Coordinated Inventory Stocking and Assortment Personalization

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

December 7, 2022

We study a joint inventory stocking and assortment personalization 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 different product preferences arrive into the system over the selling horizon. Depending on the remaining product inventories and the type of the customer, we offer a personalized product assortment to the arriving customer. The customer makes a choice within the assortment according to a choice model. Our goal is to choose the stocking quantities at the beginning of the selling horizon and to find a policy to offer a personalized assortment to each customer 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 fresh groceries, which require operating out of an urban warehouse to be close to customers, but allow the flexibility to personalize the assortment for each customer. Finding a good assortment personalization 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 personalization 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. To our knowledge, these are the first guarantees for our problem class. 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 personalize the product assortment offered to each customer is an important source of flexibility for online retailers. From the customer viewpoint, personalizing the assortment for each customer may align the products viewed by the customer with her preferences, allowing the customer to have a more satisfactory shopping experience. From the firm viewpoint, personalizing 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 an assortment personalization decision for a customer 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 personalization decisions should depend on the current inventories of the products, the stocking decisions should anticipate how the personalized assortments will deplete the inventory, thereby creating a natural interaction between inventory stocking and assortment personalization. The challenge of coordinating assortment personalization 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 tightly 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 customer, such as geographical location, age, gender and purchase history. Using this information, along with the remaining inventories of the products, they have the ability to offer a personalized assortment to each customer. Even if the platform does not attempt to use the information about an arriving customer to personalize 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 personalization 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 different preferences for the products arrive over the selling horizon. Type of a customer may encode her geographical location, age, gender and purchase history. Depending on the remaining inventories and type of the customer, we offer a personalized product assortment to each customer. The customer makes a choice within the assortment according to a choice model. Our goal is to choose the stocking quantities at the beginning of the selling horizon and to find a policy to offer a personalized assortment to each customer so that we maximize the total expected revenue over the selling horizon. Our work is motivated by online retailers making same-day delivery promises or selling fresh groceries, as both operate out of a local warehouse to be close to customers. Computing the optimal assortment personalization policy requires solving a high-dimensional dynamic program with a state variable keeping track of the remaining inventories. The initial value function of the dynamic program characterizes 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 personalization 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 personalization 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 personalization 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 personalization decisions. Letting $\widehat{\boldsymbol{c}}$ be our stocking quantities, we construct an assortment personalization 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 personalization problem, we show that using the stocking quantities computed in Step 2 and subsequently following the assortment personalization policy constructed in Step 3 yields a total expected revenue of at least $\alpha \beta \, \mathsf{opt}$ (Theorem 2.1). Thus, we get an $\alpha\beta$-approximate solution.

Performance Guarantees. To put our framework into action, we need to construct the surrogate in Step 1, choose the stocking quantities in Step 2 by approximately solving a problem that involves the surrogate in the objective function and construct an assortment personalization policy in Step 3 such that the total expected revenue of the policy is lower bounded by a certain fraction of the surrogate. We construct our surrogate by using a linear program to approximate the optimal total expected revenue from the assortment personalization decisions. This linear program is the so-called choice-based deterministic linear program in the revenue management literature. The optimal objective value of this linear program at fixed stocking quantities is an upper bound on the optimal total expected revenue from the assortment personalization decisions starting from these stocking quantities, so our linear programming-based surrogate satisfies the requirements in Step 1. Throughout the paper, the surrogate $f : \mathbb{Z}_+^n \to \mathbb{R}_+$ will always be our linear programming-based surrogate. It is, however, still not clear how to choose the stocking quantities by approximately solving the problem in Step 2 with the linear programming-based surrogate and how to construct an assortment personalization policy in Step 3 such that the total expected revenue of the policy is lower bounded by a certain fraction of the linear programming-based surrogate.

It turns out that we can execute Steps 2 and 3 of our approximation framework efficiently under our linear programming-based surrogate. In this way, using our approximation framework, if the customers choose under a multinomial logit model, then we get a $\frac{1}{4}(1-\frac{1}{e})$-approximate solution to the joint stocking and assortment personalization 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 (Theorem 3.1). To our knowledge, these are the first guarantees for our problem. 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 gets large, because even if the storage capacity in our problem is large, it is not clear that the stocking quantity of each product is large. 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. Our performance guarantees follow by designing efficient algorithms to execute Steps 2 and 3 of our approximation framework, as we explain next.

Inventory Stocking Decisions. In Step 2 of our approximation framework, we choose the stocking quantities by solving the problem $\max_{\boldsymbol{c}\in\mathbb{Z}_+^n}\{f(\boldsymbol{c}):\sum_{i=1}^n c_i\le K\}$ with the linear programming-based surrogate. We show that this problem is APX-hard even when the customers choose according to the multinomial logit model (Theorem 4.1). When the customers choose under the multinomial logit model, we construct an approximation $f_{\mathsf{app}}:\mathbb{Z}_+^n\to\mathbb{R}_+$ to the linear programming-based surrogate such that $\frac{1}{2}f(\boldsymbol{c})\le f_{\mathsf{app}}(\boldsymbol{c})\le f(\boldsymbol{c})$ for all $\boldsymbol{c}\in\mathbb{Z}_+^n$. We show that this approximate surrogate is monotone and submodular over the integer lattice (Theorem 4.3). Thus, the problem $\max_{\boldsymbol{c}\in\mathbb{Z}_+^n}\{f_{\mathsf{app}}(\boldsymbol{c}):\sum_{i=1}^n c_i\le K\}$ maximizes a monotone and submodular function with a cardinality constraint, which admits a $(1-\frac{1}{e})$-approximation; see Soma and Yoshida (2018). Collecting these results, we get a $\frac{1}{2}(1-\frac{1}{e})$-approximate solution to the problem $\max_{\boldsymbol{c}\in\mathbb{Z}_+^n}\{f(\boldsymbol{c}):\sum_{i=1}^n c_i\le K\}$ with the linear programming-based surrogate and under the multinomial logit model.

When the customers choose under a general choice model, we solve the continuous relaxation of the problem $\max_{\boldsymbol{c}\in\mathbb{Z}_+^n}\{f(\boldsymbol{c}):\sum_{i=1}^n c_i\le K\}$. In this case, we show that we can round the solution to the continuous relaxation to obtain a $(1-\sqrt[3]{\frac{n}{K}})$-approximate solution such that the stocking quantity of each product is lower bounded by $\frac{1}{2}\left(\frac{K}{n}\right)^{2/3}$ (Theorem 5.1). The lower bound on the stocking quantity of each product becomes useful when constructing a good assortment personalization policy. Thus, if the customers choose under the multinomial logit model, then we can execute Step 2 of our approximation framework with $\alpha=\frac{1}{2}(1-\frac{1}{e})$, whereas if the customers choose under a general choice model, then we can execute the same step with $\alpha=1-\sqrt[3]{\frac{n}{K}}$. Our linear programming-based surrogate corresponds to the choice-based deterministic linear program, so our results also show how to obtain approximate solutions to the choice-based deterministic linear program when the initial resource quantities are integer-valued decision variables.

Assortment Personalization Decisions. 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 show that we can efficiently construct an assortment personalization policy with a total expected revenue of at least $\max\{\frac{1}{2}, 1-\frac{1}{\sqrt{\widehat{c}_{\min}}}\}\, f(\widehat{\boldsymbol{c}})$ (Theorem 6.1). Thus, we can execute Step 3 of our approximation framework with $\beta = \max\{\frac{1}{2}, 1-\frac{1}{\sqrt{\widehat{c}_{\min}}}\}$ under the linear programming-based surrogate. When all stocking quantities are lower bounded by $\frac{1}{2}\,(\frac{K}{N})^{2/3}$, so that $\widehat{c}_{\min} \geq \frac{1}{2}\,(\frac{K}{n})^{2/3}$, the last performance guarantee becomes $\max\{\frac{1}{2}, 1-\sqrt{2}\,\sqrt[3]{\frac{n}{K}}\}\, f(\widehat{\boldsymbol{c}})$. Our assortment personalization 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 into the system, we sample an assortment from the distribution corresponding to the customer type, drop the products without remaining inventories from the sampled assortment and offer the remaining products. There is existing work that constructs policies for similar assortment personalization problems. Existing work assumes that the stocking quantities are fixed. Even under fixed stocking quantities, the policies in the existing work have some form of a preprocessing step, damaging the intuitive nature of the policy. Our assortment personalization policy directly follows the distribution that we obtain from a linear program. 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 personalization decisions are not our sole focus, we make a useful contribution in that domain.

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 personalization problem. This result is the first to give a constant-factor approximate solution for a joint stocking and assortment personalization 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. In our linear programming-based surrogate, $f(\boldsymbol{c})$ corresponds to the optimal objective value of the choice-based deterministic linear program when viewed as a function of the stocking quantities. Our approximate surrogate $f_{\mathsf{app}}(\boldsymbol{c})$ is a half-approximation to $f(\boldsymbol{c})$ and it is monotone and submodular in $\boldsymbol{c}$. Such an approximation is valuable for solving the choice-based deterministic linear program when the initial resource quantities are also decision variables with cardinality, knapsack or matroid constraints. Lastly, even if we put aside coordinating the stocking and assortment personalization decisions and only focus on assortment personalization, our assortment personalization policy is extremely simple and its performance guarantee matches the best ones available in the literature.

Computational Experiments and Practical Refinements. We give computational experiments on 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 a newsvendor heuristic. On average, we obtain solutions within 7.16% of the upper bound and improve the performance of the newsvendor heuristic by 2.66%. Furthermore, we give refinements on our approximation framework to improve its practical performance, while maintaining the same theoretical performance guarantee. In particular, our approximate surrogate is based on fixing certain decision variables when computing our linear programming-based surrogate. To refine our approximate surrogate, we try multiple values to fix the decision variables, whereas to refine our assortment personalization policy, we use rollout. Our refinements bring our solutions to within 3.26% of the upper bound on the optimal total expected revenue, resulting in solutions with an average optimality gap of only 3.26%.

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. This problem setup is referred to as stockout-based substitution. 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, as well as 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, possibility 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 personalization decisions. This linear program is known as the choice-based deterministic 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 assortment that we can possibly offer to each customer type, so it is customary to solve the the linear program through column generation. The column generation subproblem can efficiently be solved or approximated under a variety of choice models, including multinomial logit, nested logit, paired combinatorial logit, non-parameteric 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 deterministic linear program directly without resorting to column generation; see Gallego et al. (2015), Feldman and Topaloglu (2017), Cao et al. (2021) and Cao et al. (2022). Under these choice models, we can reformulate the choice-based deterministic linear program by using the expected sales of each product as the decision variable.

There is work on assortment personalization 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. All of these papers require a non-trivial preprocessing step that adjusts the revenues of the products, builds value function approximations or drops certain products from consideration. Our assortment personalization policy is $\max\{\frac{1}{2}, 1-\frac{1}{\sqrt{\widehat{c}_{\min}}}\}$-approximate and has no preprocessing step, so we match the best available performance guarantees with an extremely simple policy.

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

2. Problem Formulation and Approximation Framework

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 index 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 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 personalized assortment to offer to each arriving customer as a function of the remaining inventories and type of the customer so that we maximize the total expected revenue over the selling horizon.

We give a dynamic program to find the optimal policy to choose a personalized 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 subset of 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

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

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 $\phi_{ij}(S\cup\{k\}) \le \phi_{ij}(S)$ 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

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

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

In the problem above, opt corresponds to the maximum total expected revenue that we can obtain by jointly choosing the stocking quantities and making the personalized 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. Furthermore, even if we have access to the value functions $\{J_t : t\in\mathcal{T}\}$, the value functions may not have any structure that allows us to solve problem (2) efficiently. Motivated by these observations, 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 personalized 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 particular, we will use a linear program to construct the surrogate. 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 detailed 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 $\widehat{\boldsymbol{c}} = (\widehat{c}_1, \dots, \widehat{c}_n)$ 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: (Personalization) For $\beta \in (0,1]$, construct a policy to offer personalized 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 gives a blueprint with gaps to fill in. In particular, we need to choose a surrogate that forms an upper bound on the value function in Step 1, obtain an approximate solution to problem (3) with the chosen surrogate in Step 2 and construct an assortment personalization policy whose total expected revenue is lower bounded by a fraction of the chosen surrogate in Step 3. Thus, we can view the approximation framework as a meta-algorithm with gaps. We will show that we can fill all of the gaps for our joint stocking and assortment personalization problem. Throughout the paper, we will use one surrogate function, but it is entirely possible that others come up with different surrogates that allow us to satisfy the three steps in the approximation framework, resulting in different approximation strategies. In the next theorem, we give a performance guarantee for our approximation framework.

Theorem 2.1 (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 personalization 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 personalization 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 the chain of inequalities $\mathsf{Rev}(\widehat{\boldsymbol{c}}) \geq \beta f(\widehat{\boldsymbol{c}}) \geq \alpha\beta f(\boldsymbol{c}^*) \geq \alpha\beta J_1(\boldsymbol{c}^*) = \alpha\beta\,\mathsf{opt}$. ■

Thus, we can obtain an $\alpha\beta$-approximate solution to the joint stocking and assortment personalization problem through our approximation framework. In the next section, we give our surrogate and describe the performance guarantees that we can attain by using this surrogate.

3. 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 personalization 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. 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. Using the decision variables $\boldsymbol{w} = (w_j(S) : j \in \mathcal{M},\ S \subseteq \mathcal{N})$ and using $\tau_j = \sum_{t\in\mathcal{T}} \lambda_{jt}$ to denote the total expected number of customer arrivals of type j over the selling horizon, our surrogate $f : \mathbb{Z}^n_+ \to \mathbb{R}_+$ is given by

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

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. Linear programming approximations of this form 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 this surrogate and $f(\boldsymbol{c})$ will always denote the surrogate given by the optimal objective function of (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. It is straightforward to extend their result to multiple customer types. Thus, we have $f(\boldsymbol{c}) \ge J_1(\boldsymbol{c})$, which is the property that the surrogate function needs to satisfy in Step 1 of our approximation framework. The number of decision variables in (4) increases exponentially with the number of products, but we can solve the separation subproblem efficiently under a variety of choice models, such as the multinomial logit, nested logit, generalized attraction, Markov chain and a mixture of independent demand and multinomial logit models; Talluri and van Ryzin (2004), Davis et al. (2014), Gallego et al. (2015), Blanchet et al. (2016) and Cao et al. (2021). Therefore, we can solve problem (4) efficiently under many choice models. Moving forward, we assume that we can solve the separation subproblem efficiently. Using the linear programming-based surrogate, we will be able to execute Steps 2 and 3 of our approximation framework to get performance guarantees for our joint stocking and assortment personalization problem, as explained next.

Outline and Main Results:

Using our approximation framework, we give two performance guarantees for our joint stocking and assortment personalization problem. To get either of the two performance guarantees, we use the linear programing-based surrogate in (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 4, we show that we can obtain a $\frac{1}{2}(1-\frac{1}{e})$-approximate solution $\widehat{\boldsymbol{c}}$ to problem (3) with the linear programming-based surrogate in polynomial time. Thus, we can execute Step 2 with $\alpha=\frac{1}{2}(1-\frac{1}{e})$ when the customers choose 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 5, 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 6, 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 personalization 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 2.1, we get the following two performance guarantees for our joint stocking and assortment personalization problem.

Theorem 3.1 (Performance Guarantees) *We have the following two performance guarantees for the joint stocking and assortment personalization 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 2.1, 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 2.1, 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 for our joint stocking and assortment personalization problem. 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 a solution that is asymptotically optimal. There are no hidden constants in the second part, so we can use the second part of the theorem to also obtain performance guarantees that depend on n and K. For example, if the storage capacity satisfies $K \geq \kappa n$ for some $\kappa \geq 1$, then we obtain a performance guarantee of $1-(\sqrt{2}+1)/\kappa^{1/3}$ when the customers choose according to a general choice model. This performance guarantee does not depend on problem data other than n and K. We emphasize that the asymptotic optimality result in the second part of the theorem is different from other asymptotic optimality results that appear 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 formulations of dynamic pricing, capacity control and assortment offering problems; see, for example, Gallego and van Ryzin (1994), Jasin and Kumar (2012), Feng et al. (2020) and Ma et al. (2021). In our asymptotic optimality result, the storage capacity grows arbitrarily large, which does not necessarily imply that the initial inventories for all products grow large. Another point worth noting is that we do not make any assumptions on at what rate the demand volume grows either. Next, we focus on finding approximate solutions to problem (3) when the customers choose according to the multinomial logit and a general choice model, as well as constructing an assortment personalization policy.

4. 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 it is NP-hard to approximate problem (3) within a factor $1-\frac{1}{e}$ even under the multinomial logit model.

Theorem 4.1 (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, which is given in Appendix A, uses a reduction from the maximum coverage problem. Motivated by this complexity result, 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 were a monotone and submodular function, then problem (3) would correspond to 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 counterexamples to show that the linear programming-based surrogate $f(\boldsymbol{c})$ is not submodular in $\boldsymbol{c}$, even under the multinomial logit model. We give one counterexample in Appendix B. To get around this difficulty, 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 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

$$f(\boldsymbol{c}) \;=\; \max_{(\boldsymbol{y},\boldsymbol{y}_0)\in\mathbb{R}_+^{nm+m}}\left\{\sum_{j\in\mathcal{M}}\sum_{i\in\mathcal{N}} r_i\,y_{ij} \;:\; \sum_{j\in\mathcal{M}} y_{ij}\le c_i\quad\forall\, i\in\mathcal{N},\right. \qquad (5)$$
$$\left.\sum_{i\in\mathcal{N}} y_{ij}+y_{0j}\le\tau_j\quad\forall\, j\in\mathcal{M},\qquad \frac{y_{ij}}{v_{ij}}\le y_{0j}\quad\forall\, i\in\mathcal{N},\ j\in\mathcal{M}\right\}.$$

We can show that if the customers choose under the multinomial logit model so that the choice probabilities are of the form $\phi_{ij}(S)=v_{ij}/(1+\sum_{k\in S}v_{kj})$ for all $i\in S$, 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 $\boldsymbol{c}$. Problem (5) has an intuitive interpretation. Noting the definition of the decision variable y_{ij}, the objective function accounts for the total expected revenue. The first constraint ensures that the total expected purchases for product i by customers of all 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, ensures the purchases for the different products are aligned 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

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

$$\left. \sum_{i\in\mathcal{N}} y_{ij} \le \frac{1}{2}\,\tau_j \quad \forall\, j\in\mathcal{M}, \qquad y_{ij} \le \frac{1}{2}\, v_{ij}\,\tau_j \quad \forall\, i\in\mathcal{N},\; j\in\mathcal{M} \right\}.$$

We obtain (6) directly by fixing $y_{0j} = \tau_j/2$ in (5). By the next lemma, the optimal objective value of (6) approximates the linear programming-based surrogate within a factor of two.

Lemma 4.2 (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 some of the decision variables 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 (6). Because $(\boldsymbol{y}^*, \boldsymbol{y}_0^*)$ is feasible to (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 (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, $\frac{1}{2}\boldsymbol{y}^*$ is a feasible solution to problem (6). 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 the lemma above, the approximate surrogate approximates the linear programming-based surrogate 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 also 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 $f_{\mathsf{app}}(\boldsymbol{c}) = \min_{\boldsymbol{\mu} \in \mathbb{R}_+^n} \left\{ L(\boldsymbol{c}, \boldsymbol{\mu}) + \sum_{j \in \mathcal{M}} \frac{\tau_j}{2}\, G_j(\boldsymbol{\mu}) \right\}$. 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 $G_j(\boldsymbol{\mu}) = \max_{\boldsymbol{z}_j \in \mathbb{R}_+^n} \left\{ \sum_{i \in \mathcal{N}} (r_i - \mu_i)\, z_{ij} : \sum_{i \in \mathcal{N}} z_{ij} \leq 1,\ z_{ij} \leq v_{ij} \ \forall\, i \in \mathcal{N} \right\}$, which is a knapsack problem. Thus, $G_j(\boldsymbol{\mu})$ is the optimal objective value of a knapsack problem when viewed as a function of its objective function coefficients. In the next theorem, we build on these observations to show that the approximate surrogate is submodular.

Theorem 4.3 (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 $f_{\mathsf{app}}(\boldsymbol{c}) = \min_{\boldsymbol{\mu} \in \mathbb{R}_+^n} \left\{ L(\boldsymbol{c}, \boldsymbol{\mu}) + \sum_{j \in \mathcal{M}} \frac{\tau_j}{2}\, G_j(\boldsymbol{\mu}) \right\}$. 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 $\boldsymbol{\mu}, \boldsymbol{\eta} \in \mathbb{R}_+^n$. A simple lemma, given as Lemma C.1 in Appendix C, 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

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

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 transportation problem on a bipartite graph, 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 transportation problem on a bipartite graph 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 4.3, allowing upper bounds on the flows, is new. Since $f_{\mathsf{app}}(\boldsymbol{c})$ is monotone and submodular in $\boldsymbol{c}$ over the integer lattice, 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. 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).

Note that our equivalent reformulation of the linear programming-based surrogate as in (5) and approximating the optimal objective value of (5) by using a transportation problem as in (6) both play a role in our approach for making stocking decisions under the multinomial logit model. By the discussion in this section, we can execute Step 2 of our approximation framework under the multinomial logit model. Next, we focus on the same step under a general choice model.

5. 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

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

The number of decision variables above increases exponentially with the number of products, but the separation subproblem for problem (7) has the same structure as the separation subproblem for problem (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 5.1 (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

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

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 $x \ge 1$, 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 $\widehat{c}_i \ge \lfloor(\frac{K}{n})^{2/3}\rfloor \ge \frac{1}{2}(\frac{K}{n})^{2/3}$ 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 personalization 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.

6. Assortment Personalization Policy

We consider constructing an assortment personalization 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 personalization 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 personalization 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 personalization 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 Personalization 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 personalization policy over time periods $t,\dots,T$ 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

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

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 the approximate 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 personalization 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 personalization 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 policy in Step 3 of our approximation framework with $\beta = \max\{\frac{1}{2}, 1-\frac{1}{\sqrt{\widehat{c}_{\min}}}\}$.

Theorem 6.1 (Policy Performance) *The total expected revenue obtained by the assortment personalization policy with the initial stocking quantities $\widehat{\boldsymbol{c}}$ satisfies $V_1(\widehat{\boldsymbol{c}}) \geq \max\{\frac{1}{2}, 1-\frac{1}{\sqrt{\widehat{c}_{\min}}}\}\, f(\widehat{\boldsymbol{c}})$.*

In the proof of the theorem, we consider an inventory-agnostic assortment personalization 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 can show that the total expected revenue of the inventory-agnostic policy provides a lower bound on that of our assortment personalization policy, so lower bounding the former is enough to lower bound the latter. Since the inventory-agnostic policy does not pay attention to the remaining inventories, 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 a novel inequality that shows 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 gets large, this inequality becomes 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 personalization 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.

Proof of Theorem 6.1:

Let $\widetilde{V}_t(\boldsymbol{x})$ be the total expected revenue of the inventory-agnostic 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 have

$$\widetilde{V}_t(\boldsymbol{x}) = \sum_{j\in\mathcal{M}} \lambda_{jt} \sum_{S\subseteq\mathcal{N}} \frac{\widehat{w}_j(S)}{\tau_j} \left\{ \sum_{i\in\mathcal{N}} \phi_{ij}(S)\, \mathbf{1}(x_i \geq 1) \left[r_i + \widetilde{V}_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i) - \widetilde{V}_{t+1}(\boldsymbol{x}) \right] \right\} + \widetilde{V}_{t+1}(\boldsymbol{x}), \qquad (9)$$

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. We can show that the value functions computed through (9) provide lower bounds on those computed through (8). That is, we have $\widetilde{V}_t(\boldsymbol{x}) \leq V_t(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{Z}^n_+$ and $t \in \mathcal{T}$. We show this result in Lemma D.1 in Appendix D. The proof of the result uses induction over the time periods. We proceed to giving a closed form expression for the value functions in (9) next.

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 (9).

Lemma 6.2 (Equivalent Form for Value Functions) *Letting the value functions* $\{\widetilde{V}_t : t \in \mathcal{T}\}$ *be computed through the dynamic program in* (9)*, 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 $\sum_{\tau=t+1}^T \beta_{i\tau} - \mathbb{E}\{[Z_{i,t+1} - x_i]^+\} = \mathbf{1}(x_i \geq 1)\,(\sum_{\tau=t+1}^T \beta_{i\tau} - \mathbb{E}\{[Z_{i,t+1} - x_i]^+\})$. 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 (9), we obtain the chain of equalities

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

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 $\mathbb{P}\{Y_{it} = 1\} = \beta_{it}$. ■

We give the proof of Theorem 6.1 using the lemma above.

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 $\sum_{i \in \mathcal{N}} \sum_{t \in \mathcal{T}} r_i\, \beta_{it} = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} \sum_{S \subseteq \mathcal{M}} \phi_{ij}(S)\, \widehat{w}_j(S) = f(\widehat{\boldsymbol{c}})$, 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 D.1 in Appendix D, 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 E.1 in Appendix E, 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 $\mathbb{E}\{[Z-a]^+\} \leq \min\{\frac{1}{2}, \frac{1}{\sqrt{a}}\}\,\mathbb{E}\{Z\}$. Using these observations together, we obtain

$$\begin{aligned}
V_1(\widehat{\boldsymbol{c}}) \;\geq\; \widetilde{V}_1(\widehat{\boldsymbol{c}}) &\overset{(a)}{=} \sum_{i\in\mathcal{N}} r_i \left(\sum_{t\in\mathcal{T}} \beta_{it} - \mathbb{E}\Big\{[Z_{i1}-\widehat{c}_i]^+\Big\}\right) \\
&\overset{(b)}{=} \sum_{i\in\mathcal{N}}\sum_{t\in\mathcal{T}} r_i\,\beta_{it}\left(1-\frac{\mathbb{E}\{[Z_{i1}-\widehat{c}_i]^+\}}{\mathbb{E}\{Z_{i1}\}}\right) \overset{(c)}{\geq} \sum_{i\in\mathcal{N}}\sum_{t\in\mathcal{T}} r_i\,\beta_{it}\left(1-\min\left\{\frac{1}{2},\frac{1}{\sqrt{\widehat{c}_i}}\right\}\right) \\
&\geq \sum_{i\in\mathcal{N}}\sum_{t\in\mathcal{T}} r_i\,\beta_{it}\left(1-\min\left\{\frac{1}{2},\frac{1}{\sqrt{\widehat{c}_{\min}}}\right\}\right) \overset{(d)}{=} \max\left\{\frac{1}{2}, 1-\frac{1}{\sqrt{\widehat{c}_{\min}}}\right\} f(\widehat{\boldsymbol{c}}), \qquad (10)
\end{aligned}$$

where (a) uses Lemma 6.2, (b) holds because $\mathbb{E}\{Z_{i1}\} = \sum_{t\in\mathcal{T}} \beta_{it}$, (c) uses Lemma E.1 and the fact that $\mathbb{E}\{Z_{i1}\} \leq \widehat{c}_i$ and (d) follows because $f(\widehat{\boldsymbol{c}}) = \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{N}} r_i\,\beta_{it}$.

7. Computational Experiments

We give two sets of computational experiments. The first one is based on synthetically generated datasets. The second one is based on a dataset from real-world supermarket purchases.

7.1 Synthetic Datasets

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

Experimental Setup: We use the following approach to generate our test problems. 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 in our computational experiments. 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 purchasing 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 and 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 $\mathsf{Demand} = \sum_{i \in \mathcal{N}} \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{M}} \lambda_{jt}\, \phi_{ij}(\widetilde{S}_j)$. 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. 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 obtain 24 parameter configurations. For each parameter configuration, we generate a test problem as above.

Benchmark Policies: We use six benchmark policies motivated by our approximation framework, as well as a heuristic based on a newsvendor approximation.

Greedy Stocking and Randomized Personalization (GRa): Here, for stocking, 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 at the end of Section 4, we execute Step 2 in our approximation framework with $\alpha = \frac{1}{2}(1 - \frac{1}{e})$. For assortment personalization, we use the policy in Section 6, so by Theorem 6.1, we have $\beta = \frac{1}{2}$ in Step 3.

Greedy Stocking and Rollout Personalization (GRo): In this benchmark, we make the stocking decisions as in GRa, so we execute Step 2 in our approximation framework with $\alpha = \frac{1}{2}(1 - \frac{1}{e})$. By Lemma 6.2, 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 the assortment personalization policy, if we are at time period t with the remaining inventories $\boldsymbol{x}$, then we offer the assortment $\arg\max_{S\subseteq\mathcal{N}} \sum_{i\in\mathcal{N}} \phi_{ij}(S)\,\mathbf{1}(x_i \geq 1)\,[r_i + \widetilde{V}_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i) - \widetilde{V}_{t+1}(\boldsymbol{x})]$ to a customer of type j, which corresponds to the greedy policy with respect to the value functions $\{\widetilde{V}_t : t\in\mathcal{T}\}$. There exists an optimal solution to the last optimization problem such that if product i does not have remaining inventory so that $x_i = 0$, then this product is not offered. In particular, the revenue contribution of product i in the last problem is $\mathbf{1}(x_i \geq 1)\,[r_i + \widetilde{V}_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i) - \widetilde{V}_{t+1}(\boldsymbol{x})]$. If $x_i = 0$, then the revenue contribution of product i is zero. If we remove all products with non-positive revenue contributions from the optimal solution, then the choice probabilities of all remaining products does not decrease and we eliminate all products with non-positive revenue contributions, yielding a solution that must be at least as good as the optimal solution. So, the assortment personalization policy does not offer a product that does not have remaining inventory. This assortment personalization policy corresponds to performing rollout on the inventory-agnostic policy; see, Section 6.1.3 in Bertsekas and Tsitsiklis (1996). Thus, 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 $\widetilde{V}_1(\widehat{\boldsymbol{c}})$. By (10), we have $\widetilde{V}_1(\widehat{\boldsymbol{c}}) \geq \frac{1}{2} f(\widehat{\boldsymbol{c}})$, so if we use this assortment personalization policy, then we can execute Step 3 in our approximation framework with $\beta = \frac{1}{2}$.

<u>Stocking with Multiple Estimates and Randomized Personalization (MRa)</u>: We construct the approximate surrogate in (6) by fixing the value of the decision variable y_{0j} in problem (5) at $\tau_j/2$ for all $j\in\mathcal{M}$. In this benchmark, we try other values for y_{0j}. In particular, 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 at the end of Section 4, the solution $\widehat{\boldsymbol{c}}^{0.5}$ is a $\frac{1}{2}(1-\frac{1}{e})$-approximation to problem (3), so the solution $\widehat{\boldsymbol{c}}$ is a $\frac{1}{2}(1-\frac{1}{e})$-approximation to problem (3) as well. Thus, we execute Step 2 in our approximation framework with $\alpha = \frac{1}{2}\,(1-\frac{1}{e})$. We follow the same assortment personalization policy that we use for GRa, so we execute Step 3 in our approximation framework with $\beta = \frac{1}{2}$. The benchmarks GRa, GRo and MRa all yield the same performance guarantee, but GRo uses a more sophisticated approach for assortment personalization and MRa uses a more sophisticated approach for stocking.

<u>Stocking with Multiple Estimates and Rollout Personalization (MRo)</u>: In this benchmark, we make the stocking decisions as in MRa and assortment personalization decisions as in GRo.

<u>Rounded Stocking and Randomized Personalization (RRa)</u>: 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 5, 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 personalization, we use the policy in Section 6, so noting Theorem 6.1, we execute Step 3 of our approximation framework with $\beta = 1 - \sqrt{2}\,\sqrt[3]{\frac{n}{K}}$. In the proof of Theorem 5.1, 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 Personalization (RRo): In this benchmark, we make the stocking decisions as in RRa and assortment personalization decisions as in GRo.

Newsvendor Heuristic (NVH): In this benchmark, we follow a sensible solution approach that one could implement in practice for the joint stocking and assortment customization problem. The goal of this benchmark is to test our approximation framework against a solution strategy that one could devise without having access to the ideas in this paper. In this benchmark, we pre-compute the assortments to offer to each customer type and solve a newsvendor-like model to find the stocking quantities under the demand distributions that are driven by the pre-computed 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 offer the myopic assortment $\widetilde{S}_j$ to a customer of type j at all time periods, then the demand for product i at time period t is given by 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 $X_i = \sum_{t\in\mathcal{T}} X_{it}$. To make the stocking decisions, we solve the problem $\max_{\boldsymbol{c}\in\mathbb{Z}_+^n} \{\sum_{i\in\mathcal{N}} r_i\, \mathbb{E}\{\min\{c_i, X_i\}\} : \sum_{i\in\mathcal{N}} c_i \leq K\}$. In this problem, the understanding is that if we stock c_i units for product i, then the sales for this product is $\min\{c_i, X_i\}$. The last objective function is concave in $\boldsymbol{c}$ and additive by the products, so we can find the stocking quantities in polynomial time, after approximating the distribution of X_i via Monte Carlo simulation. In the assortment personalization policy, 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 $\boldsymbol{x}$. Note that this benchmark does not come with a performance guarantee.

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

Params. $(T, P_0, \theta_{\min}, \eta)$	K	Total Exp. Rev. GRA	GRO	MRA	MRO	RRA	RRO	NVH	Rankings of Total Exp. Rev.
(4000, 0.1, 0.005, 0.25)	758	96.14	99.50	96.49	99.60	74.99	79.55	93.34	[4, 2, 3, 1, 7, 6, 5]
(4000, 0.1, 0.005, 0.50)	1517	95.30	98.17	95.30	98.17	76.67	80.47	90.31	[3, 1, 3, 1, 7, 6, 5]
(4000, 0.1, 0.010, 0.25)	755	96.07	99.46	96.49	99.63	74.86	79.48	93.46	[4, 2, 3, 1, 7, 6, 5]
(4000, 0.1, 0.010, 0.50)	1511	94.83	98.04	94.83	98.04	76.70	80.44	89.33	[3, 1, 3, 1, 7, 6, 5]
(4000, 0.3, 0.005, 0.25)	563	90.56	95.84	93.54	96.69	66.22	73.44	92.31	[5, 2, 3, 1, 7, 6, 4]
(4000, 0.3, 0.005, 0.50)	1126	90.06	94.68	92.34	95.72	74.88	79.31	89.85	[4, 2, 3, 1, 7, 6, 5]
(4000, 0.3, 0.010, 0.25)	560	89.79	95.54	93.07	96.76	66.24	73.33	91.32	[5, 2, 3, 1, 7, 6, 4]
(4000, 0.3, 0.010, 0.50)	1120	89.52	94.47	91.96	94.86	74.77	79.70	89.69	[5, 2, 3, 1, 7, 6, 4]
(8000, 0.1, 0.005, 0.25)	1517	97.13	99.50	97.45	99.65	76.17	79.56	93.76	[4, 2, 3, 1, 7, 6, 5]
(8000, 0.1, 0.005, 0.50)	3035	96.27	98.49	96.27	98.49	82.74	85.35	90.65	[3, 1, 3, 1, 7, 6, 5]
(8000, 0.1, 0.010, 0.25)	1511	97.09	99.46	97.42	99.65	76.05	79.49	93.88	[4, 2, 3, 1, 7, 6, 5]
(8000, 0.1, 0.010, 0.50)	3022	95.95	98.29	95.95	98.29	82.73	85.32	89.78	[3, 1, 3, 1, 7, 6, 5]
(8000, 0.3, 0.005, 0.25)	1126	92.23	95.87	95.05	97.61	72.69	77.88	93.03	[5, 2, 3, 1, 7, 6, 4]
(8000, 0.3, 0.005, 0.50)	2252	91.59	95.01	93.86	96.61	82.18	85.77	90.29	[4, 2, 3, 1, 7, 6, 5]
(8000, 0.3, 0.010, 0.25)	1120	91.67	95.66	94.70	97.54	72.48	77.76	92.05	[5, 2, 3, 1, 7, 6, 4]
(8000, 0.3, 0.010, 0.50)	2240	91.18	94.85	93.64	95.88	82.20	86.07	90.26	[4, 2, 3, 1, 7, 6, 5]
(16000, 0.1, 0.005, 0.25)	3035	97.88	99.50	98.13	99.66	82.43	84.67	94.03	[4, 2, 3, 1, 7, 6, 5]
(16000, 0.1, 0.005, 0.50)	6070	97.01	98.64	97.01	98.64	86.05	87.79	90.93	[3, 1, 3, 1, 7, 6, 5]
(16000, 0.1, 0.010, 0.25)	3022	97.79	99.46	98.07	99.65	82.39	84.62	94.21	[4, 2, 3, 1, 7, 6, 5]
(16000, 0.1, 0.010, 0.50)	6045	96.65	98.34	96.65	98.34	85.87	87.76	90.14	[3, 1, 3, 1, 7, 6, 5]
(16000, 0.3, 0.005, 0.25)	2252	93.32	95.89	96.25	98.32	81.01	84.51	93.32	[5, 3, 2, 1, 7, 6, 4]
(16000, 0.3, 0.005, 0.50)	4505	92.64	95.14	94.92	97.20	85.37	88.01	90.70	[4, 2, 3, 1, 7, 6, 5]
(16000, 0.3, 0.010, 0.25)	2240	92.83	95.69	95.93	98.15	80.83	84.44	92.57	[4, 3, 2, 1, 7, 6, 5]
(16000, 0.3, 0.010, 0.50)	4481	92.31	94.94	94.84	96.63	85.33	88.15	90.55	[4, 2, 3, 1, 7, 6, 5]
Avg.		93.99	97.10	95.42	97.91	78.41	82.20	91.66	

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

expected revenue. In particular, our linear programming-based surrogate $f(\boldsymbol{c})$ satisfies $f(\boldsymbol{c}) \geq J_1(\boldsymbol{c})$ 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 on the optimal total expected revenue efficiently, we solve the linear programming relaxation of problem (3). This linear programming relaxation is equivalent to problem (4) if we treat the stocking quantities $\boldsymbol{c}$ in this problem as continuous decision variables as well, instead of fixed numbers. In the table, the first column gives parameter configuration for each test problem using the tuple $(T, P_0, \theta_{\min}, \eta)$. The second column gives the value of the storage capacity K. The next seven columns give the total expected revenues obtained by each benchmark expressed as a percentage of the upper bound. The last column gives the rankings of the total expected revenues obtained by the benchmarks in the order they are listed in the previous seven columns. For the first test problem, for example, GRA has the 4-th highest total expected revenue, whereas NVH has the 5-th. 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. 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 personalization 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 personalization decisions, GRa directly uses the assortment personalization policy in Section 6. Thus, MRo goes one step beyond GRa in both stocking and assortment personalization decisions. This effort pays off and MRo improves the performance of GRa by 4.16% on average. In addition to consistently being the strongest benchmark, MRo obtains total expected revenues within 2.09% of the upper bound on the optimal total expected revenue on average, indicating that this benchmark leaves little on the table. 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 personalization decisions. The total expected revenues of MRo and GRo are followed by those of MRa, then those of GRa.

The benchmarks RRa and RRo are inferior to MRo, GRo, MRa and GRa. In RRa and RRo, we execute Steps 2 and 3 in 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, respectively, by 19.73%, 14.42% and 9.07%. The analogous gaps for RRo are, respectively, 14.20%, 10.34% and 6.26%. Thus, both RRa and RRo start catching up with MRo as the storage capacity gets large. For the test problems with $T = 4000, 8000$ and 16000, the average storage capacities are, respectively, 988, 1977 and 3955. We shortly study the performance of RRa and RRo with larger storage capacities.

The total expected revenues of NVH are noticeably smaller than those of MRo, GRo, MRa and GRa, but larger than those of RRa and RRo. Over all of our test problems, the average total expected revenues of NVH is within 9.34% of the upper bound, but using our strongest benchmark, we can obtain average total expected revenues within 2.09% of the upper bound. Thus, there is significant value in carefully coordinating the stocking and assortment personalization decisions through our approximation framework. Lastly, we observe that the performance of GRa and MRa are identical for some test problems. In these test problems, MRa chooses the stocking decisions obtained by the surrogate $f^{0.5}_{\mathsf{app}}(\boldsymbol{c})$, which is the approximate surrogate $f_{\mathsf{app}}(\boldsymbol{c})$ used by GRa. Similarly, the performance of GRo and MRo may be identical for some test problems as well. We focus on the performance of RRa and RRo for test problems with larger storage capacities next.

In Table 2, we consider the parameters $P_0 = 0.1$, $\theta_{\min} = 0.01$ and $\eta = 0.75$. Varying $T \in \{4000, 8000, 16000, 32000, 64000, 128000\}$, we obtain six test problems. The first column in the table shows the value of the number of time periods in the selling horizon T, whereas the second column shows the value of the corresponding storage capacity K. The next three columns show

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

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.

7.2 Supermarket Purchase Datasets

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

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 $\{P_{ij}(\ell) : i \in \mathcal{N},\ j \in \mathcal{M},\ \ell = 1, \dots, 52\}$. 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 Vulcano and van Ryzin (2020) 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 give the details of our fitting approach in Appendix F.

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 length of the selling horizon corresponds to one 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 arrival probabilities for different customer types are stationary and the probability of observing a customer of type j 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. Total number of weeks product i is offered in supermarket j is $\kappa_{ij} = \sum_{\ell=1}^{52} \mathbf{1}(i \in S_j(\ell))$. 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 $r_i = \sum_{\ell=1}^{52} \sum_{j\in\mathcal{M}} \mathbf{1}(i \in S_j(\ell))\, p_{ij}(\ell) / \sum_{j\in\mathcal{M}} \kappa_{ij}$, which corresponds to 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 $\mathsf{Demand} = \sum_{i\in\mathcal{N}}\sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{M}} \lambda_{jt}\,\phi_{ij}(\widetilde{S}_j)$. 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 personalization 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 $\eta\in\{0.25, 0.5, 0.75\}$, using $\{1,2,3,4\}$ to denote the four product categories, we have 24 parameter configurations for our test problems.

Computational Results: We continue using the same seven benchmarks that we used for the synthetic datasets. We give our computational results in Table 3. In the table, 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. 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 eight columns in the table have the same interpretation as those in Table 1. Our results are largely aligned with those for the synthetic datasets. The strongest benchmark is MRo, followed by GRo and MRa. Working with multiple surrogates when making the stocking decisions and using rollout when making the assortment personalization decisions pay off and MRo consistently performs better than the other benchmarks. The performance of RRa and RRo are not competitive to other benchmarks for these test problems, but their performance clearly starts improving as K gets larger. For one instance NVH is the strongest benchmark, but NVH generally lags behind MRo, GRo and MRa.

8. 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 personalization 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

Params. (C, P_0, η)	T	K	Total Exp. Rev. GRA	GRO	MRA	MRO	RRA	RRO	NVH	Rankings of Total Exp. Rev.
$(1, 0.1, 0.25)$	4532	775	93.82	97.60	95.35	97.78	71.22	76.44	89.71	$[4, 2, 3, 1, 7, 6, 5]$
$(1, 0.1, 0.50)$	4532	1551	88.22	91.43	94.27	94.22	74.82	78.64	80.99	$[4, 3, 1, 2, 7, 6, 5]$
$(1, 0.1, 0.75)$	4532	2326	94.28	93.97	94.28	93.97	83.05	85.15	86.65	$[1, 3, 1, 3, 7, 6, 5]$
$(1, 0.3, 0.25)$	5827	986	78.45	84.59	91.33	93.16	71.19	77.19	72.13	$[4, 3, 2, 1, 7, 5, 6]$
$(1, 0.3, 0.50)$	5827	1973	85.32	89.39	93.79	93.80	83.36	86.81	79.16	$[5, 3, 2, 1, 6, 4, 7]$
$(1, 0.3, 0.75)$	5827	2960	93.70	92.78	93.70	92.78	90.15	91.64	86.90	$[1, 3, 1, 3, 6, 5, 7]$
$(2, 0.1, 0.25)$	455	94	95.39	99.79	94.84	99.42	70.01	78.57	94.47	$[3, 1, 4, 2, 7, 6, 5]$
$(2, 0.1, 0.50)$	455	188	93.20	97.64	94.85	97.25	78.18	83.87	88.97	$[4, 1, 3, 2, 7, 6, 5]$
$(2, 0.1, 0.75)$	455	283	92.23	92.24	92.94	93.55	84.41	88.29	88.87	$[4, 3, 2, 1, 7, 6, 5]$
$(2, 0.3, 0.25)$	585	101	88.85	95.82	91.92	97.22	69.47	80.27	87.78	$[4, 2, 3, 1, 7, 6, 5]$
$(2, 0.3, 0.50)$	585	202	88.99	92.96	89.37	92.39	80.48	87.09	84.81	$[4, 1, 3, 2, 7, 5, 6]$
$(2, 0.3, 0.75)$	585	304	92.33	93.25	92.33	93.25	86.78	90.66	89.46	$[3, 1, 3, 1, 7, 5, 6]$
$(3, 0.1, 0.25)$	3439	661	95.68	97.69	97.21	98.58	83.15	86.38	88.26	$[4, 2, 3, 1, 7, 6, 5]$
$(3, 0.1, 0.50)$	3439	1323	95.18	96.81	95.93	96.91	89.42	91.49	93.19	$[4, 2, 3, 1, 7, 6, 5]$
$(3, 0.1, 0.75)$	3439	1985	94.37	95.15	96.87	97.23	91.38	92.70	95.88	$[5, 4, 2, 1, 7, 6, 3]$
$(3, 0.3, 0.25)$	4421	701	93.17	96.25	95.47	97.45	85.03	88.26	92.00	$[4, 2, 3, 1, 7, 6, 5]$
$(3, 0.3, 0.50)$	4421	1402	95.22	96.27	96.01	96.94	88.83	91.08	93.68	$[4, 2, 3, 1, 7, 6, 5]$
$(3, 0.3, 0.75)$	4421	2103	93.23	93.76	93.23	93.76	90.42	91.69	92.55	$[3, 1, 3, 1, 7, 6, 5]$
$(4, 0.1, 0.25)$	509	102	94.10	98.64	94.10	98.64	76.11	83.18	96.44	$[4, 1, 4, 1, 7, 6, 3]$
$(4, 0.1, 0.50)$	509	204	91.39	94.87	94.71	96.46	72.44	77.53	96.59	$[5, 3, 4, 2, 7, 6, 1]$
$(4, 0.1, 0.75)$	509	306	91.38	92.44	93.87	95.29	78.32	81.14	93.36	$[5, 4, 2, 1, 7, 6, 3]$
$(4, 0.3, 0.25)$	654	109	90.57	96.21	92.76	96.39	60.74	70.02	88.56	$[4, 2, 3, 1, 7, 6, 5]$
$(4, 0.3, 0.50)$	654	219	88.86	92.35	92.03	94.02	75.06	80.92	86.80	$[4, 2, 3, 1, 7, 6, 5]$
$(4, 0.3, 0.75)$	654	329	92.56	93.09	92.56	93.09	80.78	83.79	90.36	$[3, 1, 3, 1, 7, 6, 5]$
Avg.			91.69	94.37	93.90	95.56	79.78	84.28	89.07	

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

for joint stocking and assortment optimization 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, as long as we can solve the separation subproblem for problem (4) efficiently. Our efforts so far to tackle this problem have not been fruitful. 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, our stocking decisions are constrained by a cardinality constraint of the form $\sum_{i\in\mathcal{N}} c_i \le K$. Using results on maximizing submodular functions under a knapsack constraint, letting w_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 $\sum_{i\in\mathcal{N}} w_i\, c_i \le K$; see Lee et al. (2009). One can study other constraints, beside knapsack constraints. Fourth, online retailers stock products in multiple product categories. One way to address multiple product categories is to use our model to assess the optimal total expected revenue as a function of the capacity K allocated to one product category, in which case, we can solve an auxiliary optimization problem to allocate the total stocking capacity to each product category. The implicit assumption behind this approach is that the choice behavior in different product categories are not related to each other. One can work on extending our results to choice models that explicitly capture the interaction between multiple product categories. Fifth, throughout the paper, we use the linear programming-based surrogate. Our approximation framework is general enough to work with other surrogates. One can study other surrogates to obtain potentially stronger performance guarantees.

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.

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. 2021. Revenue management under a mixture of independent demand and multinomial logit models. *Operations Research* (forthcoming).

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.

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.

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.

Lee, J., V. S. Mirrokni, V. Nagarajan, M. Sviridenko. 2009. Non-monotone submodular maximization under matroid and knapsack constraints. *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*. Association for Computing Machinery, New York, NY, 323–332.

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.

Polard, 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.

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

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

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 Personalization

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

December 7, 2022

Appendix A: Computational Complexity

In this section, we give a proof for Theorem 4.1. 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

$$\max_{(\boldsymbol{c},\boldsymbol{y},\boldsymbol{y}_0)\in\mathbb{Z}_+^n\times\mathbb{R}_+^{nm+m}} \left\{ \sum_{j\in\mathcal{M}}\sum_{i\in\mathcal{N}} y_{ij} \; : \; \sum_{i\in\mathcal{N}} c_i \le K, \quad \sum_{j\in\mathcal{M}} y_{ij} \le c_i \;\; \forall\, i\in\mathcal{N}, \right. \\ \left. \sum_{i\in\mathcal{N}} y_{ij} + y_{0j} \le \tau_j \;\; \forall\, j\in\mathcal{M}, \quad y_{ij} \le v_{ij}\, y_{0j} \;\; \forall\, i\in\mathcal{N},\; j\in\mathcal{M} \right\}. \tag{11}$$

Our proof of Theorem 4.1 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 4.1:

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 (11) 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 (11) with an objective value of at least $\frac{1}{m}(1-\epsilon)\,Z^*$. In particular, we set $\widehat{c}_i = x_i^*$,

$$\widehat{y}_{ij} = \frac{v_{ij}\, x_i^*}{m\,(1+\sum_{k\in\mathcal{N}} v_{kj}\, x_k^*)}, \qquad \widehat{y}_{0j} = \frac{1}{m\,(1+\sum_{k\in\mathcal{N}} v_{kj}\, x_k^*)}. \tag{12}$$

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 (11). Note that $\widehat{y}_{ij} \le x_i^*/m$ by (12), 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 (11). By the definition of $\widehat{y}_{ij}$ and $\widehat{y}_{0j}$ in (12), 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 (11). Lastly, once again, by the definition of $\widehat{y}_{ij}$ and $\widehat{y}_{0j}$ in (12), 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 (11) as well. Thus, the solution $(\widehat{\boldsymbol{c}}, \widehat{\boldsymbol{y}}, \widehat{\boldsymbol{y}}_0)$ is feasible to problem (11). 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 (11), 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 (11) 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 (11), 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 (11), 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 (11), we have $\sum_{i\in\mathcal{N}} y_{ij}^* \le \tau_j = 1/m \le 1$. By the second constraint in (11), we have $y_{ij}^* \le c_i^* = \widehat{x}_i \le 1$, whereas by the third constraint in (11) 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 $\sum_{i\in\mathcal{N}} y^*_{ij} \le 1/m$ implies that $\sum_{i\in\mathcal{N}} y^*_{ij} \le \frac{1}{m}\max_{i\in\mathcal{N}}\{\mathbf{1}(j\in S_i)\,\widehat{x}_i\}$. Thus, for the maximum coverage problem, the solution $\widehat{\boldsymbol{x}}$ provides an objective value of

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

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 (11) with objective value $\widehat{R}$ such that $\widehat{R} \ge \frac{1}{m}(1-\epsilon)\,Z^*$. Furthermore, give any feasible solution to problem (11) with objective value R^*, we can construct a feasible solution to the maximum coverage problem with objective value $\widehat{Z}$ such that $\frac{1}{m}\widehat{Z} \ge R^*$. 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 (11) within a factor better than $(1-\frac{1}{e})\,(1-\epsilon) = 1-\frac{1}{e}-\epsilon\,(1-\frac{1}{e})$.

Appendix B: 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 $\boldsymbol{c} \ge \boldsymbol{b}$, 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 C: Submodularity of Approximate Surrogate

We use the following lemma in the proof of Theorem 4.3, 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

$$G_j(\boldsymbol{\mu}) = \max_{\boldsymbol{z}_j\in\mathbb{R}^n_+}\left\{\sum_{i\in\mathcal{N}}(r_i-\mu_i)\,z_{ij} \;:\; \sum_{i\in\mathcal{N}} z_{ij}\le 1,\; z_{ij}\le v_{ij}\;\forall\, i\in\mathcal{N}\right\}. \tag{13}$$

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

$$L(\boldsymbol{c}, \boldsymbol{\mu}) + L(\boldsymbol{b}, \boldsymbol{\eta}) \;\ge\; L(\boldsymbol{c}, \boldsymbol{\mu}\wedge\boldsymbol{\eta}) + L(\boldsymbol{b}, \boldsymbol{\mu}\vee\boldsymbol{\eta})$$
$$G_j(\boldsymbol{\mu}) + G_j(\boldsymbol{\eta}) \;\ge\; G_j(\boldsymbol{\mu}\wedge\boldsymbol{\eta}) + G_j(\boldsymbol{\mu}\vee\boldsymbol{\eta}).$$

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-diffentiability. 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 C.1. In the next auxiliary lemma, let $\boldsymbol{z}^*_j(\boldsymbol{\mu})$ be an optimal solution to problem (13) as a function of $\boldsymbol{\mu}$.

Lemma C.2 (Optimal Solution to Knapsack) *There exists an optimal solution to problem* (13) *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 (13). 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 (13) 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 C.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 C.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 C.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 D: Performance of Assortment Personalization Policy

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 (9), we show that $\widetilde{V}_t(\boldsymbol{x})$ lower bounds $V_t(\boldsymbol{x})$.

Lemma D.1 (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 (9), we have $V_t(\boldsymbol{x}) \geq \widetilde{V}_t(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{Z}^n_+$ and $t \in \mathcal{T}$.*

To show the lemma above, we will use the auxiliary lemma below, where we give an upper bound on the first difference of the value functions $\{\widetilde{V}_t : t \in \mathcal{T}\}$.

Lemma D.2 (First Differences) *Letting the value functions $\{\widetilde{V}_t : t \in \mathcal{T}\}$ be computed by (9), we have $\widetilde{V}_t(\boldsymbol{x}) - \widetilde{V}_t(\boldsymbol{x} - \boldsymbol{e}_i) \leq r_i$ for all $\boldsymbol{x} \in \mathbb{Z}^n_+$ such that $x_i \geq 1$ 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

$$\widetilde{v}_{it}(x_i) = \sum_{j \in \mathcal{M}} \lambda_{jt} \sum_{S \subseteq \mathcal{N}} \frac{\widehat{w}_j(S)}{\tau_j} \left\{ \phi_{ij}(S)\, \mathbf{1}(x_i \geq 1) \left[r_i + \widetilde{v}_{i,t+1}(x_i - 1) - \widetilde{v}_{i,t+1}(x_i) \right] \right\} + \widetilde{v}_{i,t+1}(x_i). \quad (14)$$

Comparing the dynamic programs in (9) and (14), 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 $\widetilde{v}_{it}(x_i) - \widetilde{v}_{it}(x_i - 1) \leq r_i$ for all $x_i \in \mathbb{Z}_+$ with $x_i \geq 1$ 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 $\widetilde{v}_{i,t+1}(x_i) - \widetilde{v}_{i,t+1}(x_i - 1) \leq r_i$, we show that $\widetilde{v}_{it}(x_i) - \widetilde{v}_{it}(x_i - 1) \leq r_i$ 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 (14) equivalently as $\widetilde{v}_{it}(x_{it}) = \psi_{it}\, \mathbf{1}(x_i \geq 1)\, [r_i + \widetilde{v}_{i,t+1}(x_i - 1) + \widetilde{v}_{i,t+1}(x_i)] + \widetilde{v}_{i,t+1}(x_i)$. Since $\widetilde{v}_{i,t+1}(x_i) - \widetilde{v}_{i,t+1}(x_i - 1) \leq r_i$ 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 (14) and substracting $v_{it}(x_i-1)$ from both sides of this expression, we obtain the chain of inequalities

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

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. ■

Proof of Lemma D.1:

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. If $i\notin S$, then $\phi_{ij}(S)=0$. Also $x_i\geq 1$ if and only if $i\in\mathcal{N}(\boldsymbol{x})$, so $\phi_{ij}(S\cap\mathcal{N}(\boldsymbol{x}))=\mathbf{1}(x_i\geq 1)\,\phi_{ij}(S\cap\mathcal{N}(\boldsymbol{x}))$. 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 $\sum_{j\in\mathcal{M}}\lambda_{jt}\leq 1$, $\sum_{S\subseteq\mathcal{N}}\widehat{w}_t(S)=\tau_j$ and $\sum_{i\in\mathcal{N}}\phi_{ij}(S\cap\mathcal{N}(\boldsymbol{x}))\leq 1$. Thus, noting that $V_{t+1}(\boldsymbol{x})\geq\widetilde{V}_{t+1}(\boldsymbol{x})$ 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

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

where (a) holds because $\phi_{ij}(S)\geq\phi_{ij}(Q)$ for all $i\in S$ and $S\subseteq Q$ by our assumption on the choice probabilities in Section 2 and $r_i\geq\widetilde{V}_{t+1}(\boldsymbol{x})-\widetilde{V}_{t+1}(\boldsymbol{x}-\boldsymbol{e}_i)$ by Lemma D.2, whereas (b) is by (9).

Appendix E: 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 6.1.

Lemma E.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 E.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 claim in the previous paragraph again, we immediately get $\mathbb{E}\{(X-a)^+\} \leq \frac{1}{2}\mathbb{E}\{X\} = \frac{1}{2}\sqrt{np} \leq \sqrt{np} = \frac{1}{\sqrt{a}}\mathbb{E}\{X\}$. 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

$$\begin{aligned}\mathbb{E}\{[X-a]^+\} \;=\; \sum_{\ell=1}^{n} \mathbb{P}\{[X-a]^+ \geq \ell\} \;=\; \sum_{\ell=1}^{n-a} \mathbb{P}\{X \geq a+\ell\} \\ \leq\; \sum_{\ell=1}^{n-a} \beta^\ell\, \mathbb{P}\{X \geq a\} \;\leq\; \frac{\beta}{1-\beta}\, \mathbb{P}\{X \geq a\}, \qquad (15)\end{aligned}$$

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 (15) 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 E.1, we will use the following theorem, which is given as Theorem 28 in Polard (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 E.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 E.2 to show Lemma E.1. Here is the proof of Lemma E.1.

Proof of Lemma E.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

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

where (a) is by Theorem E.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 E.2 and (d) holds because $\mathbb{E}\{Z\} = \mathbb{E}\{X\}$. ■

Appendix F: 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 in Section 7.2, 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 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.