{\mathbf{1}(j\in S_i)\,x_i^*\}}{1+\gamma\,\max_{i\in\mathcal{N}}\{\mathbf{1}(j\in S_i)\,x_i^*\}} \\
&\overset{(c)}{=} \frac{1}{m}\sum_{j\in\mathcal{M}} \frac{\gamma}{1+\gamma}\,\max_{i\in\mathcal{N}}\{\mathbf{1}(j\in S_i)\,x_i^*\} \overset{(d)}{=} \frac{1}{m}\,(1-\epsilon)\,Z^*,
\end{aligned}$$

where (a) holds by the definition of v_{ij}, (b) holds because $z/(1+z)$ is increasing in z, (c) uses the fact that if $z\in\{0,1\}$, then $\frac{\gamma z}{1+\gamma z} = \frac{\gamma z}{1+\gamma}$ and (d) uses the definition of γ.

Second, assuming that there exists a feasible solution $(\boldsymbol{c}^*, \boldsymbol{y}^*, \boldsymbol{y}_0^*)$ to problem (12) with an objective value of R^*, we construct a feasible solution $\widehat{\boldsymbol{x}}$ to the maximum coverage problem with an objective value of at least mR^*. In particular, we set $\widehat{x}_i = c_i^*$. Since $(\boldsymbol{c}^*, \boldsymbol{y}^*, \boldsymbol{y}_0^*)$ is feasible to (12), by the third constraint, we have $y_{ij}^* \le \tau_j = \frac{1}{m}$, so the left side of the second constraint satisfies $\sum_{j\in\mathcal{M}} y_{ij}^* \le 1$. Thus, there is no reason to use a value for c_i^* that is strictly larger than one and we can assume that $c_i^* \in \{0,1\}$. Therefore, we have $\widehat{x}_i \in \{0,1\}$. Also, by the first constraint in (12), we have $\sum_{i\in\mathcal{N}} \widehat{x}_i = \sum_{i\in\mathcal{N}} c_i^* \le K$, which implies that the solution $\widehat{\boldsymbol{x}}$ is feasible to the maximum coverage problem. By the third constraint in (12), we have $\sum_{i\in\mathcal{N}} y_{ij}^* \le \tau_j = 1/m \le 1$. By the second constraint in (12), we have $y_{ij}^* \le c_i^* = \widehat{x}_i \le 1$, whereas by the third constraint in (12) and the definition of v_{ij}, we have $y_{ij}^* \le \gamma\,\mathbf{1}(j\in S_i)\,y_{0j}^*$. The last two inequalities imply that $y_{ij}^* \le \mathbf{1}(j\in S_i)\,\widehat{x}_i$, but since $\sum_{i\in\mathcal{N}} y_{ij}^* \le 1$, we get $\sum_{i\in\mathcal{N}} y_{ij}^* \le \max_{i\in\mathcal{N}}\{\mathbf{1}(j\in S_i)\,\widehat{x}_i\}$. In this case, having $\sum_{i\in\mathcal{N}} y_{ij}^* \le$

$\max_{i \in \mathcal{N}}\{\mathbf{1}(j \in S_i)\,\widehat{x}_i\}$ and $\sum_{i \in \mathcal{N}} y^*_{ij} \leq 1/m$ implies that $\sum_{i \in \mathcal{N}} y^*_{ij} \leq \frac{1}{m} \max_{i \in \mathcal{N}}\{\mathbf{1}(j \in S_i)\,\widehat{x}_i\}$. Thus, for the maximum coverage problem, the solution $\widehat{\boldsymbol{x}}$ provides an objective value of

$$\sum_{j \in \mathcal{M}} \max_{i \in \mathcal{N}}\{\mathbf{1}(j \in S_i)\,\widehat{x}_i\} \;\geq\; m \sum_{j \in \mathcal{M}} \sum_{i \in \mathcal{N}} y^*_{ij} \;=\; mR^*.$$

By the discussion so far, given any feasible solution to the maximum coverage problem with objective value Z^*, we can construct a feasible solution to problem (12) with objective value $\widehat{R}$ such that $\widehat{R} \geq \frac{1}{m}\,(1-\epsilon)\,Z^*$. Furthermore, given any feasible solution to problem (12) with objective value R^*, we can construct a feasible solution to the maximum coverage problem with objective value $\widehat{Z}$ such that $\frac{1}{m}\,\widehat{Z} \geq R^*$. Unless $P = NP$, we know that it is NP-hard to approximate the maximum coverage problem within a factor better than $1 - \frac{1}{e}$, which implies that it is also NP-hard to approximate problem (12) within a factor better than $(1-\frac{1}{e})\,(1-\epsilon) = 1 - \frac{1}{e} - \epsilon\,(1-\frac{1}{e})$.

Appendix C: Counterexample to Submodularity of Linear Programming-Based Surrogate

We give a counterexample to demonstrate that $f(\boldsymbol{c})$ is not submodular in $\boldsymbol{c}$ under the multinomial logit model even when we have a single customer type. Consider an instance of problem (4) with $n = 3$ and $m = 1$. The product revenues and preference weights are given by $(r_1, r_2, r_3) = (3, 2, 1)$ and $(v_{11}, v_{21}, v_{31}) = (1, 1, 100)$. For the single customer type, the total expected number of customer arrivals is $\tau_1 = 1$. Considering the vectors $\boldsymbol{c} = (0, 1, 1)$, $\boldsymbol{b} = (0, 0, 1)$ and $\boldsymbol{e}_1 = (1, 0, 0)$, solving (4), we can verify that $f(\boldsymbol{c}) = 1$, $f(\boldsymbol{c} + \boldsymbol{e}_1) = 5/3$, $f(\boldsymbol{b}) = 100/101$ and $f(\boldsymbol{b} + \boldsymbol{e}_1) = 3/2$. We have $\boldsymbol{c} \geq \boldsymbol{b}$, but $f(\boldsymbol{c} + \boldsymbol{e}_1) - f(\boldsymbol{c}) = 2/3 > 103/202 = f(\boldsymbol{b} + \boldsymbol{e}_1) - f(\boldsymbol{b})$, so $f(\boldsymbol{c})$ is not submodular in $\boldsymbol{c}$.

Appendix D: Submodularity of Approximate Surrogate

We use the following lemma in the proof of Theorem 5.3, where we show that $f_{\mathsf{app}}(\boldsymbol{c})$ is submodular in $\boldsymbol{c}$. Recall that $L(\boldsymbol{c}, \boldsymbol{\mu}) = \sum_{i \in \mathcal{N}} c_i\,\mu_i$ and

$$G_j(\boldsymbol{\mu}) = \max_{\boldsymbol{z}_j \in \mathbb{R}^n_+} \left\{ \sum_{i \in \mathcal{N}} (r_i - \mu_i)\,z_{ij} \;:\; \sum_{i \in \mathcal{N}} z_{ij} \leq 1,\;\; z_{ij} \leq v_{ij} \;\forall\, i \in \mathcal{N} \right\}. \tag{14}$$

Lemma D.1 (Weak DR-Submodularity) *For all $\boldsymbol{c}, \boldsymbol{b} \in \mathbb{R}^n_+$ with $\boldsymbol{c} \geq \boldsymbol{b}$ and $\boldsymbol{\mu}, \boldsymbol{\eta} \in \mathbb{R}^n_+$, we have the inequalities*

$$\begin{aligned} L(\boldsymbol{c}, \boldsymbol{\mu}) + L(\boldsymbol{b}, \boldsymbol{\eta}) \;&\geq\; L(\boldsymbol{c}, \boldsymbol{\mu} \wedge \boldsymbol{\eta}) + L(\boldsymbol{b}, \boldsymbol{\mu} \vee \boldsymbol{\eta}) \\ G_j(\boldsymbol{\mu}) + G_j(\boldsymbol{\eta}) \;&\geq\; G_j(\boldsymbol{\mu} \wedge \boldsymbol{\eta}) + G_j(\boldsymbol{\mu} \vee \boldsymbol{\eta}). \end{aligned}$$

The function $G_j : \mathbb{R}^n_+ \to \mathbb{R}_+$ is said to be weak-DR submodular if it satisfies the second inequality in the lemma above. To give the proof of the lemma above, we start with an observation. Let the functions $p, q : \mathbb{R} \to \mathbb{R}$ be continuous over $[0, h]$ with finite numbers of points of non-differentiability. If $p'(x) \geq q'(x)$ at all $x \in [0, h]$ where both p and q are differentiable, then $p(h) - p(0) \geq q(h) - q(0)$. We use this observation, along with the next auxiliary lemma, in the proof for Lemma D.1. In the next auxiliary lemma, let $\boldsymbol{z}_j^*(\boldsymbol{\mu})$ be an optimal solution to problem (14) as a function of $\boldsymbol{\mu}$.

Lemma D.2 (Optimal Solution to Knapsack) *There exists an optimal solution to problem* (14) *such that if* $\boldsymbol{\mu} \geq \boldsymbol{\eta}$ *and* $\mu_i = \eta_i$ *for some* $i \in \mathcal{N}$*, then* $z_{ij}^*(\boldsymbol{\mu}) \geq z_{ij}^*(\boldsymbol{\eta})$.

Proof: Fix $i \in \mathcal{N}$ such that $\mu_i = \eta_i$. If the objective function of the decision variable z_{ij} is negative, then we can set the value of this decision variable to zero at an optimal solution to problem (14). Thus, if $r_i - \mu_i = r_i - \eta_i \leq 0$, then we immediately have $z_{ij}^*(\boldsymbol{\mu}) = 0 = z_{ij}^*(\boldsymbol{\eta})$. Consider the case $r_i - \mu_i = r_i - \eta_i > 0$. Problem (14) is a knapsack problem, so we can obtain an optimal solution to this problem by sorting the decision variables according to their objective function coefficients and filling the capacity of the knapsack starting from the decision variable with the largest objective function coefficient. Thus, letting $A_{\boldsymbol{\mu}} = \{k \in \mathcal{N} \setminus \{i\} : r_k - \mu_k \geq r_i - \mu_i\}$, we have $z_{ij}^*(\boldsymbol{\mu}) = \min\{v_{ij}, [1 - \sum_{k \in A_{\boldsymbol{\mu}}} v_{kj}]^+\}$. Similarly, we have $z_{ij}^*(\boldsymbol{\eta}) = \min\{v_{ij}, [1 - \sum_{k \in A_{\boldsymbol{\eta}}} v_{kj}]^+\}$, where we let $A_{\boldsymbol{\eta}} = \{k \in \mathcal{N} \setminus \{i\} : r_k - \eta_k \geq r_i - \eta_i\}$. Noting that $\boldsymbol{\mu} \geq \boldsymbol{\eta}$ and $\mu_i = \eta_i$, we have $A_{\boldsymbol{\mu}} \subseteq A_{\boldsymbol{\eta}}$, in which case, we get $z_{ij}^*(\boldsymbol{\mu}) = \min\{v_{ij}, [1 - \sum_{k \in A_{\boldsymbol{\mu}}} v_{kj}]^+\} \geq \min\{v_{ij}, [1 - \sum_{k \in A_{\boldsymbol{\eta}}} v_{kj}]^+\} = z_{ij}^*(\boldsymbol{\eta})$. ■

Proof of Lemma D.1:

We have $x + y = (x \wedge y) + (x \vee y)$, so that $x - (x \wedge y) = (x \vee y) - y$. In this case, letting $\delta_i = \mu_i - (\mu_i \wedge \eta_i) = (\mu_i \vee \eta_i) - \eta_i \geq 0$, noting that $\boldsymbol{c} \geq \boldsymbol{b}$, we have

$$\begin{aligned} L(\boldsymbol{c}, \boldsymbol{\mu}) - L(\boldsymbol{c}, \boldsymbol{\mu} \wedge \boldsymbol{\eta}) \;&=\; \sum_{i \in \mathcal{N}} c_i(\mu_i - (\mu_i \wedge \eta_i)) \;=\; \sum_{i \in \mathcal{N}} c_i\, \delta_i \\ &\geq\; \sum_{i \in \mathcal{N}} b_i\, \delta_i \;=\; \sum_{i \in \mathcal{N}} b_i((\mu_i \vee \eta_i) - \eta_i) \;=\; L(\boldsymbol{b}, \boldsymbol{\mu} \vee \boldsymbol{\eta}) - L(\boldsymbol{b}, \boldsymbol{\eta}). \end{aligned}$$

The chain of inequalities above establishes that the first inequality in the lemma holds. We turn to the second inequality in the lemma. Using the decision variables $\boldsymbol{x} = (x_1, \ldots, x_n)$ and letting $\mathcal{X} \subseteq \mathbb{R}_+^n$ be a polytope, consider the generic linear program $\max_{\boldsymbol{x} \in \mathcal{X}} \sum_{i=1}^n c_i\, x_i$. When viewed as a function of the objective function coefficients, let $\mathsf{LP}(\boldsymbol{c})$ be the optimal objective value and $\boldsymbol{x}^*(\boldsymbol{c})$ be an optimal solution to the linear program. By linear programming theory, if LP is differentiable at $\boldsymbol{b}$, then $\frac{\partial \mathsf{LP}(\boldsymbol{c})}{\partial c_i}|_{\boldsymbol{c}=\boldsymbol{b}} = x_i^*(\boldsymbol{b})$. Furthermore, $\mathsf{LP}(\boldsymbol{c})$ is continuous in $\boldsymbol{c}$ and it has a finite number of points of non-differentiability. To show the second inequality in the lemma, we use an equivalent definition of a weak-DR submodular function; see Proposition 1 in Bian et al. (2017). The function G_j is weak-DR submodular if and only if for all $h \in \mathbb{R}_+$ and $\boldsymbol{\mu}, \boldsymbol{\eta} \in \mathbb{R}_+^n$ with $\boldsymbol{\mu} \geq \boldsymbol{\eta}$ and $\mu_i = \eta_i$ for some $i \in \mathcal{N}$, we have $G_j(\boldsymbol{\mu} + h\, \boldsymbol{e}_i) - G_j(\boldsymbol{\mu}) \leq G_j(\boldsymbol{\eta} + h\, \boldsymbol{e}_i) - G_j(\boldsymbol{\eta})$. Thus, consider $\boldsymbol{\mu} \geq \boldsymbol{\eta}$ and $\mu_i = \eta_i$ for some $i \in \mathcal{N}$. Fixing $i \in \mathcal{N}$, let $g_{\boldsymbol{\mu}}(t) = G_j(\boldsymbol{\mu} + t\,\boldsymbol{e}_i)$, so that $g_{\boldsymbol{\mu}}(0) = G_j(\boldsymbol{\mu})$ and $g_{\boldsymbol{\mu}}(h) = G_j(\boldsymbol{\mu} + h\,\boldsymbol{e}_i)$. Similarly, let $g_{\boldsymbol{\eta}}(t) = G_j(\boldsymbol{\eta} + t\,\boldsymbol{e}_i)$. By the earlier discussion in this paragraph, if $g_{\boldsymbol{\mu}}$ and $g_{\boldsymbol{\eta}}$ are both differentiable at t, then $g'_{\boldsymbol{\mu}}(t) = -z^*_{ij}(\boldsymbol{\mu} + t\,\boldsymbol{e}_i)$ and $g'_{\boldsymbol{\eta}}(t) = -z^*_{ij}(\boldsymbol{\eta} + t\,\boldsymbol{e}_i)$. Furthermore, $g_{\boldsymbol{\mu}}(t)$ and $g_{\boldsymbol{\eta}}(t)$ are continuous in t and have finite numbers of points of non-differentiability. Since $\boldsymbol{\mu} \geq \boldsymbol{\eta}$ and $\mu_i = \eta_i$, by Lemma D.2, $g'_{\boldsymbol{\mu}}(t) = -z^*_{ij}(\boldsymbol{\mu} + t\,\boldsymbol{e}_i) \leq -z^*_{ij}(\boldsymbol{\eta} + t\,\boldsymbol{e}_i) = g'_{\boldsymbol{\eta}}(t)$, in which case, by the observation right before Lemma D.2, we obtain $g_{\boldsymbol{\mu}}(h) - g_{\boldsymbol{\mu}}(0) \leq g_{\boldsymbol{\eta}}(h) - g_{\boldsymbol{\eta}}(0)$. Noting that $g_{\boldsymbol{\mu}}(0) = G_j(\boldsymbol{\mu})$, $g_{\boldsymbol{\mu}}(h) = G_j(\boldsymbol{\mu} + h\,\boldsymbol{e}_i)$, $g_{\boldsymbol{\eta}}(0) = G_j(\boldsymbol{\eta})$ and $g_{\boldsymbol{\eta}}(h) = G_j(\boldsymbol{\eta} + h\,\boldsymbol{e}_i)$, the last inequality shows that the second inequality in the lemma holds. ■

Appendix E: Incorporating Salvage Values

We consider incorporating salvage values for the units left at the end of the selling horizon. For economy of space, we focus on the case where the customers choose under the multinomial logit model and show that we can slightly modify our approximation framework to obtain a $\frac{1}{4}(1 - \frac{1}{e})$-approximate solution. We can also slightly modify our approximation framework to get an asymptotically optimal solution for large storage capacity. We do not give the proofs of our claims as they are similar to other proofs in the paper. We continue using the notation introduced in Section 2 with the addition that we obtain a salvage value of $\alpha_i \in [0, r_i]$ for each unit of product i. We can compute the optimal assortment customization policy using the dynamic program

$$\overline{J}_t(\boldsymbol{x}) = \sum_{j \in \mathcal{M}} \lambda_{jt} \max_{S \subseteq \mathcal{N}(\boldsymbol{x})} \left\{ \sum_{i \in \mathcal{N}} \phi_{ij}(S) \Big[r_i + \overline{J}_{t+1}(\boldsymbol{x} - \boldsymbol{e}_i) - \overline{J}_{t+1}(\boldsymbol{x}) \Big] \right\} + \overline{J}_{t+1}(\boldsymbol{x}), \tag{15}$$

with the boundary condition that $\overline{J}_{T+1}(\boldsymbol{x}) = \sum_{i \in \mathcal{N}} \alpha_i\, x_i$. If the state variable at the beginning of the selling horizon is $\boldsymbol{x}$, then the optimal total expected revenue is $\overline{J}_1(\boldsymbol{x})$.

Equivalent Dynamic Program:

The boundary condition of the dynamic program in (15) involves a non-zero value function, which prevents us from directly using our approximation framework to handle salvage values. To get around this difficulty, we formulate an alternative dynamic program such that the boundary condition of the alternative dynamic program involves a zero value function. It will turn out that the alternative dynamic program is equivalent to the dynamic program in (15) in the sense that we can obtain the value functions in (15) by solving the alternative dynamic program. In the alternative dynamic program, we obtain a revenue of $r_i - \alpha_i$ for each unit of product i that we sell at any time period in the selling horizon, so we consider the dynamic program

$$J_t(\boldsymbol{x}) = \sum_{j \in \mathcal{M}} \lambda_{jt} \max_{S \subseteq \mathcal{N}(\boldsymbol{x})} \left\{ \sum_{i \in \mathcal{N}} \phi_{ij}(S) \Big[(r_i - \alpha_i) + J_{t+1}(\boldsymbol{x} - \boldsymbol{e}_i) - J_{t+1}(\boldsymbol{x}) \Big] \right\} + J_{t+1}(\boldsymbol{x}), \tag{16}$$

with the boundary condition that $J_{T+1} = 0$. We can use induction over the time periods to show that $\overline{J}_t(\boldsymbol{x}) = J_t(\boldsymbol{x}) + \sum_{i \in \mathcal{N}} \alpha_i\, x_i$. Thus, we can solve the dynamic program in (16) to compute the value functions in (15). We can make the optimal stocking decisions by solving the problem $\mathsf{opt} = \max_{\boldsymbol{c}\in\mathbb{Z}_+^n} \big\{ \overline{J}_1(\boldsymbol{c}) : \sum_{i\in\mathcal{N}} c_i \le K \big\} = \max_{\boldsymbol{c}\in\mathbb{Z}_+^n} \big\{ J_1(\boldsymbol{c}) + \sum_{i\in\mathcal{N}} \alpha_i\, c_i : \sum_{i\in\mathcal{N}} c_i \le K \big\}$.

Surrogate Function:

To approximate the value function at the first time period computed through the dynamic program in (16), we use the surrogate function given by

$$f(\boldsymbol{c}) \;=\; \max_{\boldsymbol{w}\in\mathbb{R}_+^{m\,2^n}} \Bigg\{ \sum_{j\in\mathcal{M}} \sum_{S\subseteq\mathcal{N}} \sum_{i\in\mathcal{N}} (r_i-\alpha_i)\, \phi_{ij}(S)\, w_j(S) \;:\; \sum_{j\in\mathcal{M}} \sum_{S\subseteq\mathcal{N}} \phi_{ij}(S)\, w_j(S) \le c_i \quad \forall\, i\in\mathcal{N}, \quad \sum_{S\subseteq\mathcal{N}} w_j(S) \le \tau_j \quad \forall\, j\in\mathcal{M} \Bigg\}, \tag{17}$$

where we recall that $\tau_j = \sum_{t\in\mathcal{T}} \lambda_{jt}$, corresponding to the total expected number of customer arrivals of type j over the selling horizon. We can show that the optimal objective value of the problem above is an upper bound on the value function at the first time period in (16) computed at the stocking quantities $\boldsymbol{c}$. In other words, we have $f(\boldsymbol{c}) \ge J_1(\boldsymbol{c})$. Thus, a natural approach for making the stocking decisions is to solve the problem $\max_{\boldsymbol{c}\in\mathbb{Z}_+^n} \big\{ f(\boldsymbol{c}) + \sum_{i\in\mathcal{N}} \alpha_i\, c_i : \sum_{i\in\mathcal{N}} c_i \le K \big\}$. However, we can come up with examples to demonstrate that $f(\boldsymbol{c})$ is not submodular in $\boldsymbol{c}$, preventing us from designing an efficient approach to obtain a solution to the last problem. Instead, we work with an approximate surrogate, which we construct next.

Approximate Surrogate:

Under the assumption that the customers choose according to the multinomial logit model, we consider an approximation to the surrogate in (17) given by

$$f_{\mathsf{app}}(\boldsymbol{c}) \;=\; \max_{\boldsymbol{y}\in\mathbb{R}_+^{nm}} \Bigg\{ \sum_{j\in\mathcal{M}} \sum_{i\in\mathcal{N}} (r_i-\alpha_i)\, y_{ij} \;:\; \sum_{j\in\mathcal{M}} y_{ij} \le c_i \quad \forall\, i\in\mathcal{N}, \quad \sum_{i\in\mathcal{N}} y_{ij} \le \frac{1}{2}\,\tau_j \quad \forall\, j\in\mathcal{M}, \qquad y_{ij} \le \frac{1}{2}\, v_{ij}\,\tau_j \quad \forall\, i\in\mathcal{N},\; j\in\mathcal{M} \Bigg\}. \tag{18}$$

In the problem above, if we introduce the additional decision variables $(y_{0j} : j\in\mathcal{M})$ and replace the last two constraints, respectively, with $\sum_{i\in\mathcal{N}} y_{ij} + y_{0j} \le \tau_j$ for all $j\in\mathcal{M}$ and $y_{ij}/v_{ij} \le y_{0j}$ for all $i\in\mathcal{N}$, $j\in\mathcal{M}$, then the optimal objective value of the problem above is equal to that of problem (17). The way problem (18) is formulated, we can show that its optimal objective value satisfies $\frac{1}{2} f(\boldsymbol{c}) \le f_{\mathsf{app}}(\boldsymbol{c}) \le f(\boldsymbol{c})$. Furthermore, we can show that $f_{\mathsf{app}}(\boldsymbol{c})$ is monotone and submodular in $\boldsymbol{c}$. To make the stocking decisions, we solve the problem $\max_{\boldsymbol{c}\in\mathbb{Z}_+^n} \big\{ f_{\mathsf{app}}(\boldsymbol{c}) + \sum_{i\in\mathcal{N}} \alpha_i\, c_i : \sum_{i\in\mathcal{N}} c_i \le K \big\}$. Noting that $f_{\mathsf{app}}(\boldsymbol{c})$ is monotone and submodular in $\boldsymbol{c}$, we can obtain a $(1-\frac{1}{e})$-approximate solution to this problem efficiently. We use $\widehat{\boldsymbol{c}}$ to denote such a $(1-\frac{1}{e})$-approximate solution.

In this way, the $(1-\frac{1}{e})$-approximate solution $\widehat{\boldsymbol{c}}$ obtained through the approach discussed in the previous paragraph corresponds to our approximate stocking quantities. Our approach for obtaining this approximate solution also uses the fact that if $f_{\mathsf{app}}(\boldsymbol{c})$ is submodular in $\boldsymbol{c}$, then $f_{\mathsf{app}}(\boldsymbol{c}) + \sum_{i\in\mathcal{N}} \alpha_i\, c_i$ is still submodular in $\boldsymbol{c}$. We focus on the assortment customization decisions.

Assortment Customization Policy:

To make the assortment customization decisions, we solve problem (17) with $\boldsymbol{c} = \widehat{\boldsymbol{c}}$ once at the beginning of the selling horizon and use $\widehat{\boldsymbol{w}}$ to denote the optimal solution. In our assortment customization policy, if we are at time period t with the remaining inventories given by the vector $\boldsymbol{x}$ and a customer of type j arrives, then we sample an assortment $\widehat{S}$ from the probability distribution characterized by $\{\widehat{w}_j(S)/\tau_j : S \subseteq \mathcal{N}\}$ and offer the assortment $\widehat{S} \cap \mathcal{N}(\boldsymbol{x})$. The total expected revenue from our assortment customization policy is characterized by the dynamic program

$$\overline{V}_t(\boldsymbol{x}) = \sum_{j\in\mathcal{M}} \lambda_{jt} \sum_{S\subseteq\mathcal{N}} \frac{\widehat{w}_j(S)}{\tau_j} \left\{ \sum_{i\in\mathcal{N}} \phi_{ij}(S\cap\mathcal{N}(\boldsymbol{x})) \Big[r_i + \overline{V}_{t+1}(\boldsymbol{x} - \boldsymbol{e}_i) - \overline{V}_{t+1}(\boldsymbol{x}) \Big] \right\} + \overline{V}_{t+1}(\boldsymbol{x}), \quad (19)$$

with the boundary condition that $\overline{V}_{T+1}(\boldsymbol{x}) = \sum_{i\in\mathcal{N}} \alpha_i\, x_i$. Starting with the stocking quantities $\widehat{\boldsymbol{c}}$, the total expected revenue from our assortment customization policy is $\overline{V}_1(\widehat{\boldsymbol{c}})$.

Alternative Accounting for Total Expected Revenue:

When we use our assortment customization policy, we obtain a revenue of r_i each time a customer purchases product i. Thus, in (19), the immediate reward associated with product i is r_i. In (18), considering the approximate surrogate that we use to make our inventory stocking decisions, we obtain a revenue of $r_i - \alpha_i$ each time a customer purchases product i. To get around this discrepancy, we give a dynamic program equivalent to the one in (19), but if a customer purchases product i, then we obtain a revenue of $r_i - \alpha_i$. Consider the dynamic program

$$V_t(\boldsymbol{x}) = \sum_{j\in\mathcal{M}} \lambda_{jt} \sum_{S\subseteq\mathcal{N}} \frac{\widehat{w}_j(S)}{\tau_j} \left\{ \sum_{i\in\mathcal{N}} \phi_{ij}(S\cap\mathcal{N}(\boldsymbol{x})) \Big[(r_i - \alpha_i) + V_{t+1}(\boldsymbol{x} - \boldsymbol{e}_i) - V_{t+1}(\boldsymbol{x}) \Big] \right\} + V_{t+1}(\boldsymbol{x}), (20)$$

with the boundary condition that $V_{T+1} = 0$. We can use induction over the time periods to show that $\overline{V}_t(\boldsymbol{x}) = V_t(\boldsymbol{x}) + \sum_{i\in\mathcal{N}} \alpha_i\, x_i$. Thus, (19) and (20) are equivalent to each other.

Performance Guarantee:

Using the surrogate in (17) and the dynamic program in (20), we can show that $V_1(\widehat{\boldsymbol{c}}) \geq \frac{1}{2}\, f(\widehat{\boldsymbol{c}})$. This result is the analogue of Theorem 7.1 after noting that both (17) and (20) use the assumption that we obtain a revenue of $r_i - \alpha_i$ each time a customer purchases product i. The stocking quantities from our approximation framework are given by $\widehat{\boldsymbol{c}}$, which is a $(1 - \frac{1}{e})$-approximate solution to the problem $\max_{\boldsymbol{c}\in\mathbb{Z}_+^n} \big\{ f_{\mathsf{app}}(\boldsymbol{c}) + \sum_{i\in\mathcal{N}} \alpha_i\, c_i : \sum_{i\in\mathcal{N}} c_i \leq K \big\}$. Starting with the stocking quantities $\widehat{\boldsymbol{c}}$, the total expected revenue from our assortment customization policy is $\overline{V}_1(\widehat{\boldsymbol{c}})$.

To get the performance guarantee of $\frac{1}{4}\,(1 - \frac{1}{e})$ for our approximation framework, we proceed as follows. Let $\boldsymbol{c}^*$ be an optimal solution to $\mathsf{opt} = \max_{\boldsymbol{c}\in\mathbb{Z}_+^n} \big\{ \overline{J}_1(\boldsymbol{c}) : \sum_{i\in\mathcal{N}} c_i \leq K \big\}$. By the discussion so far in this section, we have the following inequalities. First, we have $V_1(\widehat{\boldsymbol{c}}) \geq \frac{1}{2} f(\widehat{\boldsymbol{c}})$. Second, we have $f(\widehat{\boldsymbol{c}}) \geq f_{\mathsf{app}}(\widehat{\boldsymbol{c}})$. Third, we have $f_{\mathsf{app}}(\boldsymbol{c}^*) \geq \frac{1}{2} f(\boldsymbol{c}^*)$. Fourth, we have $f(\boldsymbol{c}^*) \geq J_1(\boldsymbol{c}^*)$. Thus, we get

$$\begin{aligned}
\overline{V}_1(\widehat{\boldsymbol{c}}) \;&=\; V_1(\widehat{\boldsymbol{c}}) + \sum_{i \in \mathcal{N}} \alpha_i \, \widehat{c}_i \;\overset{(a)}{\geq}\; \frac{1}{2}\Big(f(\widehat{\boldsymbol{c}}) + \sum_{i \in \mathcal{N}} \alpha_i \, \widehat{c}_i \Big) \;\overset{(b)}{\geq}\; \frac{1}{2}\Big(f_{\mathsf{app}}(\widehat{\boldsymbol{c}}) + \sum_{i \in \mathcal{N}} \alpha_i \, \widehat{c}_i \Big) \\
&\overset{(c)}{\geq}\; \frac{1}{2}\Big(1 - \tfrac{1}{e}\Big)\Big(f_{\mathsf{app}}(\boldsymbol{c}^*) + \sum_{i \in \mathcal{N}} \alpha_i \, c_i^* \Big) \;\overset{(d)}{\geq}\; \frac{1}{4}\Big(1 - \tfrac{1}{e}\Big)\Big(f(\boldsymbol{c}^*) + \sum_{i \in \mathcal{N}} \alpha_i \, c_i^* \Big) \\
&\overset{(e)}{\geq}\; \frac{1}{4}\Big(1 - \tfrac{1}{e}\Big)\Big(J_1(\boldsymbol{c}^*) + \sum_{i \in \mathcal{N}} \alpha_i \, c_i^* \Big) \;=\; \frac{1}{4}\Big(1 - \tfrac{1}{e}\Big) \overline{J}_1(\boldsymbol{c}^*) \;=\; \frac{1}{4}\Big(1 - \tfrac{1}{e}\Big) \,\mathsf{opt},
\end{aligned}$$

where (a), (b), (d) and (e) holds by the four inequalities above, whereas (c) holds because $\widehat{\boldsymbol{c}}$ is a $(1 - \frac{1}{e})$-approximate solution to the problem $\max_{\boldsymbol{c} \in \mathbb{Z}_+^n} \big\{ f_{\mathsf{app}}(\boldsymbol{c}) + \sum_{i \in \mathcal{N}} \alpha_i \, c_i : \sum_{i \in \mathcal{N}} c_i \leq K \big\}$.

Appendix F: Performance of Assortment Customization Policy

We give a proof for Theorem 7.1. Let $\widetilde{V}_t(\boldsymbol{x})$ be the total expected revenue of the inventory-agnostic policy over time periods $t, \ldots, T$ when the state at the beginning of time period t is $\boldsymbol{x}$. We have

$$\widetilde{V}_t(\boldsymbol{x}) = \sum_{j \in \mathcal{M}} \lambda_{jt} \sum_{S \subseteq \mathcal{N}} \frac{\widehat{w}_j(S)}{\tau_j} \left\{ \sum_{i \in \mathcal{N}} \phi_{ij}(S) \, \mathbf{1}(x_i \geq 1) \Big[r_i + \widetilde{V}_{t+1}(\boldsymbol{x} - \boldsymbol{e}_i) - \widetilde{V}_{t+1}(\boldsymbol{x}) \Big] \right\} + \widetilde{V}_{t+1}(\boldsymbol{x}), \quad (21)$$

with the boundary condition that $\widetilde{V}_{T+1} = 0$. In the dynamic program above, we use $\mathbf{1}(\cdot)$ to denote the indicator function. The dynamic program above is similar to the one in (1), but the inventory-agnostic policy offers assortment S to a customer of type j with probability $\frac{\widehat{w}_j(S)}{\tau_j}$, in which case, the customer chooses product i with probability $\phi_{ij}(S)$. If we have remaining inventory for the product, then the customer makes a purchase. Otherwise, the customer leaves without making a purchase. Thus, the decisions of the inventory-agnostic policy is pre-fixed by the optimal solution $\widehat{\boldsymbol{w}}$, which is an optimal solution to problem (4) when we solve this problem with $\boldsymbol{c} = \widehat{\boldsymbol{c}}$. To give a proof for Theorem 4.1, we need three auxiliary lemmas. The inventory-agnostic policy offers assortment S to a customer of type j with probability $\frac{\widehat{w}_j(S)}{\tau_j}$. A customer of type j chooses product i out of this assortment with probability $\phi_{ij}(S)$. Thus, under the inventory-agnostic policy, the demand for product i at time period t has Bernoulli distribution with parameter $\beta_{it} = \sum_{j \in \mathcal{M}} \lambda_{jt} \sum_{S \subseteq \mathcal{N}} \frac{\widehat{w}_j(S)}{\tau_j} \, \phi_{ij}(S)$. We let Y_{it} be the Bernoulli random variable with parameter β_{it}, capturing the demand for product i at time period t under the inventory-agnostic policy. Furthermore, we use $Z_{it} = \sum_{\tau = t}^{T} Y_{i\tau}$ to capture the total demand for product i over time periods $t, \ldots, T$. Thus, if the inventory-agnostic policy has x_i unit of inventory for product i at the beginning of time period t, then we can view $\sum_{\tau = t}^{T} \beta_{i\tau}$ as the total expected demand for product i, whereas $\mathbb{E}\{[Z_{it} - x_i]^+\}$ as the expected lost demand due to limited inventory.

In the next lemma, we show that the accounting process discussed in the previous paragraph provides a closed form expression for the value functions in (21).

Lemma F.1 (Equivalent Form for Value Functions) *Letting the value functions $\{\widetilde{V}_t : t \in \mathcal{T}\}$ be computed through the dynamic program in* (21)*, for all $\boldsymbol{x} \in \mathbb{Z}_+^n$ and $t \in \mathcal{T}$, we have*

$$\widetilde{V}_t(\boldsymbol{x}) = \sum_{i \in \mathcal{N}} r_i \left(\sum_{\tau=t}^{T} \beta_{i\tau} - \mathbb{E}\Big\{ [Z_{it} - x_i]^+ \Big\} \right).$$

Proof: We show the result by using induction over the time periods. For time period $T+1$, both sides of the inequality in the lemma is equal to zero, so the result holds at time period $T+1$. Assuming that the result holds at time period $t+1$, we show that the result holds at time period t as well. Note that if $x_i = 0$, then we have $\mathbb{E}\{[Z_{i,t+1} - x_i]^+\} = \mathbb{E}\{Z_{i,t+1}\} = \sum_{\tau=t+1}^{T} \beta_{i\tau}$, which implies that we always have $\sum_{\tau=t+1}^{T} \beta_{i\tau} - \mathbb{E}\{[Z_{i,t+1} - x_i]^+\} = \mathbf{1}(x_i \geq 1)\,(\sum_{\tau=t+1}^{T} \beta_{i\tau} - \mathbb{E}\{[Z_{i,t+1} - x_i]^+\})$. Similarly, we have $\sum_{\tau=t}^{T} \beta_{i\tau} - \mathbb{E}\{[Z_{it} - x_i]^+\} = \mathbf{1}(x_i \geq 1)\,(\sum_{\tau=t}^{T} \beta_{i\tau} - \mathbb{E}\{[Z_{it} - x_i]^+\})$ as well. Using the induction assumption in (21), we obtain the chain of equalities

$$\begin{aligned}
\widetilde{V}_t(\boldsymbol{x}) = {} & \sum_{j \in \mathcal{M}} \lambda_{jt} \sum_{S \subseteq \mathcal{N}} \frac{\widehat{w}_j(S)}{\tau_j} \left\{ \sum_{i \in \mathcal{N}} \phi_{ij}(S)\, \mathbf{1}(x_i \geq 1)\, r_i \Big[1 - \mathbb{E}\Big\{ [Z_{i,t+1} - x_i + 1]^+ \Big\} + \mathbb{E}\Big\{ [Z_{i,t+1} - x_i]^+ \Big\} \Big] \right\} \\
& \qquad + \sum_{i \in \mathcal{N}} r_i \left(\sum_{\tau=t+1}^{T} \beta_{i\tau} - \mathbb{E}\Big\{ [Z_{i,t+1} - x_i]^+ \Big\} \right) \\
\stackrel{(a)}{=} {} & \sum_{i \in \mathcal{N}} \mathbf{1}(x_i \geq 1) \left\{ \beta_{it}\, r_i \Big[1 - \mathbb{E}\Big\{ [Z_{i,t+1} - x_i + 1]^+ \Big\} + \mathbb{E}\Big\{ [Z_{i,t+1} - x_i]^+ \Big\} \Big] \right\} \\
& \qquad + \sum_{i \in \mathcal{N}} \mathbf{1}(x_i \geq 1)\, r_i \left(\sum_{\tau=t+1}^{T} \beta_{i\tau} - \mathbb{E}\Big\{ [Z_{i,t+1} - x_i]^+ \Big\} \right) \\
\stackrel{(b)}{=} {} & \sum_{i \in \mathcal{N}} \mathbf{1}(x_i \geq 1)\, r_i \left\{ \sum_{\tau=t}^{T} \beta_{i\tau} - \beta_{it}\, \mathbb{E}\Big\{ [Z_{i,t+1} - x_i + 1]^+ \Big\} - (1 - \beta_{it})\, \mathbb{E}\Big\{ [Z_{i,t+1} - x_i]^+ \Big\} \right\} \\
\stackrel{(c)}{=} {} & \sum_{i \in \mathcal{N}} \mathbf{1}(x_i \geq 1)\, r_i \left\{ \sum_{\tau=t}^{T} \beta_{i\tau} - \mathbb{E}\Big\{ [Z_{it} - x_i]^+ \Big\} \right\} \\
= {} & \sum_{i \in \mathcal{N}} r_i \left\{ \sum_{\tau=t}^{T} \beta_{i\tau} - \mathbb{E}\Big\{ [Z_{it} - x_i]^+ \Big\} \right\},
\end{aligned}$$

where (a) uses the definition of β_{it} and the fact that if $x_i = 0$, then $\sum_{\tau=t+1}^{T} \beta_{i\tau} - \mathbb{E}\{[Z_{i,t+1} - x_i]^+\} = 0$, (b) holds by arranging the terms and (c) holds because $Z_{it} = Y_{it} + Z_{i,t+1}$ and $\mathbb{P}\{Y_{it} = 1\} = \beta_{it}$. ■

In the next lemma, letting the value functions $\{V_t : t \in \mathcal{T}\}$ and $\{\widetilde{V}_t : t \in \mathcal{T}\}$, respectively, be computed by (8) and (21), we show that $\widetilde{V}_t(\boldsymbol{x})$ lower bounds $V_t(\boldsymbol{x})$.

Lemma F.2 (Lower Bound) *Letting the value functions $\{V_t : t \in \mathcal{T}\}$ and $\{\widetilde{V}_t : t \in \mathcal{T}\}$, respectively, be computed by* (8) *and* (21)*, we have $V_t(\boldsymbol{x}) \geq \widetilde{V}_t(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{Z}_+^n$ and $t \in \mathcal{T}$.*

Proof: We show the result by using induction over the time periods. We have $V_{T+1} = 0 = \widetilde{V}_{T+1}$, so the result holds at time period $T+1$. Assuming that the result holds at time period $t+1$, we show that the result holds at time period t as well. A simple lemma, given as Lemma F.3 shortly, shows that $\widetilde{V}_{t+1}(\boldsymbol{x}) - \widetilde{V}_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i) \le r_i$. If $i \notin S$, then $\phi_{ij}(S)=0$. Also $x_i \ge 1$ if and only if $i \in \mathcal{N}(\boldsymbol{x})$, so $\phi_{ij}(S\cap\mathcal{N}(\boldsymbol{x})) = \mathbf{1}(x_i \ge 1)\,\phi_{ij}(S\cap\mathcal{N}(\boldsymbol{x}))$. Furthermore, arranging the terms, the coefficient of $\widetilde{V}_{t+1}(\boldsymbol{x})$ on the right side of (8) is $1-\sum_{j\in\mathcal{M}}\lambda_{jt}\sum_{S\subseteq\mathcal{N}}\frac{\widehat{w}_j(S)}{\tau_j}\sum_{i\in\mathcal{N}}\phi_{ij}(S\cap\mathcal{N}(\boldsymbol{x}))$, which is non-negative since $\sum_{j\in\mathcal{M}}\lambda_{jt}\le 1$, $\sum_{S\subseteq\mathcal{N}}\widehat{w}_t(S)=\tau_j$ and $\sum_{i\in\mathcal{N}}\phi_{ij}(S\cap\mathcal{N}(\boldsymbol{x}))\le 1$. Thus, noting that $V_{t+1}(\boldsymbol{x}) \ge \widetilde{V}_{t+1}(\boldsymbol{x})$ by the induction assumption, if we replace $V_{t+1}(\boldsymbol{x})$ and $V_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i)$ in (8) with $\widetilde{V}_{t+1}(\boldsymbol{x})$ and $\widetilde{V}_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i)$, then the right side of (8) gets smaller. So, by (8), we get

$$\begin{aligned}
V_t(\boldsymbol{x}) &\ge \sum_{j\in\mathcal{M}}\lambda_{jt}\sum_{S\subseteq\mathcal{N}}\frac{\widehat{w}_j(S)}{\tau_j}\left\{\sum_{i\in\mathcal{N}}\phi_{ij}(S\cap\mathcal{N}(\boldsymbol{x}))\Big[r_i+\widetilde{V}_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i)-\widetilde{V}_{t+1}(\boldsymbol{x})\Big]\right\}+\widetilde{V}_{t+1}(\boldsymbol{x})\\
&= \sum_{j\in\mathcal{M}}\lambda_{jt}\sum_{S\subseteq\mathcal{N}}\frac{\widehat{w}_j(S)}{\tau_j}\left\{\sum_{i\in\mathcal{N}}\mathbf{1}(x_i\ge 1)\,\phi_{ij}(S\cap\mathcal{N}(\boldsymbol{x}))\Big[r_i+\widetilde{V}_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i)-\widetilde{V}_{t+1}(\boldsymbol{x})\Big]\right\}+\widetilde{V}_{t+1}(\boldsymbol{x})\\
&\overset{(a)}{\ge} \sum_{j\in\mathcal{M}}\lambda_{jt}\sum_{S\subseteq\mathcal{N}}\frac{\widehat{w}_j(S)}{\tau_j}\left\{\sum_{i\in\mathcal{N}}\mathbf{1}(x_i\ge 1)\,\phi_{ij}(S)\Big[r_i+\widetilde{V}_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i)-\widetilde{V}_{t+1}(\boldsymbol{x})\Big]\right\}+\widetilde{V}_{t+1}(\boldsymbol{x}) \overset{(b)}{=} \widetilde{V}_t(\boldsymbol{x}),
\end{aligned}$$

where (a) holds because $\phi_{ij}(S)\ge\phi_{ij}(Q)$ for all $i\in S$ and $S\subseteq Q$ by our assumption on the choice probabilities in Section 2 and $r_i \ge \widetilde{V}_{t+1}(\boldsymbol{x})-\widetilde{V}_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i)$ by Lemma F.3, whereas (b) is by (21).

In this next lemma, we give an upper bound on the first difference of the value functions $\{\widetilde{V}_t : t\in\mathcal{T}\}$. We use this lemma in the proof of Lemma F.2.

Lemma F.3 (First Differences) *Letting the value functions* $\{\widetilde{V}_t : t\in\mathcal{T}\}$ *be computed by* (21), *we have* $\widetilde{V}_t(\boldsymbol{x})-\widetilde{V}_t(\boldsymbol{x}-\boldsymbol{e}_i)\le r_i$ *for all* $\boldsymbol{x}\in\mathbb{Z}_+^n$ *such that* $x_i\ge 1$ *and* $t\in\mathcal{T}$.

Proof: For each product i, using the boundary condition that $\widetilde{v}_{i,T+1}=0$, we compute the value functions $\{\widetilde{v}_{it} : t\in\mathcal{T}\}$ through the dynamic program

$$\widetilde{v}_{it}(x_i)=\sum_{j\in\mathcal{M}}\lambda_{jt}\sum_{S\subseteq\mathcal{N}}\frac{\widehat{w}_j(S)}{\tau_j}\left\{\phi_{ij}(S)\,\mathbf{1}(x_i\ge 1)\Big[r_i+\widetilde{v}_{i,t+1}(x_i-1)-\widetilde{v}_{i,t+1}(x_i)\Big]\right\}+\widetilde{v}_{i,t+1}(x_i). \quad (22)$$

Comparing the dynamic programs in (21) and (22), using backwards induction over the time periods, we can show that $\widetilde{V}_t(\boldsymbol{x})=\sum_{i\in\mathcal{N}}\widetilde{v}_{it}(x_i)$ for all $\boldsymbol{x}\in\mathbb{Z}_+^n$ and $t\in\mathcal{T}$. Thus, it is enough to show that $\widetilde{v}_{it}(x_i)-\widetilde{v}_{it}(x_i-1)\le r_i$ for all $x_i\in\mathbb{Z}_+$ with $x_i\ge 1$ and $t\in\mathcal{T}$. We show the latter result by using induction over the time periods. We have $\widetilde{v}_{i,T+1}=0$ at time period $T+1$, so the result holds at time period $T+1$. Assuming that $\widetilde{v}_{i,t+1}(x_i)-\widetilde{v}_{i,t+1}(x_i-1)\le r_i$, we show that $\widetilde{v}_{it}(x_i)-\widetilde{v}_{it}(x_i-1)\le r_i$ as well. Letting $\psi_{it}=\sum_{j\in\mathcal{M}}\lambda_{jt}\sum_{S\subseteq\mathcal{N}}\frac{\widehat{w}_j(S)}{\tau_j}\phi_{ij}(S)$ for notational brevity, we write (22) equivalently as $\widetilde{v}_{it}(x_i)=\psi_{it}\,\mathbf{1}(x_i\ge 1)\,[r_i+\widetilde{v}_{i,t+1}(x_i-1)-\widetilde{v}_{i,t+1}(x_i)]+\widetilde{v}_{i,t+1}(x_i)$. Since $\widetilde{v}_{i,t+1}(x_i)-\widetilde{v}_{i,t+1}(x_i-1)\le r_i$

by the induction assumption, the last equality yields $\widetilde{v}_{it}(x_i) \geq \widetilde{v}_{i,t+1}(x_i)$ for all $x_i \in \mathbb{Z}_+$. In this case, using the equivalent expression for (22), we obtain the chain of inequalities

$$\begin{aligned}\widetilde{v}_{it}(x_i) - \widetilde{v}_{it}(x_i - 1) = &\ \psi_{it}\, \mathbf{1}(x_i \geq 1) \Big[r_i + \widetilde{v}_{i,t+1}(x_i - 1) + \widetilde{v}_{i,t+1}(x_i)\Big] + \widetilde{v}_{i,t+1}(x_i) - \widetilde{v}_{it}(x_i - 1) \\ \stackrel{(a)}{\leq} &\ \psi_{it}\, \mathbf{1}(x_i \geq 1)\, r_i + \Big[1 - \psi_{it}\, \mathbf{1}(x_i \geq 1)\Big] (\widetilde{v}_{i,t+1}(x_i) - \widetilde{v}_{i,t+1}(x_i - 1)) \stackrel{(b)}{\leq} r_i,\end{aligned}$$

where (a) holds by noting that $\widetilde{v}_{i,t+1}(x_i - 1) \leq \widetilde{v}_{it}(x_i - 1)$ and arranging the terms, whereas (b) holds because $\widetilde{v}_{i,t+1}(x_i) - \widetilde{v}_{i,t+1}(x_i - 1) \leq r_i$ by the induction assumption. ■

We give the proof of Theorem 7.1 using Lemmas F.1 and F.2.

Proof of Theorem 7.1:

Noting that $\tau_j = \sum_{t \in \mathcal{T}} \lambda_{jt}$ and using the definition of β_{it}, we have $\mathbb{E}\{Z_{i1}\} = \sum_{t \in \mathcal{T}} \beta_{it} = \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{M}} \lambda_{jt} \sum_{S \subseteq \mathcal{N}} \frac{\widehat{w}_j(S)}{\tau_j}\, \phi_{ij}(S) = \sum_{j \in \mathcal{M}} \sum_{S \subseteq \mathcal{M}} \phi_{ij}(S)\, \widehat{w}_j(S) \leq \widehat{c}_i$, where the last inequality holds because $\widehat{\boldsymbol{w}}$ is an optimal solution to problem (4) when we solve this problem with $\boldsymbol{c} = \widehat{\boldsymbol{c}}$. Similarly, we have $\sum_{i \in \mathcal{N}} \sum_{t \in \mathcal{T}} r_i\, \beta_{it} = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} \sum_{S \subseteq \mathcal{M}} r_i\, \phi_{ij}(S)\, \widehat{w}_j(S) = f(\widehat{\boldsymbol{c}})$, where the last equality, once again, uses the fact that $\widehat{\boldsymbol{w}}$ is an optimal solution to problem (4) when we solve this problem with $\boldsymbol{c} = \widehat{\boldsymbol{c}}$. In Lemma F.2, as discussed earlier in this section, we show that $\widetilde{V}_t(\boldsymbol{x}) \leq V_t(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{Z}_+^n$ and $t \in \mathcal{T}$. In Lemma G.1 in Appendix G, on the other hand, we show that if Z is a sum of independent Bernoulli random variables and $a \in \mathbb{Z}_+$ satisfies $a \geq \mathbb{E}\{Z\}$, then we have $\mathbb{E}\{[Z - a]^+\} \leq \min\{\frac{1}{2}, \frac{1}{\sqrt{a}}\}\, \mathbb{E}\{Z\}$. Using these observations together, we obtain

$$\begin{aligned}V_1(\widehat{\boldsymbol{c}}) \ \geq\ \widetilde{V}_1(\widehat{\boldsymbol{c}}) &\stackrel{(a)}{=} \sum_{i \in \mathcal{N}} r_i \left(\sum_{t \in \mathcal{T}} \beta_{it} - \mathbb{E}\Big\{[Z_{i1} - \widehat{c}_i]^+\Big\} \right) \stackrel{(b)}{=} \sum_{i \in \mathcal{N}} \sum_{t \in \mathcal{T}} r_i\, \beta_{it} \left(1 - \frac{\mathbb{E}\{[Z_{i1} - \widehat{c}_i]^+\}}{\mathbb{E}\{Z_{i1}\}} \right) \\ &\stackrel{(c)}{\geq} \sum_{i \in \mathcal{N}} \sum_{t \in \mathcal{T}} r_i\, \beta_{it} \left(1 - \min\left\{ \frac{1}{2}, \frac{1}{\sqrt{\widehat{c}_i}} \right\} \right) \ \geq\ \sum_{i \in \mathcal{N}} \sum_{t \in \mathcal{T}} r_i\, \beta_{it} \left(1 - \min\left\{ \frac{1}{2}, \frac{1}{\sqrt{\widehat{c}_{\min}}} \right\} \right)\end{aligned}$$

where (a) uses Lemma F.1, (b) holds because $\mathbb{E}\{Z_{i1}\} = \sum_{t \in \mathcal{T}} \beta_{it}$ and (c) uses Lemma G.1 and the fact that $\mathbb{E}\{Z_{i1}\} \leq \widehat{c}_i$. The result follows as $f(\widehat{\boldsymbol{c}}) = \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{N}} r_i\, \beta_{it}$ by the definition of β_{it}. ■

Appendix G: Tail Expectation of Sums of Bernoullis

In the next lemma, we give an upper bound on the expectation of the tail of a sum of Bernoulli random variables. We use this lemma in the proof of Theorem 7.1 in Appendix F.

Lemma G.1 (Tail Expectation of a Bernoulli Sum) *If Z is a sum of independent Bernoulli random variables and $a \in \mathbb{Z}_+$ satisfies $a \geq \mathbb{E}\{Z\}$, then we have*

$$\mathbb{E}\{[Z - a]^+\} \leq \min\left\{ \frac{1}{2}, \frac{1}{\sqrt{a}} \right\} \mathbb{E}\{Z\}.$$

The key part of the proof of the lemma above is showing that an analogous result holds for a binomial random variable. We show this result in the next lemma.

Lemma G.2 (Tail Expectation of a Binomial) *If X is a binomial random variable and $a \in \mathbb{Z}_+$ satisfies $a \geq \mathbb{E}\{X\}$, then we have*

$$\mathbb{E}\{[X-a]^+\} \;\leq\; \min\left\{\frac{1}{2}, \frac{1}{\sqrt{a}}\right\} \mathbb{E}\{X\}.$$

Proof: Let X be a binomial random variable with parameters (n,p). The result follows if $p=0$ or $p=1$, so we assume that $p \in (0,1)$. We have $\mathbb{E}\{X\} = np$ and $\mathrm{Var}(X) = np\,(1-p)$. First, we claim that $\mathbb{E}\{[X-a]^+\} \leq \frac{1}{2}\,\mathbb{E}\{X\}$. By a standard lemma, given as Lemma 1 in Gallego and Moon (1993), if $a \geq \mathbb{E}\{X\}$, then $\mathbb{E}\{[X-a]^+\} \leq \frac{1}{2}\sqrt{\mathrm{Var}(X)}$. Thus, we have $\mathbb{E}\{[X-a]^+\} \leq \frac{1}{2}\sqrt{np(1-p)} \leq \frac{1}{2}\sqrt{np}$. On the other hand, we have $\mathbb{E}\{X^2\} = \mathrm{Var}(X) + \mathbb{E}\{X\}^2 = np(1-p) + (np)^2 \leq np + (np)^2$. For any $x, b > 0$, it is simple to show that we have the inequality $[x-b]^+ \leq \frac{1}{4b}\,x^2$. In particular, setting $f(x) = \frac{1}{4b}\,x^2$ and $g(x) = [x-b]^+$, we have $f(0) = 0 = g(0)$, $f(2b) = b = g(2b)$ and $f(x) \geq g(x)$ for all $x \in \mathbb{R}$. Thus, we have $\mathbb{E}\{[X-a]^+\} \leq \frac{1}{4a}\,\mathbb{E}\{X^2\} \leq \frac{1}{4a}(np + (np)^2) \leq \frac{1}{4}\,(np + (np)^2)$, where the last inequality holds because $a \in \mathbb{Z}_+$ and $a \geq \mathbb{E}\{X\} > 0$, so $a \geq 1$. By the discussion so far in this paragraph, we have the chain of inequalities

$$\mathbb{E}\{[X-a]^+\} \;\leq\; \min\left\{\frac{1}{2}\sqrt{np}, \frac{1}{4}\,(np + (np)^2)\right\} \;=\; \frac{np}{2}\min\left\{\frac{1}{\sqrt{np}}, \frac{1+np}{2}\right\} \;\leq\; \frac{np}{2} \;=\; \frac{1}{2}\mathbb{E}\{X\},$$

where the last inequality uses the fact that $\min\{\frac{1}{\sqrt{x}}, \frac{1+x}{2}\} \leq 1$ for all $x \in \mathbb{R}_+$, because if $x \leq 1$, then $\frac{1+x}{2} \leq 1$, whereas if $x > 1$, then $\frac{1}{\sqrt{x}} \leq 1$. Thus, the claim follows.

The claim that we established in the previous paragraph follows from a somewhat standard argument. Second, we claim that $\mathbb{E}\{(X-a)^+\} \leq \frac{1}{\sqrt{a}}\mathbb{E}\{X\}$. The proof of this claim is novel and more difficult. If $a \in \{1,2,3,4\}$, then $\frac{1}{\sqrt{a}} \geq \frac{1}{2}$, so by claim in the previous paragraph, we immediately obtain $\mathbb{E}\{(X-a)^+\} \leq \frac{1}{2}\,\mathbb{E}\{X\} \leq \frac{1}{\sqrt{a}}\mathbb{E}\{X\}$. Furthermore, if $a = np$, then using the result cited in the previous paragraph, we immediately get $\mathbb{E}\{(X-a)^+\} \leq \frac{1}{2}\mathrm{Var}(X) = \frac{1}{2}\sqrt{np} \leq \sqrt{np} = \frac{1}{\sqrt{a}}\mathbb{E}\{X\}$. Lastly, since $X \leq n$ with probability 1, if $a > n$, then the claim trivially holds. Thus, In the rest of the proof, we proceed with the assumption that $a \geq 5$, $a > np$ and $a \leq n$. Let k be an integer such that $a \leq k \leq n$. Since $a > np$, we have $np < k \leq n$. For the conditional tail probability of the binomial random variable X, by Lemma 2.5 in Pelekis (2016), we have the bound

$$\frac{\mathbb{P}\{X \geq k+1\}}{\mathbb{P}\{X \geq k\}} \;\leq\; \frac{p\,(n-k)}{k\,(1-p)}.$$

Using the fact that $\frac{p\,(n-k)}{k\,(1-p)}$ is decreasing in k for $k \in [0,n]$, since $a \leq k$, we obtain $\frac{\mathbb{P}\{X \geq k+1\}}{\mathbb{P}\{X \geq k\}} \leq \frac{p\,(n-a)}{a\,(1-p)}$. Let $\beta = \frac{p\,(n-a)}{a\,(1-p)}$. Since $np < a \leq n$, we have $\beta < \frac{p\,(n-np)}{np(1-p)} = 1$.

By the discussion in the previous paragraph, if k is an integer such that $a \leq k \leq n$, then we have $\mathbb{P}\{X \geq k+1\} \leq \beta\,\mathbb{P}\{X \geq k\}$. Thus, starting with $\mathbb{P}\{X \geq a+1\} \leq \beta\,\mathbb{P}\{X \geq a\}$ and using the last inequality recursively, we obtain $\mathbb{P}\{X \geq a+\ell\} \leq \beta^\ell\, \mathbb{P}\{X \geq a\}$ for all $\ell = 1,\ldots,n-a$. In this case, computing the expectation through complementary cumulative distribution, we get

$$\begin{aligned}\mathbb{E}\{[X-a]^+\} \;&=\; \sum_{\ell=1}^{n} \mathbb{P}\{[X-a]^+ \geq \ell\} \;=\; \sum_{\ell=1}^{n-a} \mathbb{P}\{X \geq a+\ell\} \\ &\leq\; \sum_{\ell=1}^{n-a} \beta^\ell\, \mathbb{P}\{X \geq a\} \;\leq\; \frac{\beta}{1-\beta}\, \mathbb{P}\{X \geq a\}, \qquad (23)\end{aligned}$$

where the last inequality uses the fact that $\sum_{\ell=1}^{\infty} \beta^\ell \leq \frac{\beta}{1-\beta}$ for $\beta < 1$. To complete the proof of the claim, we consider the two cases $\frac{\beta}{1-\beta} \leq \sqrt{a}$ and $\frac{\beta}{1-\beta} > \sqrt{a}$.

Case 1: Assume that $\frac{\beta}{1-\beta} \leq \sqrt{a}$. By Markov inequality, $\mathbb{P}\{X \geq a\} \leq \frac{1}{a}\,\mathbb{E}\{X\}$, so using (23) along with the fact that $\frac{\beta}{1-\beta} \leq \sqrt{a}$, we get $\mathbb{E}\{[X-a]^+\} \leq \frac{\beta}{1-\beta}\,\frac{1}{a}\,\mathbb{E}\{X\} \leq \frac{1}{\sqrt{a}}\mathbb{E}\{X\}$.

Case 2: Assume that $\frac{\beta}{1-\beta} > \sqrt{a}$. Using the definition of β, we have $1-\beta = 1 - \frac{p\,(n-a)}{a\,(1-p)} = \frac{a-np}{a(1-p)}$, in which case, noting that $a \geq 5$ so $\sqrt{a} \geq 2$, as well as the fact that $\frac{\beta}{1-\beta} > \sqrt{a}$, we get

$$2 \leq \sqrt{a} \leq \frac{\beta}{1-\beta} = \frac{p\,(n-a)}{a-np} \leq \frac{a}{a-np},$$

where the equality is by the definition of β and the last inequality holds because $a > np$. Note that having $2 \leq \frac{a}{a-np}$ implies that $np \geq \frac{1}{2}a$. By Lemma 1 in Gallego and Moon (1993), if $a \geq \mathbb{E}\{X\}$, then $\mathbb{E}\{[X-a]^+\} \leq \frac{1}{2}\sqrt{\mathrm{Var}(X)}$, in which case, we obtain $\mathbb{E}\{[X-a]^+\} \leq \frac{1}{2}\sqrt{\mathrm{Var}(X)} = \frac{1}{2}\sqrt{np(1-p)} \leq \frac{1}{2}\sqrt{np} = \frac{np}{\sqrt{4np}} \leq \frac{np}{\sqrt{2a}} \leq \frac{1}{\sqrt{a}}\,\mathbb{E}\{X\}$, where second to last inequality uses the fact that $np \geq \frac{1}{2}a$. Thus, the claim holds under both cases that we considered. By the two claims established so far in the proof, we have the two inequalities $\mathbb{E}\{[X-a]^+\} \leq \frac{1}{2}\,\mathbb{E}\{X\}$ and $\mathbb{E}\{[X-a]^+\} \leq \frac{1}{\sqrt{a}}\,\mathbb{E}\{X\}$, in which case, it follows that $\mathbb{E}\{[X-a]^+\} \leq \min\{\frac{1}{2}, \frac{1}{\sqrt{a}}\}\,\mathbb{E}\{X\}$. ∎

To give a proof of Lemma G.1, we will use the following theorem, which is given as Theorem 28 in Pollard (2021). This theorem compares the tail probability of a sum of independent Bernoulli random variables with that of a binomial random variable with the same mean.

Theorem G.3 (Tail Comparison) *Letting $\{Y_i : i = 1,\ldots,n\}$ be independent Bernoulli random variables with $\mathbb{E}\{Y_i\} = p_i$, $Z = \sum_{i=1}^n Y_i$ and X be a binomial random variable with parameters $(n, \frac{1}{n}\sum_{i=1}^n p_i)$, for any $k \in \mathbb{Z}_+$ with $\sum_{i=1}^n p_i + 1 \leq k \leq n$, we have $\mathbb{P}\{Z \geq k\} \leq \mathbb{P}\{X \geq k\}$.*

Using the theorem above, we will be able to leverage Lemma G.2 to show Lemma G.1. Here is the proof of Lemma G.1.

Proof of Lemma G.1:

The random variable Z is a sum of independent Bernoullis, so let $Z = \sum_{i=1}^n Y_i$, where the random variables $\{Y_i : i = 1,\ldots,n\}$ are independent Bernoullis with $\mathbb{E}\{Y_i\} = p_i$. Furthermore, let X be a binomial random variable with parameters $(n, \frac{1}{n}\sum_{i=1}^n p_i)$. Note that $\mathbb{E}\{Z\} = \sum_{i=1}^n p_i = \mathbb{E}\{X\}$. Computing the expectation through the complementary cumulative distribution, we have

$$\begin{aligned}\mathbb{E}\{[Z-a]^+\} &= \sum_{\ell=1}^{n} \mathbb{P}\{[Z-a]^+ \geq \ell\} = \sum_{\ell=1}^{n-a} \mathbb{P}\{Z \geq a+\ell\} = \sum_{\ell=a+1}^{n} \mathbb{P}\{Z \geq \ell\} \\ &\overset{(a)}{\leq} \sum_{\ell=a+1}^{n} \mathbb{P}\{X \geq \ell\} \overset{(b)}{=} \mathbb{E}\{[X-a]^+\} \overset{(c)}{\leq} \min\left\{\frac{1}{2}, \frac{1}{\sqrt{a}}\right\} \mathbb{E}\{X\} \overset{(d)}{=} \min\left\{\frac{1}{2}, \frac{1}{\sqrt{a}}\right\} \mathbb{E}\{Z\},\end{aligned}$$

where (a) is by Theorem G.3 along with $a+1 \geq \mathbb{E}\{Z\}+1 = \sum_{i=1}^n p_i + 1$, (b) uses the same argument in the first three equalities above, (c) is by Lemma G.2 and (d) holds because $\mathbb{E}\{Z\} = \mathbb{E}\{X\}$. ■

Appendix H: Dynamic Programming Decomposition Benchmark

We explain the dynamic programming decomposition benchmark. Fixing some product $k \in \mathcal{N}$, we approximate the value function $J_t(\boldsymbol{x})$ in the dynamic program in (1) with a function of the form $\Theta_t(\boldsymbol{x}) = \nu_{kt}(x_k) + \sum_{i \in \mathcal{N}\setminus\{k\}} \widehat{\mu}_i\, x_i$. This value function approximation is nonlinear in x_k but linear in x_i for all $i \in \mathcal{N} \setminus \{k\}$. Therefore, the difference $\nu_{kt}(x_k+1) - \nu_{kt}(x_k)$ captures the marginal value of having one more unit of product k at time period t when we have x_k units of remaining inventory for product k, whereas the slope $\widehat{\mu}_i$ captures the marginal value of having one more unit of product i for each $i \in \mathcal{N} \setminus \{k\}$. The latter marginal value does not depend on the remaining inventory of the product. We use a linear programming approximation to estimate the marginal value of having one more unit of each product $i \in \mathcal{N} \setminus \{k\}$, but solve a dynamic program to carefully compute the marginal value of having one more unit of product k. In particular, viewing problem (7) as a linear programming approximation to the joint stocking and assortment customization problem, we solve this problem and choose $\widehat{\mu}_i$ as the optimal value of the dual variable associated with the first constraint in this problem. On the other hand, to compute $\nu_{kt}(x_k)$, we define the set $\mathcal{N}_k(x_k) = \mathcal{N}$ if $x_k > 0$ and $\mathcal{N}_k(x_k) = \mathcal{N} \setminus \{k\}$ if $x_k = 0$. Note that the set $\mathcal{N}_k(x_k)$ includes all products that we can offer when product k has x_k units of inventory and we ignore the availability of other products. In this case, replacing the value function $J_t(\boldsymbol{x})$ on both sides of (1) with the approximation $\nu_{kt}(x_k) + \sum_{i \in \mathcal{N}\setminus\{k\}} \widehat{\mu}_i\, x_i$, we solve the dynamic program

$$\nu_{kt}(x_k) = \sum_{j \in \mathcal{M}} \lambda_{jt} \max_{S \subseteq \mathcal{N}_k(x_k)} \left\{ \phi_{kj}(S) \Big[r_k + \nu_{k,t+1}(x_k - 1) - \nu_{k,t+1}(x_k) \Big] + \sum_{i \in \mathcal{N}\setminus\{k\}} \phi_{ij}(S)\,(r_i - \widehat{\mu}_i) \right\} + \nu_{k,t+1}(x_k),$$

The dynamic program above follows by replacing the value function $J_t(\boldsymbol{x})$ on both sides of (1) with the approximation $\nu_{kt}(x_k) + \sum_{i \in \mathcal{N}\setminus\{k\}} \widehat{\mu}_i\, x_i$ and arranging the terms.

We solve the dynamic program above for each product k and use $\nu_{kt}(x_k)$ to approximate the optimal total expected revenue over time periods $t, \ldots, T$ from x_k units of product k. In this case,

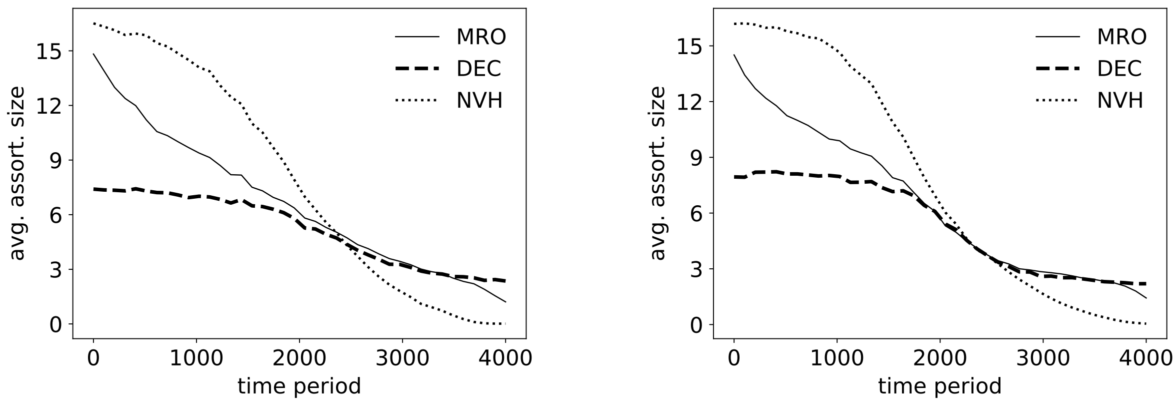

Figure 2 Average number of products in the assortments offered by the benchmarks.

for stocking decisions, we solve the problem $\max_{\boldsymbol{c}\in\mathbb{Z}_+^n}\left\{\sum_{i\in\mathcal{N}}\nu_{i1}(c_i)\,:\,\sum_{i\in\mathcal{N}}c_i\le K\right\}$. We can show that $\nu_{i1}(c_i)$ is concave in c_i, so we can solve this problem efficiently. For assortment customization decisions, we use the greedy policy with respect to the value function approximation $\sum_{i\in\mathcal{N}}\nu_{it}(x_i)$, so if we are at time period t with the remaining inventories $\boldsymbol{x}$, then we offer the assortment $\arg\max_{S\subseteq\mathcal{N}(\boldsymbol{x})}\sum_{i\in\mathcal{N}}\phi_{ij}(S)\,[r_i+\nu_{i,t+1}(x_i-1)-\nu_{i,t+1}(x_i)]$ to a customer of type j. This approach is called dynamic programming decomposition in the revenue management literature; see Section 3.4.4 in Talluri and van Ryzin (2005) and Section 4.2 in Zhang and Adelman (2009). It yields a strong heuristic for network revenue management problems, but has no performance guarantee.

Appendix I: Comparison of the Assortments Offered by Different Benchmarks

We investigate the number of products in the assortments offered by MRo, Dec and NVH as a function of time. In Figure 2, the two charts focus on two test problems with the parameter configurations $(4000, 0.3, 0.005, 0.5)$ and $(4000, 0.3, 0.01, 0.5)$. The solid, dashed and dotted data series, respectively, correspond to MRo, Dec and NVH. The horizontal axis shows the time period in the selling horizon, whereas the vertical axis shows the average number of products in the offered assortments, where the average is computed over the 1000 sample paths used to estimate the total expected revenues. As discussed in Section 8, we focus on MRo, Dec and NVH because MRo is the strongest benchmark from our approximation framework, whereas Dec and NVH are heuristics motivated by other work in the revenue management literature. We report results for two test problems, but our observations are consistent over our test problems.

Myopic assortments computed by NVH do not try to protect inventory for customers arriving later in the selling horizon, so NVH offers larger assortments early in the selling horizon, but as the inventory is depleted, this benchmark needs to focus on the products with remaining inventories and ends up offering smaller assortments later in the selling horizon. If we are at time period t and the remaining inventories are given by the vector $\boldsymbol{x}$, then NVH offers the assortment $\widetilde{S}_j\cap\mathcal{N}(\boldsymbol{x})$ to a customer of type j, where $\widetilde{S}_j$ is the myopic assortment for customer type j. As the inventories deplete over time, fewer components of the vector $\boldsymbol{x}$ remain positive and the set $\mathcal{N}(\boldsymbol{x})$ becomes smaller, so the assortments offered by NVH can only get smaller over time. To make up for this shortcoming NVH appears to start with larger assortments early in the selling horizon. Somewhat reverse trend holds for Dec. This benchmark is conservative, offering smaller assortments early in the selling horizon, but keeps its assortments on the large side later. In contrast, MRo appears to strike a balance between the two other benchmarks. We emphasize that while these observations consistently applied to our test problems, they are based on numerical results

Appendix J: Performance of the Benchmarks on Different Synthetic Datasets

In test problems, we sample the preference weights of the products from the uniform distribution over $[1, 10]$, customers with smaller consideration sets tend to arrive later in the selling horizon and we reorder the preference weights for half of the customer types so that the preference weights follow the reverse order of the revenues. In this section, we test our benchmarks when we deviate from this base experimental setup. Trying all combinations of possible deviations from the base experimental setup yields an exorbitant number of parameter configurations, so we focus on the test problems in the parameter configuration $(8000, 0.1, 0.005, 0.25)$ and change one aspect of our base experimental setup at a time. We give our computational results in Table 2. The table has three portions. In the top portion of the table, we vary the way we generate the preference weights of the products. In the four rows in the top portion, we sample the preference weights from the uniform distribution, respectively, over $[1, 10]$, $[1, 100]$, $[1, 1000]$ and $[1, 10000]$. All other aspects of our approach for generating our test problems remain unchanged. In the middle portion of the table, we vary the arrival order of the customers with smaller consideration sets. In the three rows in the middle portion, the customers with smaller consideration sets tend to arrive, respectively, later, earlier and uniformly in the selling horizon. In Section 8, to come up with test problems where the customers with smaller consideration sets tend to arrive later in the selling horizon, letting L_j be the length of the consideration set for customers of type j, we set the arrival probability for a customer of type j at time period t proportional to $\exp(-\gamma\, L_j\,(t - T/2))$ for some $\gamma > 0$. To come up with test problems where the customers with smaller consideration sets tend to arrive earlier and uniformly in the selling horizon, we use the same approach, respectively, with $\gamma < 0$ and $\gamma = 0$. In the bottom portion of the table, we vary the fraction of customer types with reordered preference weights. In the five rows in the bottom portion, we vary the fraction of customer types with reordered preference weights over $\{0, 0.25, 0.5, 0.75, 1\}$.

The first column in Table 2 gives the problem parameter we vary. The remaining twelve columns have the same interpretation as those in Table 1. In our experimental setup in Section 8, we sample

Varying the Interval for Preference Weights

Interval for Pref. Weights	Total Exp. Rev. GRA	GRO	MRA	MRO	DEC	NVH	Ranking of Total Exp. Rev. GRA	GRO	MRA	MRO	DEC	NVH
$[1,10]$	96.64	99.17	96.89	99.37	97.24	94.81	4.5	1.8	3.8	1.0	3.6	5.9
$[1,100]$	96.55	99.26	96.59	99.29	97.22	95.36	4.2	1.4	4.0	1.0	3.7	5.5
$[1,1000]$	96.14	98.83	96.30	98.95	96.30	93.43	4.1	1.5	3.6	1.0	3.9	5.9
$[1,10000]$	96.50	99.14	96.60	99.21	97.48	94.26	4.3	1.7	3.9	1.2	3.3	6.0

Varying the Arrival Order for Smaller Consideration Sets

Smaller Cons. Sets	Total Exp. Rev. GRA	GRO	MRA	MRO	DEC	NVH	Ranking of Total Exp. Rev. GRA	GRO	MRA	MRO	DEC	NVH
Later	96.64	99.17	96.89	99.37	97.24	94.81	4.5	1.8	3.8	1.0	3.6	5.9
Earlier	96.67	99.08	96.95	99.36	97.32	95.49	4.7	2.0	3.6	1.0	3.7	6.0
Uniform	96.64	99.26	96.97	99.52	98.11	95.53	4.9	2.0	3.9	1.0	3.2	6.0

Varying the Fraction of Customer Types with Reordered Preference Weights

Fraction of Reordered	Total Exp. Rev. GRA	GRO	MRA	MRO	DEC	NVH	Ranking of Total Exp. Rev. GRA	GRO	MRA	MRO	DEC	NVH
0.00	97.37	99.50	97.59	99.67	98.73	96.21	4.9	1.9	3.9	1.0	3.1	5.8
0.25	97.05	99.36	97.27	99.52	97.87	95.64	4.6	1.8	4.0	1.1	3.5	5.8
0.50	96.64	99.17	96.89	99.37	97.25	94.81	4.5	1.8	3.8	1.0	3.6	5.9
0.75	95.95	98.75	96.34	98.95	96.63	93.66	4.6	2.0	3.7	1.3	3.4	6.0
1.00	94.77	97.95	96.15	98.78	96.68	92.26	4.9	2.2	3.8	1.0	3.1	6.0

Table 2 Total expected revenues obtained by the benchmarks under deviations from the base experimental setup.

the preference weights of the products from the uniform distribution over $[1,10]$, customers with smaller consideration sets tend to arrive later in the selling horizon and we reorder the preference weights for half of the customer types. Thus, one row in each of the three portions of Table 2 matches the row corresponding to the parameter configuration $(8000, 0.1, 0.005, 0.25)$ in Table 1. In Table 2, MRO continues to emerge as the strongest benchmark and its strong performance is quite consistent. This benchmark is followed by GRO and DEC. The performance of NVH lags behind that of the remaining five benchmarks. Overall, our results indicate that the findings in Section 8 are robust to deviations from the base experimental setup.

Appendix K: Stocking Decisions Using a Continuous Relaxation

In Section 6, we describe an approach to make the stocking decisions by using a continuous relaxation of problem (3). We test the following two benchmarks motivated by this approach.

ROUNDED STOCKING AND RANDOMIZED CUSTOMIZATION (RRA): Letting $(\boldsymbol{c}^*, \boldsymbol{w}^*)$ be an optimal solution to problem (7), fixing the value of $\gamma = \lfloor (\frac{K}{n})^{2/3} \rfloor$, we set the stocking quantity of product i as $\widehat{c}_i = \lfloor (1 - \gamma \frac{n}{K}) c_i^* \rfloor + \gamma$. In this way, by the discussion at the end of Section 6, we execute Step 2 of our approximation framework with $\alpha = 1 - \sqrt[3]{\frac{n}{K}}$. By our choice of $\widehat{c}_i$, we have $\widehat{c}_{\min} = \min\{\widehat{c}_i : i \in \mathcal{N}\} \geq \gamma \geq \frac{1}{2} (\frac{K}{n})^{2/3}$, which implies that $1 - \frac{1}{\sqrt{\widehat{c}_{\min}}} \geq 1 - \sqrt{2} \sqrt[3]{\frac{n}{K}}$. For assortment customization, we use the policy in Section 7, so noting Theorem 7.1, we execute Step 3 of our approximation framework with $\beta = 1 - \sqrt{2} \sqrt[3]{\frac{n}{K}}$. In the proof of Theorem 6.1, we show that the stocking quantities $(\widehat{c}_i : i \in \mathcal{N})$ satisfy $\sum_{i \in \mathcal{N}} \widehat{c}_i \leq K$, but this inequality can be strict. If that happens to be the case, then we increase the stocking quantities until we reach the storage capacity K. In particular, we start with the stocking vector $\widehat{\boldsymbol{c}} = (\widehat{c}_i : i \in \mathcal{N})$ and increase one component of this vector at a time that provides the largest improvement in the value of the surrogate $f(\boldsymbol{c})$. In this way, we ensure that we do not waste unused storage capacity.

Rounded Stocking and Rollout Customization (RRo): We make the stocking decisions as in RRa and assortment customization decisions as in GRo. Similar to RRa, this benchmark executes Steps 2 and 3 of our approximation framework with $\alpha = 1 - \sqrt[3]{\frac{n}{K}}$ and $\beta = 1 - \sqrt{2}\,\sqrt[3]{\frac{n}{K}}$, so the solutions from RRa and RRo are asymptotically optimal as the storage capacity gets large.

We test the performance of RRa and RRo on the same set of test problems in Table 1. We give our results in Table 3. In this table, we report the performance of RRa and RRo, along with that of MRo, Dec and NVH. We report the performance of MRo, Dec and NVH because MRo is the strongest benchmark based on our approximation framework in Section 8, whereas Dec and NVH are the two benchmarks that do not use our approximation framework. In Table 3, the first column gives the parameter configurations for our test problems using the tuple $(T, P_0, \theta_{\min}, \eta)$. Recall that we generate 10 test problems for each parameter configuration. The second column shows the average storage capacity, where the average is computed over the 10 test problems in a parameter configuration. The last five columns give the average total expected revenues for the benchmarks expressed as a percentage of the upper bound on the optimal total expected revenue. The average total expected revenues for MRo, Dec and NVH correspond to those in Table 1.

Our results indicate that the benchmarks RRa and RRo are inferior to MRo, Dec and NVH. Both RRa and RRo execute Steps 2 and 3 of our approximation framework with $\alpha = 1 - \sqrt[3]{\frac{n}{K}}$ and $\beta = 1 - \sqrt{2}\,\sqrt[3]{\frac{n}{K}}$, so RRa and RRo are asymptotically optimal as the storage capacity gets large. For our test problems, the storage capacity gets larger as we have larger number of time periods in the selling horizon. For the test problems with $T = 4000, 8000$ and 16000, the average total expected revenues of RRa lag behind that of our strongest benchmark MRo, respectively, by 24.09%, 19.40% and 14.91%. The analogous gaps for RRo are, respectively, 18.47%, 15.38% and 12.15%. For the test problems with $T = 4000, 8000$ and 16000, the average storage capacities are, respectively, 985.89, 1972.28 and 3945.11. Thus, both RRa and RRo start closing the gap with MRo as the storage capacity gets large, which is expected due to their asymptotically optimal performance guarantees. We proceed to examining the performance of RRa and RRo under larger storage capacities.

In Table 4, we consider the values of the parameters $P_0 = 0.1$, $\theta_{\min} = 0.01$ and $\eta = 0.75$ for our test problems. Varying $T \in \{4000, 8000, 16000, 32000, 64000, 128000\}$, we obtain six parameter configurations. We generate one test problem for each parameter configuration. The first column in the table shows the value of the number of time periods in the selling horizon T, whereas the second

Params. $(T, P_0, \theta_{\min}, \eta)$	K	Total Exp. Rev. MRo	Dec	NVH	RRa	RRo
$(4000, 0.1, 0.005, 0.25)$	749.4	99.25	97.17	94.44	75.18	80.10
$(4000, 0.1, 0.005, 0.50)$	1499.5	97.72	93.96	91.95	78.12	82.01
$(4000, 0.1, 0.010, 0.25)$	748.5	99.30	97.56	94.37	75.05	80.07
$(4000, 0.1, 0.010, 0.50)$	1497.6	98.00	94.23	91.51	77.69	81.67
$(4000, 0.3, 0.005, 0.25)$	566.0	97.20	96.65	93.13	67.20	74.67
$(4000, 0.3, 0.005, 0.50)$	1132.1	95.81	92.74	90.49	76.08	81.41
$(4000, 0.3, 0.010, 0.25)$	564.5	97.30	97.06	92.77	67.02	74.63
$(4000, 0.3, 0.010, 0.50)$	1129.5	95.82	90.96	90.34	75.95	81.60
$(8000, 0.1, 0.005, 0.25)$	1499.5	99.37	97.24	94.81	77.52	81.14
$(8000, 0.1, 0.005, 0.50)$	2999.6	98.18	93.95	92.37	83.05	85.75
$(8000, 0.1, 0.010, 0.25)$	1497.6	99.40	97.71	94.76	77.15	80.78
$(8000, 0.1, 0.010, 0.50)$	2995.7	98.32	94.42	91.97	83.00	85.74
$(8000, 0.3, 0.005, 0.25)$	1132.1	98.11	97.01	93.63	74.42	79.80
$(8000, 0.3, 0.005, 0.50)$	2264.7	96.84	92.93	91.06	81.67	85.61
$(8000, 0.3, 0.010, 0.25)$	1129.5	98.21	97.59	93.35	74.30	79.75
$(8000, 0.3, 0.010, 0.50)$	2259.5	96.86	90.65	90.93	81.64	85.74
$(16000, 0.1, 0.005, 0.25)$	2999.6	99.41	97.50	95.09	82.69	85.08
$(16000, 0.1, 0.005, 0.50)$	5999.8	98.37	93.78	92.66	86.33	88.21
$(16000, 0.1, 0.010, 0.25)$	2995.7	99.41	97.78	95.04	82.71	85.06
$(16000, 0.1, 0.010, 0.50)$	5992.1	98.41	94.63	92.28	86.21	88.11
$(16000, 0.3, 0.005, 0.25)$	2264.7	98.65	97.17	94.01	80.53	84.22
$(16000, 0.3, 0.005, 0.50)$	4530.0	97.43	92.88	91.49	85.70	88.52
$(16000, 0.3, 0.010, 0.25)$	2259.5	98.70	97.90	93.76	80.44	84.19
$(16000, 0.3, 0.010, 0.50)$	4519.5	97.46	90.17	91.32	85.67	88.59
Avg.		98.06	95.15	92.81	78.97	83.02

Table 3 Total expected revenues obtained by the benchmarks for the synthetic datasets.

column shows the value of the corresponding storage capacity K. The next three columns show the total expected revenues obtained by the benchmarks RRa, RRo and NVH. Our results indicate that as the number of time periods in the selling horizon gets larger, so that the storage capacity gets larger as well, the performance of RRa and RRo gets better, which aligned with the fact that RRa and RRo both have performance guarantees of $1-(\sqrt{2}+1)\sqrt[3]{\frac{n}{K}}$. For the test problem with $K = 72751$, corresponding to $T = 128000$, the total expected revenues obtained by RRa and RRo are, respectively, 94.54% and 95.10% of the upper bound on the optimal total expected revenue. In all of the test problems, NVH never obtains more than 91.61% of the upper bound.

Appendix L: Maximizing the Total Expected Profit

Using our approximation framework, we can construct a heuristic to maximize the total expected profit with no constraint on the stocking quantities, where the total expected profit is given by the difference between the total expected revenue from the sales and the total cost of the stocked units. Letting σ_i be the unit procurement cost for product i, to maximize the total expected profit, we need to solve the problem $\max_{\boldsymbol{c}\in\mathbb{Z}_+^n}\big\{J_1(\boldsymbol{c})-\sum_{i\in\mathcal{N}}\sigma_i\,c_i\big\}$. To obtain a heuristic solution for this problem, as explained at the end of Section 2, using $\{B^1,\dots,B^q\}$ to denote a collection of possible values for the total inventory investment that we are willing to make, we try to solve the problem $\max_{\boldsymbol{c}\in\mathbb{Z}_+^n}\big\{J_1(\boldsymbol{c}):\sum_{i\in\mathcal{N}}\sigma_i\,c_i\le B^\ell\big\}$ for each $\ell=1,\dots,q$. In the last problem, we maximize

T	K	Total Exp. Rev. RRA	RRO	NVH
4000	2267	81.15	84.40	90.51
8000	4534	85.91	88.29	90.93
16000	9068	88.66	90.24	91.13
32000	18137	91.02	92.18	91.35
64000	36275	93.06	93.88	91.50
128000	72551	94.54	95.10	91.61
Avg.		89.06	90.68	91.17

Table 4 Total expected revenues obtained by the benchmarks for larger values of storage capacity.

the total expected revenue subject to the constraint that the total inventory investment is at most B^ℓ. Because we can incorporate knapsack constraints into our approximation framework, replacing the value function $J_1(\boldsymbol{c})$ with the approximate surrogate $f_{\mathsf{app}}(\boldsymbol{c})$ and recalling that $f_{\mathsf{app}}(\boldsymbol{c})$ is monotone and submodular in $\boldsymbol{c}$, we can obtain a $\frac{1}{2}(1-\frac{1}{e})$-approximate solution to the last problem; see Soma and Yoshida (2018). Letting $\widehat{\boldsymbol{c}}^\ell$ be such an approximate solution as a function of ℓ, we can use $\widehat{\boldsymbol{c}}^\ell$ as the approximate stocking quantities when the total inventory investment is at most B^ℓ. In this case, we use $\arg\max_{\ell=1,\ldots,q}\{f_{\mathsf{app}}(\widehat{\boldsymbol{c}}^\ell)-\sum_{i\in\mathcal{N}}\sigma_i\,\widehat{c}_i^\ell\}$ as a heuristic solution to the problem $\max_{\boldsymbol{c}\in\mathbb{Z}_+^n}\{J_1(\boldsymbol{c})-\sum_{i\in\mathcal{N}}\sigma_i\,c_i\}$. This approach requires trying out a reasonably large set of possible values $\{B^1,\ldots,B^q\}$ for the total inventory investment that we are willing to make and it does not have a performance guarantee, but we will experiment with this approach in this section to demonstrate that it performs quite well. We focus on the test problems in Section 8 under the parameter configuration $(8000, 0.1, 0.005, 0.25)$. To come up with the unit procurement cost of each product, we sample δ_i from the uniform distribution over $[\Delta, 0.9]$ and set the unit procurement cost of product i as $\sigma_i=\delta_i\,r_i$. In this way, the profit for product i is $r_i-\sigma_i=(1-\delta_i)\,r_i$, corresponding to a profit margin of $1-\delta_i$. We vary Δ vary over $\{0.1, 0.3, 0.5, 0.7\}$. For each value of Δ, we generate 10 test problems. We use the same approach discussed in Section 8 to generate all aspects of our test problems other than the unit procurement cost.

Considering the four benchmarks based on our approximation framework, which we referred to as GRA, GRO, MRA and MRO in Section 8, MRO consistently performed the best, so we report the results for MRO. When we modified the benchmark DEC to maximize the total expected profit, its performance was not satisfactory, so we do not report the results for DEC. To modify the benchmark NVH to maximize the total expected profit, we compute the myopic assortment that maximizes the expected profit from a customer of type j as $\widetilde{S}_j=\arg\max_{S\subseteq\mathcal{N}}\sum_{i\in\mathcal{N}}(r_i-\sigma_i)\,\phi_{ij}(S)$. We use the random variable X_i to capture the total demand for product i when we always offer the myopic assortment $\widetilde{S}_j$ to a customer of type j. In particular, we have $X_i=\sum_{i\in\mathcal{T}}X_{it}$, where $\{X_{it}:t\in\mathcal{T}\}$ are independent Bernoullis with X_{it} having parameter $\sum_{j\in\mathcal{M}}\lambda_{jt}\,\phi_{ij}(\widetilde{S}_j)$. To make the stocking decisions, we solve the problem $\max_{\boldsymbol{c}\in\mathbb{Z}_+^n}\{\sum_{i\in\mathcal{N}}r_i\,\mathbb{E}\{\min\{c_i,X_i\}\}-\sum_{i\in\mathcal{N}}\sigma_i\,c_i\}$. On the other hand, to make the assortment customization decisions, if we are at time period t with the remaining inventories given by the vector $\boldsymbol{x}$, then we offer the assortment $\widetilde{S}_j \cap \mathcal{N}(\boldsymbol{x})$ to a customer of type j. Lastly, to obtain an upper bound on the optimal total expected profit, we replace the objective function of problem (3) with $f(\boldsymbol{c}) - \sum_{i\in\mathcal{N}} \sigma_i\, c_i$, drop the constraint $\sum_{i\in\mathcal{N}} c_i \le K$ and solve the linear programming relaxation of this problem. This linear programming relaxation is equivalent to replacing the objective function of problem (7) with $\sum_{j\in\mathcal{M}} \sum_{S\subseteq\mathcal{N}} \sum_{i\in\mathcal{N}} r_i\, \phi_{ij}(S)\, w_j(S) - \sum_{i\in\mathcal{N}} \sigma_i\, c_i$ and dropping the constraint $\sum_{i\in\mathcal{N}} c_i \le K$ in this problem. We normalize the total expected profits obtained by our benchmarks by using the upper bound.

We give our computational results in Table 5. In this table, the first column gives the parameter configuration for each test problem by specifying the value of Δ. Recall that we generate 10 test problems for each value of Δ. The second and third columns give the average total expected profits obtained by each benchmark expressed as a percentage of the upper bound, where the average is computed over the 10 test problems. The last two columns give the average rankings of the total expected profits obtained by the benchmarks, where, once again, the average is computed over the 10 test problems. In 37 out of 40 test problems, MRo performs better than NVH. The average gap between the total expected profits obtained by the two benchmarks is 4.64%. On average MRo obtains the 89.43% of the upper bound on the optimal total expected profit, whereas NVH obtains 85.28% of the upper bound. Compared with the case where we maximize the total expected revenue subject to a constraint on the total number of stocked units, both benchmarks obtain a smaller fraction of the upper bound, but it is not possible to tell whether this observation holds because the upper bounds are looser or the solutions are poorer when we maximize the total expected profit, unless we compute the optimal stocking quantities, along with the optimal assortment customization policy. Nevertheless, MRo, which is based on our approximation framework, provides significant and consistent improvements over NVH.

Appendix M: Unobserved Customer Types and Value of Assortment Customization

Our model works with the understanding that we observe the type of each customer before choosing an assortment to offer. When the sales take place on an online platform and the type of a customer encodes information such as zip code, age, gender and purchase history, we can observe the type of a customer when the customer is logged into the platform. In this section, we give computational experiments to understand the benefit from observing the type of each customer and the loss in the total expected revenue when we do not always know the types of the customers. In the setting that we consider, we observe the type of only a fraction of the customers. For the remaining fraction of the customers, we choose an assortment to offer without knowing their types. Using p_0 to denote

	Total		Ranking of	
	Exp. Rev.		Total Exp. Rev.	
Δ	MRo	NVH	MRo	NVH
0.1	88.12	84.51	1.2	1.8
0.3	89.49	84.63	1.0	2.0
0.5	90.84	85.17	1.0	2.0
0.7	89.27	86.82	1.1	1.9
Avg.	89.43	85.28	1.1	1.9

Table 5 Computational results for the benchmarks for maximizing the total expected profit.

the fraction of the customers for which we observe the type, we can compute the optimal assortment customization policy using the dynamic program

$$\begin{aligned} J_t(\boldsymbol{x}) \;=\; & p_0 \sum_{j\in\mathcal{M}} \lambda_{jt} \max_{S\subseteq\mathcal{N}(\boldsymbol{x})} \left\{ \sum_{i\in\mathcal{N}} \phi_{ij}(S)\Big[r_i + J_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i)\Big] + \left(1-\sum_{i\in\mathcal{N}}\phi_{ij}(S)\right) J_{t+1}(\boldsymbol{x}) \right\} \\ & + (1-p_0) \max_{S\subseteq\mathcal{N}(\boldsymbol{x})} \left\{ \sum_{i\in\mathcal{N}} \left(\sum_{j\in\mathcal{M}} \lambda_{jt}\,\phi_{ij}(S)\right)\Big[r_i + J_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i)\Big] + \sum_{j\in\mathcal{M}} \lambda_{jt} \left(1-\sum_{i\in\mathcal{N}}\phi_{ij}(S)\right) J_{t+1}(\boldsymbol{x}) \right\} \\ & + \left(1-\sum_{j\in\mathcal{M}}\lambda_{jt}\right) J_{t+1}(\boldsymbol{x}), \end{aligned} \tag{24}$$

with the boundary condition that $J_{T+1}=0$. We interpret the dynamic program above by viewing p_0 as the probability that an arriving customer is logged into the platform. With probability p_0, the customer arriving at time period t will be logged in. In this case, a customer of type j arrives with probability λ_{jt}. After observing the type of the customer, we choose an assortment to offer. If we offer the assortment S, then we make a sale for product i with probability $\phi_{ij}(S)$. With probability $1-p_0$, on the other hand, the customer arriving at time period t will not be logged in. We choose an assortment to offer without observing the type of the arriving customer. Even though we do not observe the type of the arriving customer, a customer of type j will arrive with probability λ_{jt}. Thus, if we offer the assortment S, then we make a sale for product i with probability $\sum_{j\in\mathcal{M}}\lambda_{jt}\,\phi_{ij}(S)$. We can view the dynamic program in (24) as a version of the dynamic program in (1) with $m+1$ customer types. Indexing the customer types by $\{1,\ldots,m+1\}$, for $j=1,\ldots,m$, in the dynamic program in (24), a customer of type j arrives at time period t with probability $p_0\,\lambda_{jt}$. If a customer of type j arrives and we offer the assortment S to this customer, then she chooses product i with probability $\phi_{ij}(S)$. Letting $\Lambda_t=\sum_{j\in\mathcal{M}}\lambda_{jt}$ to capture the probability that a customer arrives at time period t, a customer of type $m+1$ arrives at time period t with probability $(1-p_0)\,\Lambda_t$. If a customer of type $m+1$ arrives and we offer the assortment S to this customer, then she chooses product i with probability $\frac{1}{\Lambda_t}\sum_{j\in\mathcal{M}}\lambda_{jt}\,\phi_{ij}(S)$. Under such partially observed customer types, to find the optimal stocking quantities, we can solve the problem $\max_{\boldsymbol{c}\in\mathbb{Z}_+^n}\big\{J_1(\boldsymbol{c}) : \sum_{i\in\mathcal{N}} c_i \le K\big\}$, where the value functions $\{J_t : t\in\mathcal{T}\}$ are given by the dynamic program in (24).

Viewing the dynamic program in (24) as a version of the dynamic program in (1) with $m+1$ customer types, we can use our approximation framework under partially observed customer types.

We continue using the linear programming-based surrogate in (4), but index the customer types by $\{1, \ldots, m+1\}$. For $j = 1, \ldots, m$, if we offer the assortment S to a customer of type j arriving at time period t, then this customer chooses product i with probability $\phi_{ij}(S)$. If we offer the assortment S to a customer of type $m+1$ arriving at time period t, then this customer chooses product i with probability $\frac{1}{\Lambda_t} \sum_{j \in \mathcal{M}} \lambda_{jt}\, \phi_{ij}(S)$. Even when the choice probability $\phi_{ij}(S)$ is governed by the multinomial logit model for all $j \in \mathcal{M}$, the mixed choice probability $\frac{1}{\Lambda_t} \sum_{j \in \mathcal{M}} \lambda_{jt}\, \phi_{ij}(S)$ is not necessarily governed by the multinomial logit model. Thus, we cannot use our results on making the stocking decisions under the multinomial logit model. In Step 2 of our approximation framework, to make the stocking decisions, we formulate problem (3) as an integer program. This integer program is equivalent to problem (7) with integrality constraints on the decision variables $\boldsymbol{c}$. Thus, we execute Step 2 of our approximation framework with $\alpha = 1$. We do not claim that we can solve the integer programming formulation of problem (3) in polynomial time but efficient algorithms under partially observed customer types is beyond the scope of our paper. In Step 3 of our approximation framework, our assortment customization policy does not assume that the customers choose under a specific choice model, so we continue using our assortment customization policy without any changes. Thus, we execute Step 3 of our approximation framework with $\beta = \frac{1}{2}$.

Experimental Setup: In Proposition A.7, we use a specific problem instance to demonstrate that if we do not customize the assortment offered to each customer, then we can incur a loss in the total expected revenue by a factor arbitrarily close to m. This problem instance characterizes the largest possible loss in the total expected revenue, because we know that if we do not customize the assortment offered to each customer, then we can incur a loss in the total expected revenue by at most a factor of m. In our computational experiments, we build on the specific problem instance in Proposition A.7. In this way, we hope to work with test problems for which the loss in the total expected revenue due to not being able to customize the assortments is large. In all of our test problems, we have $n = 10$ products and $m = 10$ customer types, so that we can solve the integer programming formulation of problem (3). The fraction of customers logged into the platform is p_0. Thus, we observe the type of only p_0 fraction of the customers and customize the assortments offered to them. We vary p_0. To generate the revenues of the products, we sample γ from the uniform distribution over $[1, 1.5]$ and set the revenue of product i as $r_i = \gamma^{i-1}$. Customers choose according to the multinomial logit model. For each customer type j, we sample κ_j from the uniform distribution over $[1, 1.5]$ and set the preference weight that a customer of type j associates with product i as $v_{ij} = (\kappa_j)^{2/3}$ for $i < j$, $v_{ij} = 1$ for $i = j$ and $v_{ij} = 0$ for $i > j$.

There are $T = 4000$ time periods. To come up with the probability that a customer of each type arrives into the system at each time period, we sample ξ from the uniform distribution over

$[1, 1.5]$ and set the probability that a customer of type j arrives into the system at time period t as $\lambda_{jt} = \xi^{1-j} / \sum_{k \in \mathcal{M}} \xi^{1-k}$. If we have $\gamma = \kappa_j = \xi$ for all $j \in \mathcal{M}$, then the revenues, preference weights and customer arrival probabilities in our test problems would mimic those in the specific problem instance in Proposition A.7. To generate the storage capacity, under the multinomial logit models that we generate, we use $\phi_{ij}(S)$ to denote the choice probability of product i out of assortment S by a customer of type j. We compute the myopic assortment that maximizes the expected revenue from a customer of type j as $\widetilde{S}_j = \arg\max_{S \subseteq \mathcal{N}} \sum_{i \in \mathcal{N}} r_{ij}\, \phi_{ij}(S)$. If we always offer the myopic assortments, then the total expected demand for all products over the selling horizon is $\mathsf{Demand} = \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} \lambda_{jt}\, \phi_{ij}(\widetilde{S}_j)$. We set the storage capacity as $K = \lceil 0.25 \times \mathsf{Demand} \rceil$, so the storage capacity is about 25% of the total expected demand obtained when we observe the types of all customers and offer the myopic assortments. We vary the fraction of the customers logged into the platform over $p_0 = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ to obtain six parameter configuration. For each parameter configuration, we generate 10 test problems. Having $p_0 = 0$ corresponds to the case where we do not observe the type of any customer, so we cannot customize the assortment offered to any customer, whereas having $p_0 = 1$ corresponds to the case where we observe the types of all customers, so we can customize the assortments offered to all customers.

Computational Results: We focus on the benchmarks MRo, Dec and NVH, because MRo is the strongest benchmark based on our approximation framework in Section 8, whereas Dec and NVH are the two benchmarks that do not use our approximation framework. Similar to our approach in Section 8, we obtain an upper bound on the optimal total expected revenue. In particular, we solve the linear programming relaxation of problem (3). This linear programming relaxation is equivalent to problem (7) with the understanding that we have $m+1$ customer types, where, as discussed earlier in this section, a customer of type $m+1$ chooses product i out of assortment S at time period t with probability $\frac{1}{\Lambda_t} \sum_{j \in \mathcal{M}} \lambda_{jt}\, \phi_{ij}(S)$. In Table 6, we compare the performance of the benchmarks. In this table, the first column gives the parameter configurations for our test problems by using the value of p_0. Recall that we generate 10 test problems for each value of p_0. The second, third and fourth columns give the average total expected revenues obtained by each benchmark expressed as a percentage of the upper bound, where the average is computed over the 10 test problems. The last three columns give the average rankings of the total expected revenues obtained by the benchmarks, where, once again, the average is computed over the 10 test problems. Overall, MRo continues to provide significant improvements over Dec and NVH. On average, MRo improves the total expected revenues of Dec and NVH, respectively, by 8.38% and 4.49%.

In Table 6, we compare the performance of the three benchmarks with each other when different fractions of the customers log into the platform, so that we can observe the type of different

	Total Exp. Rev.			Ranking of Total Exp. Rev.		
p_0	MRo	Dec	NVH	MRo	Dec	NVH
1.0	97.50	87.87	96.19	1.4	3.0	1.6
0.8	97.49	88.16	95.13	1.0	3.0	2.0
0.6	97.48	87.21	93.95	1.0	3.0	2.0
0.4	97.51	89.80	92.60	1.0	2.7	2.3
0.2	97.49	90.90	91.15	1.1	2.2	2.7
0.0	97.40	91.91	89.56	1.2	2.0	2.8
Avg.	97.48	89.31	93.10	1.1	2.7	2.2

Table 6 Total expected revenues obtained by the benchmarks under partially observed customer types.

fractions of the customers. Thus, the results in this table compare the performance of the different benchmarks for a fixed value of p_0, but do not give an indication of the loss in the total expected revenues due to the fact that we do not observe the types of all customers. In Table 7, we focus on the loss in the total expected revenues obtained by the three benchmarks due to the fact that we do not observe the types of all customers. In this table, the first column gives the value of p_0 for our test problems. The remaining columns give the average percent loss in the total expected revenue for each benchmark due to the fact that we do not observe the types of all customers, where the average is computed over the 10 test problems that we generate for each value of p_0. In particular, letting $\mathsf{Rev}^k(p_0)$ be the total expected revenue obtained by a benchmark for problem instance k when we observe the type for p_0 fraction of the customers, $\mathsf{Rev}^k(1)$ corresponds to the total expected revenue when we observe the types of all customers. Thus, for each benchmark, the average loss that we report in the table for a particular value of p_0 is given by $\frac{1}{10}\sum_{k=1}^{10} 100 \times \frac{\mathsf{Rev}^k(1)-\mathsf{Rev}^k(p_0)}{\mathsf{Rev}^k(1)}$.

Our results indicate that if we observe the type for 60% of the customers, then MRo loses 4.83% of its total expected revenue on average, when compared with the case where we observe the types of all customers. The corresponding losses for Dec and NVH are, respectively, 5.30% and 7.00%. Naturally, it is unavoidable to incur losses in the total expected revenue when the fraction of customers with unobserved types increases, but it is encouraging that the performance of the benchmarks degrades in a graceful fashion. Overall, NVH appears to be more sensitive to observing the types of the customers, as its performance degrades faster. When the fraction of customers with observed types is at or above 60%, the loss in the total expected revenues for MRo is slightly smaller than those of Dec, whereas when the fraction of customers with observed types is at or below 40%, the loss in the total expected revenues for Dec is slightly smaller than that of MRo. Nevertheless, our results in Table 6 indicate that even when we do not observe the type of any of the customers so that $p_0 = 0$, MRo tends to provide significantly larger total expected revenues than Dec.

Appendix N: Supermarket Purchase Datasets

We describe our approach for using the supermarket purchase dataset to generate our test problems, followed by our computational results.

	Loss in Total Exp. Rev.		
p_0	MRo	Dec	NVH
0.8	2.31	1.73	3.37
0.6	4.83	5.30	7.00
0.4	7.65	5.37	11.05
0.2	10.78	7.54	15.38
0.0	14.40	10.26	20.10

Table 7 Percent loss in total expected revenues for the benchmarks under partially observed customer types.

N.1 Experimental Setup

We use a Nielsen dataset on supermarket purchases; see Nielsen (2021). We have access to weekly purchases from four physical supermarkets in 135 product categories over one year. We focus on each product category separately. The supermarkets are located in different geographical locations, so they represent purchasing patterns by customers with different demographics. We treat the customers shopping from different supermarkets as different customer types. Our goal is to test the effectiveness of our approximation framework if we were to operate a central online platform to serve the customers shopping from the four different supermarkets with the opportunity to customize the assortment of products offered to each customer based the knowledge of her zip code. Proceeding with the understanding that each of the four supermarkets corresponds to a different customer type, we use $\mathcal{N}$ to denote the set of products in the product category that we focus on and $\mathcal{M}$ to denote the set of supermarkets. Letting $P_{ij}(\ell)$ be the number of purchases for product i from supermarket j in week ℓ, the dataset provides the information $\{P_{ij}(\ell) : i \in \mathcal{N},\ j \in \mathcal{M},\ \ell = 1, \ldots, 52\}$. We use the following approach to fit a multinomial logit model to characterize the choice behavior of the customers in different supermarkets.

We assume that the assortment available in a supermarket in a particular week consists of the products with at least one purchase in the week, so the assortment available in supermarket j in week ℓ is $S_j(\ell) = \{i \in \mathcal{N} : P_{ij}(\ell) > 0\}$. Total number of purchases in supermarket j in week ℓ is $T_j(\ell) = \sum_{i \in \mathcal{N}} P_{ij}(\ell)$. For each of the $T_j(\ell)$ purchases, we generate one transaction record, where a transaction record is characterized by the assortment of available products and the product purchased by one customer. In each of the $T_j(\ell)$ transaction records, the assortment of available products is $S_j(\ell)$. In $P_{ij}(\ell)$ of the $T_j(\ell)$ transaction records, the customer purchases product i. In this way, we generate transaction records compatible with the dataset. In the dataset, we do not know the customers leaving without a purchase. We assume that P_0 fraction of the customers leave without a purchase, so we generate $P_0 \times T_j(\ell)$ additional transaction records, in each of which, the assortment of available products is $S_j(\ell)$, but the customer leaves without a purchase. A similar approach is used by Berbeglia et al. (2022) as well. We vary P_0. Thus, for each supermarket j and week ℓ, we generate a total of $(1 + P_0)\, T_j(\ell)$ transaction records. In all of these transaction records, the assortment of available products is $S_j(\ell)$. In $P_0 \times T_j(\ell)$ of these transaction records, the customer leaves without a purchase, whereas in $P_{ij}(\ell)$ of these transaction records, the customer purchases product i. Using the transaction records from all weeks, we fit a multinomial logit model for each supermarket separately, characterizing the choice behavior of the customers shopping from different supermarkets. We shortly give the details of our fitting approach.

We have a total of $(1+P_0)\sum_{j\in\mathcal{M}} T_j(\ell)$ transaction records from all supermarkets in week ℓ, so the average number of customer arrivals per week is $\frac{1}{52}(1+P_0)\sum_{\ell=1}^{52}\sum_{j\in\mathcal{M}} T_j(\ell)$. We set the number of time periods in the selling horizon as $T=\lceil\frac{1}{52}(1+P_0)\sum_{\ell=1}^{52}\sum_{j\in\mathcal{M}} T_j(\ell)\rceil$. Thus, if there is one customer arrival at each time period and the selling horizon corresponds to a week, then the number of arrivals over the selling horizon closely reflects the number of customer arrivals per week in the dataset. Recalling that each supermarket corresponds to a different customer type, a customer of type j arrives into the system at time period t with probability $\lambda_{jt}=\sum_{\ell=1}^{52} T_j(\ell)/\sum_{\ell=1}^{52}\sum_{k\in\mathcal{M}} T_k(\ell)$, so the probability of observing a customer of type j at any time period is proportional to the share of the purchase records from supermarket j. The dataset provides the price charged for each product in each supermarket and in each week. Using $p_{ij}(\ell)$ to denote the price of product i in supermarket j in week ℓ whenever this product is available in the week, we set the revenue associated with product i as $r_i=\sum_{\ell=1}^{52}\sum_{j\in\mathcal{M}} \mathbf{1}(i\in S_j(\ell))\,p_{ij}(\ell)/\sum_{\ell=1}^{52}\mathbf{1}(i\in S_j(\ell))$, which is the average price of product i over all supermarkets and weeks, after focusing attention only to the supermarkets and weeks in which the product was in the assortment of available products.

To generate the storage capacity, we use $\phi_{ij}(S)$ to denote the choice probability of product i out of assortment S by a customer of type j under the multinomial logit model that we fit. The myopic assortment that maximizes the expected revenue from a customer of type j is $\widetilde{S}_j=\arg\max_{S\subseteq\mathcal{N}}\sum_{i\in\mathcal{N}} r_i\,\phi_{ij}(S)$. Thus, as in our computational experiments with synthetic datasets, if we always offer the myopic assortment to all customer types, then the total expected demand for all products is $\mathsf{Demand}=\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{M}}\lambda_{jt}\,\phi_{ij}(\widetilde{S}_j)$. We set the storage capacity as $K=\lceil\eta\,\mathsf{Demand}\rceil$. We vary η. Lastly, we are interested in product categories where the choice behavior in different supermarkets is sufficiently different so that assortment customization is expected to make an impact. Letting $\boldsymbol{v}_j=(v_{ij}: i\in\mathcal{N})$ be the preference weights in the multinomial logit model fitted for supermarket j, we focus on product categories where the maximum correlation coefficient between any two vectors $\boldsymbol{v}_j$ and $\boldsymbol{v}_k$ for $j,k\in\mathcal{M}$ and $j\neq k$ is at most 0.8 and all coefficients of correlation between any other pair is at most 0.5. In this way, we end up with four product categories, which are cookies and brownies, shredded cheese, fresh potatoes and stout beer. The numbers of products in these product categories are 166, 30, 28 and 37.

Varying $P_0\in\{0.1,0.3\}$ and $\eta\in\{0.25,0.5\}$, using $\{1,2,3,4\}$ to denote the four product categories, we have 16 parameter configurations for our test problems.

N.2 Fitting a Multinomial Logit Model to the Supermarket Dataset

We give the details of our approach for fitting a multinomial logit model to the transaction records from each supermarket. By the discussion earlier in this section, we have $(1+P_0)\sum_{\ell=1}^{52} T_j(\ell)$ transaction records for supermarket j. We focus on a fixed supermarket. Letting K_j be the number of transaction records for supermarket j, we use $\{(S_j(q), i_j(q)) : q = 1, \ldots, K_j\}$ to denote these transaction records, where $S_j(q)$ is the assortment of available products and $i_j(q)$ is the product purchased, if any, in transaction record q. If the customer left without a purchase in transaction record q, then we have $i_j(q) = \varnothing$. The dataset provides the price for each product in each supermarket in each week and this price may change from one week to another. Each transaction record happens in a particular week. We use $p_{ij}(q)$ to denote the price of product i in supermarket j during the week that corresponds to transaction record q, which is provided in the dataset. In the multinomial logit model, we postulate that if the price for product i in supermarket j is p, then the preference weight that a customer shopping in supermarket j associates with product i is of the form $v_{ij}(p) = \exp(\gamma_{ij} - \beta_j\, p)$, where the parameters $\boldsymbol{\gamma}_j = (\gamma_{ij} : i \in \mathcal{N})$ and β_j for supermarket j need to be estimated from the dataset. The parameter γ_{ij} characterizes the inherent attractiveness of product i to the customers shopping in supermarket j, whereas β_j is the price sensitivity of the customers shopping in supermarket j. Thus, using the transaction records $\{(S_j(q), i_j(q)) : q = 1, \ldots, K_j\}$, the log-likelihood function for supermarket j is given by

$$\begin{aligned} L_j(\boldsymbol{\gamma}_j, \beta_j) \;=\; & \sum_{q=1}^{K_j} \sum_{i \in \mathcal{N}} \mathbf{1}(i_j(q) = i) \log\left(\frac{v_{ij}(p_{ij}(q))}{1 + \sum_{k \in S_j(q)} v_{kj}(p_{kj}(q))} \right) \\ & + \sum_{q=1}^{K_j} \mathbf{1}(i_j(q) = \varnothing) \log\left(\frac{1}{1 + \sum_{k \in S_j(q)} v_{kj}(p_{kj}(q))} \right), \end{aligned}$$

where $\{(S_j(q), i_j(q)) : q = 1, \ldots, K_j\}$ and $\{p_{ij}(q) : i \in S_j(q),\ q = 1, \ldots, K_j\}$ are provided by the dataset. We maximize $L_j(\boldsymbol{\gamma}_j, \beta_j)$ above over $(\boldsymbol{\gamma}_j, \beta_j)$ to obtain the estimates $(\widehat{\boldsymbol{\gamma}}_j, \widehat{\beta}_j)$.

Recalling that the revenue of product i is given by r_i, a customer of type j associates the preference weight $v_{ij} = \exp(\widehat{\gamma}_{ij} - \widehat{\beta}_j\, r_i)$ with product i in our computational experiments.

N.3 Computational Results

We continue using the same six benchmarks that we used in Section 8. We give our computational results in Table 8. The first column gives the parameter configuration for each test problem by using the tuple (C, P_0, η), where C stands for the product category. We have one test problem for each parameter configuration. The second and third columns, respectively, give the values of the number of time periods in the selling horizon T and the storage capacity K. The last twelve columns have the same interpretation as those in Table 1, but we do not compute averages over multiple

Params. (C, P_0, η)	T	K	Total Exp. Rev. GRA	GRO	MRA	MRO	DEC	NVH	Ranking of Total Exp. Rev. GRA	GRO	MRA	MRO	DEC	NVH
$(1, 0.1, 0.25)$	4532	775	93.82	97.60	95.35	97.78	96.42	89.71	5	2	4	1	3	6
$(1, 0.1, 0.50)$	4532	1551	88.22	91.43	94.27	94.22	94.15	80.99	5	4	1	2	3	6
$(1, 0.3, 0.25)$	5827	986	78.45	84.59	91.33	93.16	88.16	72.13	5	4	2	1	3	6
$(1, 0.3, 0.50)$	5827	1973	85.32	89.39	93.79	93.80	92.00	79.16	5	4	2	1	3	6
$(2, 0.1, 0.25)$	455	94	95.39	99.79	94.84	99.42	99.87	94.47	4	2	5	3	1	6
$(2, 0.1, 0.50)$	455	188	93.20	97.64	94.85	97.25	93.54	88.97	5	1	3	2	4	6
$(2, 0.3, 0.25)$	585	101	88.85	95.82	91.92	97.22	97.41	87.78	5	3	4	2	1	6
$(2, 0.3, 0.50)$	585	202	88.99	92.96	89.37	92.39	89.43	84.81	5	1	4	2	3	6
$(3, 0.1, 0.25)$	3439	661	95.68	97.69	97.21	98.58	89.83	88.26	4	2	3	1	5	6
$(3, 0.1, 0.50)$	3439	1323	95.18	96.81	95.93	96.91	82.80	93.19	4	2	3	1	6	5
$(3, 0.3, 0.25)$	4421	701	93.17	96.25	95.47	97.45	95.98	92.00	5	2	4	1	3	6
$(3, 0.3, 0.50)$	4421	1402	95.22	96.27	96.01	96.94	81.01	93.68	4	2	3	1	6	5
$(4, 0.1, 0.25)$	509	102	94.10	98.64	94.10	98.64	98.89	96.44	5	2	5	2	1	4
$(4, 0.1, 0.50)$	509	204	91.39	94.87	94.71	96.46	78.15	96.59	5	3	4	2	6	1
$(4, 0.3, 0.25)$	654	109	90.57	96.21	92.76	96.39	92.62	88.56	5	2	3	1	4	6
$(4, 0.3, 0.50)$	654	219	88.86	92.35	92.03	94.02	82.55	86.80	4	2	3	1	6	5
Avg.			91.02	94.89	94.00	96.29	90.80	88.35	4.7	2.4	3.3	1.5	3.6	5.4

Table 8 Total expected revenues obtained by the benchmarks for the supermarket purchase datasets.

test problems because we have one test problem in each parameter configuration. Our results are aligned with those for the synthetic datasets. Overall, the strongest benchmark is MRo. Working with multiple surrogates when making the stocking decisions and using rollout when making the assortment customization decisions pay off and MRo generally performs better. The performance of Dec lags behind that of MRo. For one test problem NVH is the strongest benchmark, but NVH generally falls behind the other benchmarks.

Appendix O: Misidentified Customer Types

We give a brief computational study to check the performance of our approximation framework when we misidentify the customer types. We consider the case where we know the composition of the market in the sense that we have access to the arrival probabilities $\{\lambda_{jt} : j \in \mathcal{M},\ t \in \mathcal{T}\}$ for the customers of each type at each time period, as well as the choice probabilities $\{\phi_{ij}(S) : i \in \mathcal{N},\ j \in \mathcal{M},\ S \subseteq \mathcal{N}\}$ for each product by each customer type within each assortment. However, when a customer arrives into the system and we need to make an assortment offer decision to this customer, we may misidentify the type of the customer and customize the assortment offered to the customer based on the wrong customer type. In particular, we use ρ to denote the probability that we misidentify the type of a customer. Because we have access to the arrival probabilities $\{\lambda_{jt} : j \in \mathcal{M},\ t \in \mathcal{T}\}$ and choice probabilities $\{\phi_{ij}(S) : i \in \mathcal{N},\ j \in \mathcal{M},\ S \subseteq \mathcal{N}\}$, we can find the stocking quantities and compute an assortment customization policy precisely as described for each benchmark in Section 8.2. When we simulate the decisions of the assortment customization policy, a customer of type j arrives into the system at time period t with probability λ_{jt}. With probability

Misid. Prob. ρ	Total Exp. Rev. GRA	NVH	Perc. Gap
0.0	98.39	96.21	2.21
0.1	95.65	93.96	1.77
0.2	92.12	90.91	1.31
0.3	88.37	87.41	1.09
0.4	84.56	83.76	0.95
0.5	80.73	80.01	0.90

Table 9 Total expected revenues obtained by the benchmarks under misidentified customer types.

$\rho \frac{\lambda_{kt}}{\sum_{\ell \in \mathcal{M}} \lambda_{\ell t}}$, we believe that this customer is of type $k \neq j$ and make an assortment customization decision for this customer assuming that she is of type k. With probability $(1-\rho) + \rho \frac{\lambda_{jt}}{\sum_{\ell \in \mathcal{M}} \lambda_{\ell t}}$, we believe that this customer is of type j and make an assortment customization decision for this customer assuming that she is of type j. Note that $\frac{\lambda_{kt}}{\sum_{\ell \in \mathcal{M}} \lambda_{\ell t}}$ is the probability that a customer arriving at time period t is of type k conditional on the fact that there is a customer arrival at this time period. Therefore, if we misidentify the type of a customer, then we sample her misidentified type according to the composition of the market. We vary the misidentification probability ρ. We generate a test problem by using an approach similar to the one in Appendix M. For this test problem, we vary $\rho \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$ allowing as much as 50% of customers with misidentified types. For economy of space. we report the results for the benchmarks GRA and NVH. Under the assumption that no customer types are misidentified, we use the same approach in Section 8.3 to compute an upper bound on the optimal total expected revenue. We normalize the total expected revenues by using this upper bound on the optimal total expected revenue. Note that when the misidentification probability gets larger, this upper bound gets looser.

We give our results in Table 9. The first column gives the value of the misidentification probability. The second and third columns give the total expected revenues obtained by GRA and NVH. The fourth column gives the percent gap between the total expected revenues of the two benchmarks. Our results indicate that our approximation framework maintains its edge even for fairly large misidentification probabilities. When we cannot identify the type of 30% of the customers, our approximation framework yields more than 1% improvement in the total expected revenue, which is considered significant. As the misidentification probability increases, the benefit from a policy that carefully makes the assortment customization decisions diminishes and the gaps between the performance of the two benchmarks decrease. For large misidentification probabilities, which makes it almost pointless to try to use an assortment customization policy, the performance of the two benchmarks is, at minimum, comparable. The results point out that our approximation framework is relatively robust to misidentifying the types of the customers, but we emphasize that one should make such robustness checks over a larger set of problem parameters and our approximation framework is not designed to explicitly handle misidentified customer types.