Joint Assortment Optimization and Customization under a Mixture of Multinomial Logit Models: Value of Personalized Assortments

Omar El Housni Huseyin Topaloglu
School of Operations Research and Information Engineering, Cornell Tech, New York, NY 10044
{oe46,ht88}@cornell.edu

July 17, 2022

Abstract

We consider a joint assortment optimization and customization problem under a mixture of multinomial logit models. In this problem, a firm faces customers of different types, each making a choice within an offered assortment according to the multinomial logit model with different parameters. The problem takes place in two stages. In the first stage, the firm picks an assortment of products to carry subject to a cardinality constraint. In the second stage, a customer of a certain type arrives into the system. Observing the type of the customer, the firm customizes the assortment that it carries by, possibly, dropping products from the assortment. The goal of the firm is to find an assortment of products to carry and a customized assortment to offer to each customer type that can arrive in the second stage to maximize the expected revenue from a customer visit. The problem arises, for example, in online platforms, where retailers commit to a selection of products before the start of the selling season, but they can potentially customize the displayed assortment for each customer type. We refer to this problem as the *Customized Assortment Problem* (CAP). Letting m be the number of customer types, we show that the optimal expected revenue of CAP can be $\Omega(m)$ times greater than the optimal expected revenue of the corresponding model without customization and this bound is tight. We establish that CAP is NP-hard to approximate within a factor better than $1 - 1/e$, so we focus on providing an approximation framework for CAP. As our main technical contribution, we design a novel algorithm, which we refer to as Augmented Greedy, and building on it, we give a $\Omega(1/\log m)$-approximation algorithm to CAP. Also, we present a fully polynomial-time approximation scheme for CAP when the number of customer types is constant. Considering the case where we have a cardinally constraint on the assortment offered to each customer type in the second stage of CAP, we give a $\Omega(1/\sqrt{m \log m})$-approximation algorithm. In our computational experiments, we demonstrate the value of customization by using a dataset from Expedia and check the practical performance of our approximation algorithm.

1 Introduction

Discrete choice models have been seeing steadily increasing attention to capture the customer choice in revenue management. Using discrete choice models, we can model the fact that customers choose and substitute among the products. If a product is not offered, then a portion of the demand for this product shifts to other products, while the remaining portion is lost. Given that the customer demand can be shaped by changing the assortment of products offered to the customers, a natural question for the retailers is to choose an assortment of products to offer to their customers to maximize their expected revenues. There is significant amount of literature indicating that both brick-and-mortar stores and online retailers can increase their revenues by carefully choosing the assortment of products they carry. In contrast to brick-and-mortar stores, online retailers have access to tremendous amount of customer browsing and purchasing data. As a result, in addition to picking the assortment of products they carry, online retailers can display a customized assortment of products to each customer based on what is known about the preferences of the customer. A personalized assortment can potentially allow enhancing the customer experience, as well as improving the revenue of the retailer.

In this paper, we study a joint assortment optimization and customization problem under a mixture of multinomial logit models. In this problem, a firm faces different customer types, each making a choice with an offered assortment according to the multinomial logit model with different parameters. The problem takes place in two stages. In the first stage, the firm picks an assortment of products to carry, subject to a cardinality constraint. In the second stage, a customer of a certain type arrives into the system. Observing the type of the customer, the firm customizes the assortment that it carries by, possibly, dropping products from the assortment. The goal of the firm is to find an assortment of products to carry and a customized assortment to offer to each customer type to maximize the expected revenue from a customer visit. We refer to this problem as the *Customized Assortment Problem* (CAP). This problem is faced by almost all online retailers. For example, online grocers make an initial decision of what product variety to carry in each product category, but the grocery choices of different customer segments dramatically differ from each other. Thus, they can adjust the assortment offered to each customer based on what is known about the segment of the customer. Often times, online grocers operate warehouses in urban centers and they are severely limited by the variety of products they can inbound to these warehouses. The cardinality constraints may capture such limitations. Though not exactly, cardinality constraints may also serve as proxy to budget or storage limitations.

Despite its ubiquitous nature, to our knowledge, CAP remained fully unexplored until our work. A closely related problem is the assortment optimization under a mixture of multinomial logit models without customization (Rusmevichientong et al., 2014; Bront et al., 2009). In this problem, the firm still faces multiple customer types, each choosing according to a multinomial logit model with different parameters. The firm picks an assortment of products to carry, possibly subject to a cardinality constraint, but does not have the opportunity to customize the assortment. Customers of each type are offered the same assortment carried by the firm. The goal of the firm is to find an assortment of products to carry to maximize the expected revenue from a customer visit. We refer to this problem as *Mixed Multinomial Logit Assortment Problem* (MMNL). For MMNL, letting m be the number of customer types, there is no polynomial time algorithm with an approximation factor better than $O(1/m^{1-\epsilon})$ for any $\epsilon > 0$ (Désir et al., 2014). On one hand, CAP is operationally more complicated than MMNL with the presence of a second stage to customize the assortment. On the other hand, if there is no cardinality constraint in the first stage, then the optimal solution in CAP is for the firm to carry all products, so the decision in the first stage becomes trivial. In contrast, MMNL is NP-hard even when there is no cardinality constraint. Thus, the computational complexity of CAP is not clear at the first glance. If CAP is not solvable in polynomial time, then it is not clear to what accuracy we can approximate the problem.

1.1 Contributions

Our main contributions are in characterizing the complexity of CAP, giving tight bounds on the value of customization, and developing approximation algorithms for CAP.

Model and computational complexity. We show that CAP is NP-hard to approximate within a factor better than $(1 - \frac{1}{e} - \epsilon)$ for any $\epsilon > 0$ even in the special case where all the product revenues are equal and customers of each type arrive with equal probabilities. In view of this hardness result, we turn out attention to developing approximation algorithms for CAP.

Value of customization. We show that CAP, by customizing the assortment offered to each customer type, can substantially increase the expected revenue obtained by MMNL, which does not customize the assortment offered to different customer types. In particular, we give a family of instances for which the optimal expected revenue of CAP exceeds the optimal expected revenue of MMNL by a factor of $\Omega(m)$, where m is the number of customer types. Furthermore, this bound is tight in the sense that the expected revenue provided by CAP cannot exceed the expected revenue provided by MMNL by more than this factor (Theorem 3.1).

Augmented Greedy and approximation algorithms. As our main technical contribution, we develop an approximation framework for CAP. In particular, we design a novel algorithm that we refer to as Augmented Greedy. This algorithm is the building block of our algorithmic framework. Augmented Greedy considers a small number of subsets of products based on the ranking of the product revenues. For each subset, it executes a standard greedy algorithm that iteratively picks the product that provides the highest increase in the objective function and adds this product to the offered assortment. Augmented Greedy returns the assortment with the highest expected revenue over all considered subsets. Our main structural result lower bounds the expected revenue of the assortment returned by Augmented Greedy with a constant factor of the optimal expected revenue from a certain portion of customer types (Theorem 4.2). Since we can compare the expected revenue from the Augmented Greedy assortment with the optimal expected revenue from only a portion of customer types, this result does not immediately give an approximation for CAP.

Nevertheless, we show that Augmented Greedy gives $\Omega(1/\log m)$-approximation to CAP when all customer types have the same arrival probabilities (Theorem 5.1). Building on this result, we design an algorithm that gives $\Omega(1/\log m)$-approximation to CAP for general arrival probabilities (Theorem 5.2). The latter algorithm is based on applying Augmented Greedy recursively on several subsets of customer types and combining the solutions using a tractable dynamic program. Thus, our work demonstrates the stark difference between the complexity of CAP and MMNL. While MMNL does not admit a polynomial-time algorithm with an approximation factor better than $O(1/m^{1-\epsilon})$ for any $\epsilon > 0$, we are able to give an approximation algorithm for CAP with an approximation factor of $\Omega(1/\log m)$, establishing that the ability to customize the assortment offered to each customer type also makes the assortment optimization problem dramatically more tractable. As discussed earlier in this section, at a first glance, it is not clear whether CAP should be easier or harder to approximate than MMNL and our work closes this gap. We also consider an extension of CAP where we have cardinality constraints on the assortments offered in both the first and second stages. We give $\Omega(1/\sqrt{m \log m})$-approximation to CAP in this case.

Fully polynomial-time approximation scheme. We give a fully polynomial-time approximation scheme (FPTAS) for CAP when the number of customer types is constant (Theorem 6.1). Our FPTAS uses a geometric grid to guess the expected revenue from each customer type and solves a dynamic program to find a first stage assortment that realizes the guessed expected revenues. We show that CAP is NP-hard even with two customer types, so FPTAS is the best approximation guarantee we can aim for under a constant number of customer types. We also build on our approach for the FPTAS to give an integer programming formulation for CAP.

Computational study. We use a dataset from Expedia to demonstrate the value of customization. Our computational study indicates that customization can be rather important for certain customer types. In particular, we can achieve significantly higher expected revenues by offering customized assortments for such customer types, as opposed to offering a single assortment to all customer types. In addition, we check the practical performance of our approximation algorithms and integer programming formulation on randomly generated problem instances. Our approximation algorithms perform remarkably well, obtaining near-optimal solutions for an overwhelming majority of our problem instances.

1.2 Related literature

There is significant work on assortment optimization under the multinomial logit model, but none of this work focuses on customization. Gallego et al. (2004) and Talluri and van Ryzin (2004) study the assortment optimization problem under the multinomial logit model with a single customer type. They show that the optimal assortment is revenue-ordered in the sense that it includes a certain number of products with the largest revenues. Rusmevichientong et al. (2010) study the same problem when there is a constraint on the number of offered products, whereas Sumida et al. (2020) incorporate constraints that can be characterized by a totally unimodular constraint structure. Wang (2012) studies joint pricing and assortment optimization under the multinomial logit model. Jagabathula (2016) examines the performance of exchange heuristics that incrementally improve the assortment on hand by adding or removing products. Within a broad class of possible selling mechanisms, under the utility structure implied by the multinomial logit model, Ma (2022) shows that offering assortments is the optimal mechanism to extract the maximum expected revenue, but this result may not hold under other utility structures.

Assortment optimization problem under a mixture of multinomial logit models is also relevant to our work. In this problem, we have customers of different types, each choosing according to a multinomial logit model with different parameters. The goal is to find a single assortment that maximizes the expected revenue from a customer visit. Bront et al. (2009) show that the problem is NP-hard when the number of customer types is as large as number of products and give an integer programming formulation. Rusmevichientong et al. (2014) show that the problem is still NP-hard even with two customer types and study the performance of revenue-ordered assortments. Désir et al. (2014) give an inapproximability result and develop an FPTAS when the number of customer types is fixed. Méndez-Díaz et al. (2014) give valid cuts for the integer programming formulation of the problem, whereas Sen et al. (2018) study a more efficiently solvable conic programming formulation. Berbeglia and Joret (2020) analyze the performance of revenue-ordered assortments for general class of random utility maximization models, sharpening some of the earlier performance bounds. Feldman and Topaloglu (2015) give a tractable upper bound on the optimal expected revenue, which is useful as a benchmark when checking the optimality gap of heuristics.

Our joint assortment optimization and customization model can be interpreted as one way of enhancing the operational flexibility of the multinomial logit model. Feldman and Topaloglu (2018) seek to enhance the operational flexibility of the multinomial logit model by incorporating consideration sets, where each customer arrives into the system with a specific consideration set, ignores all offered products that are not in her consideration set, and chooses within the remaining products according to the multinomial logit model. The authors give an FPTAS when the possible consideration sets have a nested structure. Aouad et al. (2019) give a PTAS when the consideration set of a customer includes each product with a fixed probability. Wang and Sahin (2018) work with a variant of the multinomial logit model in which the customers tradeoff product search effort with the utility to form their consideration sets. Aouad et al. (2018) study the assortment optimization problem under dynamic substitution, where the assortment viewed by the customer corresponds to the set of products with remaining inventories and the goal is to pick the initial inventory levels of the products. Gao et al. (2020) study the assortment optimization problem under the multinomial logit model when the offered assortment is gradually revealed, as in online search results.

Several recent papers studied customization, focusing on either developing sophisticated choice models or solving pricing problems. Jagabathula and Vulcano (2018) develop a choice model to predict personal preferences. In their model, each customer has a partial order between the products that is encoded by a directed acyclic graph. In each store visit, the customer samples a preference ranking over the full set of products, while staying consistent with her partial order. She drops some of the products in the preference ranking that are not in her consideration set. Among the remaining products, she purchases the highest ranking product that is also available in the offered assortment. Jagabathula et al. (2020) use a similar model for running personalized promotions. Aouad et al. (2020) develop a choice model with multiple customer segments by segmenting the customers with the help of trees, but they do not consider personalized assortment optimization or pricing problems. Berbeglia et al. (2021) study assortment optimization problems in which the firm can make certain products unattractive to customers, which is modeled by allowing the firm to change the utilities of the products. In our paper, we give a $\Omega(1/\log m)$-approximation to CAP. After our work, Udwani (2021) gave a constant-factor approximation for CAP. His work studies optimization problems where one maximizes set functions that satisfy only a limited version of submodularity defined over a fixed permutation of the ground set. The author shows that the objective function of CAP satisfies this limited version of submodularity.

The rest of the paper is organized as follows. In Section 2, we formulate CAP and characterize its computational complexity. In Section 3, we give tight bounds on the benefit of customization. In Section 4, we describe the Augmented Greedy algorithm. In Section 5, we develop and analyze our $\Omega(1/\log m)$-approximation algorithm for CAP. In Section 6, we give our FPTAS. Building the approach that we use to develop of FPTAS, we also provide an integer programming formulation for CAP. In Section 8, we give computational experiments to check the value of customization on a dataset from Expedia. In Section 9, we check the practical performance of our approximation algorithms, as well as our integer programming formulation.

2 Problem formulation and complexity

We give a formulation of our model and characterize its complexity. We consider a set of products $\mathcal{N} = \{1, \dots, n\}$. For each product $i \in \mathcal{N}$, let r_i denote its revenue. Without loss of generality, we assume that the products are indexed such that

$$r_1 \geq r_2 \geq \ldots \geq r_n > 0.$$

We use $\mathcal{M} = \{1, \ldots, m\}$ to denote the set of customer types. The probability that a customer of type j arrives into the system is θ_j, where $\sum_{j \in \mathcal{M}} \theta_j = 1$. A customer of a certain type makes a choice among the products offered to her according to the multinomial logit (MNL) model. In the multinomial logit model, let v_{ij} denote the preference weight that customer type j attaches to product i. For all customer types, we normalize the preference weights of the no-purchase option to one. Under the MNL model, given that we offer the set of products S_j to a customer of type j, she purchases product $i \in S_j$ with probability $v_{ij}/(1 + \sum_{\ell \in S_j} v_{\ell j})$. In this case, given that we offer the set of products S_j to a customer of type j, the expected revenue that we obtain from the customer is $(\sum_{i \in S_j} r_i \, v_{ij})/(1 + \sum_{i \in S_j} v_{ij})$.

In the first stage of our problem, we select a subset of at most K products. We use $S \subseteq \mathcal{N}$ to capture this subset. In the second stage, we observe the type of the arriving customer and offer her a personalized assortment, which is a subset of the products in S carried initially. We use S_j to capture the personalized set of products offered to customer type j, where $S_j \subseteq S$. For each customer type $j \in \mathcal{M}$ and an initial subset of products $S \subseteq \mathcal{N}$, let $f_j(S)$ denote the optimal expected revenue from customer type j when the universe of products is S, i.e.,

$$f_j(S) = \max_{S_j \subseteq S} \frac{\sum_{i \in S_j} r_i \, v_{ij}}{1 + \sum_{i \in S_j} v_{ij}}.$$

Our goal is to find a set of at most K products to carry in the first stage to maximize the expected revenue over all customer types. We refer to this problem as the *Customized Assortment Problem* (CAP). In particular, we want to solve the problem

$$z_{\mathsf{CAP}} = \max_{S \subseteq \mathcal{N}, |S| \leq K} \sum_{j \in \mathcal{M}} \theta_j \, f_j(S). \tag{CAP}$$

For fixed S, computing $f_j(S)$ corresponds to finding an assortment $S_j \subseteq S$ that maximizes the expected revenue from a single customer of type j. This problem is well-studied (Gallego et al., 2004; Talluri and van Ryzin, 2004). Thus, for fixed S, we can efficiently compute $f_j(S)$.

In the next theorem, we provide an inapproximability result for CAP. In particular, we show that it is NP-hard to approximate CAP within a factor better than $(1 - \frac{1}{e} - \epsilon)$ for any $\epsilon > 0$ even in the special case where the product revenues are all equal and customers of all types arrive with equal probabilities. We use a reduction from the maximum coverage problem to show this inapproximability result. The proof of the theorem is deferred to Appendix A.

Theorem 2.1. *Unless $P = NP$, there is no polynomial-time algorithm that approximates* CAP *within a factor better than $(1 - \frac{1}{e} - \epsilon)$ for any $\epsilon > 0$ even when we have $r_i = 1$ for all $i \in \mathcal{N}$ and $\theta_j = \frac{1}{m}$ for all $j \in \mathcal{M}$.*

While the proof of the inapproximability result in Theorem 2.1 necessitates a large number of customer types, we also show a weaker hardness result for constant number of customer types. In particular, we show in Appendix A that CAP is NP-hard even with two customer types. Motivated by these inapproximability and hardness results, we will focus on developing approximation algorithms for CAP. Before presenting our algorithmic framework, we discuss in the next section the value of customization capability provided by our model.

3 Value of customization

We quantify the value of customization by comparing CAP to a model where the decision-maker offers the selected products in the first stage to all customer types without customization. In particular, to contrast CAP, which customizes the products selected in the first stage to each arriving customer, with a model that does not use customization, we write CAP as

$$z_{\mathsf{CAP}} = \max_{S \subseteq \mathcal{N},\, |S| \le K} \sum_{j \in \mathcal{M}} \theta_j \cdot \max_{S_j \subseteq S} \frac{\sum_{i \in S_j} r_i \, v_{ij}}{1 + \sum_{i \in S_j} v_{ij}}.$$

The problem without customization corresponds to the assortment optimization problem under a mixture of MNL models with a cardinality constraint. We refer to this problem as *Mixed Multinomial Logit* (MMNL) problem, which is given by

$$z_{\mathsf{MMNL}} = \max_{S \subseteq \mathcal{N},\, |S| \le K} \sum_{j \in \mathcal{M}} \theta_j \cdot \frac{\sum_{i \in S} r_i \, v_{ij}}{1 + \sum_{i \in S} v_{ij}}. \qquad (\mathsf{MMNL})$$

Note that CAP is a relaxation of MMNL, so the optimal expected revenue of CAP is at least as large as the optimal expected revenue of MMNL. In the next theorem, we show that the optimal expected revenue of CAP is at most m times the optimal expected revenue of MMNL. More importantly, we show that this bound is tight by presenting a family of instances where the optimal expected revenue of CAP can be $\Omega(m)$ times larger than the optimal expected revenue of MMNL. This result shows the power of customization, as the expected revenue of our model can be significantly larger when compared to a model without customization.

Theorem 3.1. *Let z_{MMNL} be the the optimal objective value of* MMNL *and z_{CAP} be the optimal objective value of* CAP*. Then, we have $z_{\mathsf{MMNL}} \le z_{\mathsf{CAP}} \le m \cdot z_{\mathsf{MMNL}}$. Moreover, there are instances such that $z_{\mathsf{CAP}} = \Omega(m) \cdot z_{\mathsf{MMNL}}$.*

Proof. The inequality $z_{\mathsf{MMNL}} \le z_{\mathsf{CAP}}$ is immediate because CAP is a relaxation of MMNL. To show the other inequality, let S^* be the optimal solution of CAP. For each $j \in \mathcal{M}$, let $S_j^* \subseteq S^*$ be the optimal assortment to offer to customer type j in CAP. Let $q \in \mathcal{M}$ be the customer type with the largest value of $\theta_j f_j(S^*)$ for $j \in \mathcal{M}$, i.e., $\theta_j f_j(S^*) \le \theta_q f_q(S^*)$ for all $j \in \mathcal{M}$. Therefore, $z_{\mathsf{CAP}} \le m \cdot \theta_q f_q(S^*)$. Since $S_q^* \subseteq S^*$, we have $|S_q^*| \le K$, so S_q^* is feasible for MMNL, yielding

$$\theta_q f_q(S^*) = \theta_q \cdot \frac{\sum_{i \in S_q^*} r_i \, v_{iq}}{1 + \sum_{i \in S_q^*} v_{iq}} \le \sum_{j \in \mathcal{M}} \theta_j \cdot \frac{\sum_{i \in S_q^*} r_i \, v_{ij}}{1 + \sum_{i \in S_q^*} v_{ij}} \le z_{\mathsf{MMNL}}.$$

Therefore, $z_{\mathsf{CAP}} \le m \cdot z_{\mathsf{MMNL}}$. To show the tightness of the bound, we give the following problem instance. We consider a problem instance where the number of products n is equal to the number of customer types m, which is, in turn, equal to the cardinality K, i.e., $n = m = K$. We use m to denote all three parameters. Let us consider the instance given by

$$\theta_j = \frac{\alpha}{a^j} \quad \forall\, j \in \{1,\dots,m\}, \qquad r_i = a^i \quad \forall\, i \in \{1,\dots,m\},$$
$$v_{ij} = \begin{cases} b^{m-i+1} & \text{if } i \le j \\ 0 & \text{otherwise} \end{cases} \quad \forall\, i,j \in \{1,\dots,m\},$$

where α is a normalizing constant, i.e., $\alpha = (\sum_{j=1}^m 1/a^j)^{-1}$. The scalars a and b will be chosen such that $a \gg b \gg 1$. We will shortly specify the exact values of a and b.

Since the cardinality of products K allowed in an assortment is equal to the number of products n, the optimal solution of CAP is $S^* = \mathcal{N}$. Moreover, the optimal assortment to offer to a customer type j in CAP is the solution of the assortment optimization problem under her MNL model with universe of products $\mathcal{N}$. Thus, we have

$$z_{\mathsf{CAP}} \ge \sum_{j=1}^m \theta_j \frac{r_j v_{jj}}{1+v_{jj}} = \alpha \sum_{j=1}^m \frac{b^{m-j+1}}{1+b^{m-j+1}} \ge \alpha \cdot \frac{m}{2},$$

where the first inequality holds because the optimal expected revenue of CAP is greater than the expected revenue of the solution where, for each $j \in \{1,\dots,m\}$, we only offer product j to customer type j. The second inequality holds because $b \ge 1$.

Now, consider a set of products S and fix $j \in \{1,\dots,m\}$. Let $\mathsf{Rev}_j(S)$ denote the expected revenue of assortment S under the MNL model of customer type j, i.e.,

$$\mathsf{Rev}_j(S) = \frac{\sum_{i\in S} r_i\, v_{ij}}{1+\sum_{i\in S} v_{ij}} = \frac{\sum_{i\in S} a^i \cdot \mathbf{1}(i \le j)\cdot b^{m-i+1}}{1+\sum_{i\in S}\mathbf{1}(i\le j)\cdot b^{m-i+1}}.$$

We have

$$\theta_j \cdot \mathsf{Rev}_j(S) = \theta_j \cdot \frac{\sum_{i\in S} a^i \cdot \mathbf{1}(i \le j-1)\cdot b^{m-i+1}}{1+\sum_{i\in S}\mathbf{1}(i\le j)\cdot b^{m-i+1}} + \alpha \cdot \frac{\mathbf{1}(j\in S)\cdot b^{m-j+1}}{1+\sum_{i\in S}\mathbf{1}(i\le j)\cdot b^{m-i+1}}. \tag{3.1}$$

We bound the first term on the right side of (3.1) as

$$\begin{aligned}\theta_j \cdot \frac{\sum_{i\in S} a^i \cdot \mathbf{1}(i \le j-1)\cdot b^{m-i+1}}{1+\sum_{i\in S}\mathbf{1}(i\le j)\cdot b^{m-i+1}} &\le \theta_j \sum_{i\in S} a^i \cdot \mathbf{1}(i\le j-1)\cdot b^{m-i+1} \\ &\le \theta_j \cdot b^m \sum_{i\in S} a^i \cdot \mathbf{1}(i \le j-1) \le \theta_j \cdot b^m \sum_{i=1}^{j-1} a^i \le \theta_j \cdot b^m \cdot 2a^{j-1} = 2\alpha b^m/a,\end{aligned}$$

where the last inequality holds because $\sum_{i=1}^{j-1} a^i = \frac{a^j-a}{a-1} \le 2a^{j-1}$ for $a \ge 2$. Letting ℓ_S be the smallest index in S, we bound the second term on the right side of (3.1) as

$$\alpha \cdot \frac{\mathbf{1}(j\in S)\cdot b^{m-j+1}}{1+\sum_{i\in S}\mathbf{1}(i\le j)\cdot b^{m-i+1}} \le \alpha \cdot \frac{\mathbf{1}(j\in S)\cdot b^{m-j+1}}{b^{m-\ell_S+1}} = \alpha \cdot \mathbf{1}(j \in S)\cdot b^{\ell_S - j}.$$

Moreover, we have

$$\sum_{j=1}^{m} \mathbf{1}(j \in S) \cdot b^{\ell_S - j} \leq 1 + \sum_{\ell_S < j \leq m} b^{\ell_S - j} \leq 1 + \frac{m-1}{b}.$$

Therefore, we obtain

$$\sum_{j=1}^{m} \theta_j \cdot \mathsf{Rev}_j(S) \leq \alpha \cdot \frac{2mb^m}{a} + \alpha \cdot \left(1 + \frac{m-1}{b}\right).$$

Choosing $b = m - 1$ and $a = 2m(m-1)^m$, we get $\sum_{j=1}^{m} \theta_j \cdot \mathsf{Rev}_j(S) \leq 3\alpha$ for any $S \subseteq \mathcal{N}$. Thus, we have $z_{\mathsf{MMNL}} \leq 3\alpha$. Noting that $z_{\mathsf{CAP}} \geq \alpha \cdot \frac{m}{2}$, we get $z_{\mathsf{CAP}} = \Omega(m) \cdot z_{\mathsf{MMNL}}$. □

4 Augmented Greedy algorithm

In view of the computational complexity of CAP discussed in Section 2, we focus on providing approximation algorithms for CAP. Our algorithmic framework is based on a novel algorithm that we design and refer to as Augmented Greedy. Before introducing our algorithm, we define the following notation. For any subset of customer types $\mathcal{C} \subseteq \mathcal{M}$, let

$$f^{\mathcal{C}} = \sum_{j \in \mathcal{C}} \theta_j f_j.$$

Thus, $f^{\mathcal{C}}(S)$ corresponds to the optimal expected revenue from customer types $\mathcal{C}$ given that we initially pick the subset of products S. Let us also define

$$f = f^{\mathcal{M}}.$$

In this case, CAP corresponds to the problem of maximizing the set function f subject to a cardinality constraint K, i.e.,

$$z_{\mathsf{CAP}} = \max_{S \subseteq \mathcal{N},\, |S| \leq K} f(S).$$

The function f is monotone increasing, i.e., $f(A) \leq f(B)$ for any $A \subseteq B \subseteq \mathcal{N}$, because we obtain larger expected revenue when we select more products in the first stage of CAP. However, the function f is not submodular[1]. Nemhauser et al. (1978) show that a greedy algorithm, which iteratively picks the element providing the largest increase in the objective function and adding it to the solution, gives $(1 - 1/e)$-approximation to the problem of maximizing monotone submodular functions subject to a cardinality constraint. Our function f is not submodular in general, so this classical result does not apply to our problem. In Appendix B, we give a counterexample with only one customer type and three products to show that the function f is not submodular. This observation necessitates developing a more general algorithm for approximating CAP.

[1]We say that a set function g is submodular on a finite set Ω if for every $A, B \subseteq \Omega$ with $A \subseteq B$ and every $i \in \Omega \setminus B$ we have that $g(A \cup \{i\}) - g(A) \geq g(B \cup \{i\}) - g(B)$.

4.1 Description of Augmented Greedy algorithm

We present our algorithm, referred to as Augmented Greedy, which will constitute the building block of our algorithmic framework for approximating CAP. The algorithm takes as input a subset of customer types $\mathcal{C} \subseteq \mathcal{M}$ and an integer k. The goal is to find an assortment Δ of size at most k that maximizes $f^{\mathcal{C}}(\Delta)$. In general, this problem cannot be approximated in polynomial time within a factor better than $(1 - 1/e)$ due to Theorem 2.1. Augmented Greedy returns a candidate assortment for this problem that we refer to as $\mathsf{AugGreedy}(\mathcal{C}, k)$. This candidate assortment verifies a key structural property that we will present in the next subsection.

We use the classical Greedy algorithm as a subroutine in our design of Augmented Greedy. Greedy takes as input a set function g defined over a set of products $\mathcal{P}$ along with a scalar k, and tries to find an assortment of size at most k that maximizes the set function g over the set of products $\mathcal{P}$. In particular, Greedy picks iteratively a product that provides the largest increase in the objective value until reaching the cardinality k. By the discussion at the beginning of this section, if g were monotone submodular, then Greedy would return a $(1 - 1/e)$-approximation to the problem $\max_{S \subseteq \mathcal{P}, |S| \leq k} g(S)$. Below, we state the details of Greedy for completeness.

Greedy

```
1: Input: set function $g$, products $\mathcal{P}$, cardinality $k$
2: $\Delta \leftarrow \varnothing$
3: while $|\Delta| < k$ and $\mathcal{P} \setminus \Delta \neq \varnothing$ do
4:     Add to $\Delta$ a product $i \in \mathcal{P} \setminus \Delta$ that maximizes $g(\Delta \cup i)$
5: end while
6: return $\Delta$
```

With the details of Greedy in place, we can describe Augmented Greedy. Recall that the products in $\mathcal{N}$ are indexed such that $r_1 \geq r_2 \geq \ldots \geq r_n$. We consider the products in descending order of revenues. For each $i \in \mathcal{N}$, let $\mathcal{V}_i$ be the subset of products with revenues at least as large as r_i, i.e., $\mathcal{V}_i = \{1, 2, \ldots, i\}$. For each $i \in \mathcal{N}$, we use Greedy to maximize the set function $S \to \sum_{j \in \mathcal{C}} \theta_j \min(f_j(S), r_i)$ over the set of products $\mathcal{V}_i$, subject to a cardinality constraint k. We let Δ_i be the assortment returned by Greedy. Augmented Greedy returns the assortment that maximizes $f^{\mathcal{C}}$ over all n candidates Δ_i for $i \in \mathcal{N}$. Here are the details of Augmented Greedy.

Augmented Greedy

```
1: Input: customer types $\mathcal{C} \subseteq \mathcal{M}$, cardinality $k$
2: for $i = 1, 2, \ldots, n$ do
3:     Let $\mathcal{V}_i = \{1, 2, \ldots, i\}$
4:     Use Greedy to get an approximate solution to

           $$\max_{S \subseteq \mathcal{V}_i, |S| \leq k} \sum_{j \in \mathcal{C}} \theta_j \min(f_j(S), r_i)$$

       Let $\Delta_i$ be the assortment returned by Greedy
5: end for
6: return $\mathsf{AugGreedy}(\mathcal{C}, k) = \operatorname{argmax}_{i \in \mathcal{N}} \{f^{\mathcal{C}}(\Delta_i)\}$
```

4.2 Lower bound on the performance of Augmented Greedy algorithm

In this subsection, we present our key structural result that gives a guarantee on the expected revenue of the assortment returned by Augmented Greedy. Consider a subset of customer types $\mathcal{C} \subseteq \mathcal{M}$ and an integer k. Let $\mathsf{AugGreedy}(\mathcal{C}, k)$ be the assortment returned by Augmented Greedy. We will show that this assortment provides a constant fraction of the optimal expected revenue from a certain subset of customer types in $\mathcal{C}$. To formally present our structural result, let us introduce some definitions. Let S^* be the optimal solution of CAP, and for each $j \in \mathcal{M}$, let $S_j^* \subseteq S^*$ be the optimal assortment to offer to customer type j in CAP.

Definition 4.1. Consider a subset of products $\mathcal{P} \subseteq \mathcal{N}$. We say that customer type j is *complete* with respect to $\mathcal{P}$ if and only if $\mathcal{P} \cap S_j^* = \mathcal{P} \cap S^*$.

For a subset of customer types $\mathcal{C} \subseteq \mathcal{M}$ and a subset of products $\mathcal{P} \subseteq \mathcal{N}$, we let $\mathcal{C}_\mathcal{P}$ denote the set of all customer types in $\mathcal{C}$ that are complete with respect to $\mathcal{P}$, i.e.,

$$\mathcal{C}_\mathcal{P} = \{j \in \mathcal{C} \ : \mathcal{P} \cap S_j^* = \mathcal{P} \cap S^*\}.$$

In the next theorem, we present our main result in this section. We show that the subset of products $\Delta = \mathsf{AugGreedy}(\mathcal{C}, k)$ returned by Augmented Greedy gives an expected revenue $f^{\mathcal{C}}(\Delta)$ that is at least a constant fraction of $f^{\mathcal{C}_\mathcal{P}}(\mathcal{P} \cap S^*)$ for any subset of products $\mathcal{P} \subseteq \mathcal{N}$ such that $|\mathcal{P} \cap S^*| \leq k$. Note that $f^{\mathcal{C}_\mathcal{P}}(\mathcal{P} \cap S^*)$ is the expected revenue that we obtain from the complete customer types $\mathcal{C}_\mathcal{P}$ with the set of products $\mathcal{P} \cap S^*$.

Theorem 4.2. *For any subset of customer types $\mathcal{C} \subseteq \mathcal{M}$ and any subset of products $\mathcal{P} \subseteq \mathcal{N}$, let $\mathcal{C}_\mathcal{P}$ be the set of complete customer types with respect to $\mathcal{P}$. Let $k \in \mathbb{N}$ be such that $|\mathcal{P} \cap S^*| \leq k$ and let $\Delta = \mathsf{AugGreedy}(\mathcal{C}, k)$. Then, we have*

$$f^{\mathcal{C}}(\Delta) \geq \left(1 - \frac{1}{e}\right) \cdot f^{\mathcal{C}_\mathcal{P}}(\mathcal{P} \cap S^*).$$

To prove Theorem 4.2, we will use a key lemma, where we show that, for any $i \in \mathcal{N}$ and $j \in \mathcal{C}$, the set function $S \to \min(f_j(S), r_i)$ is submodular on $\mathcal{V}_i$. Recall that we say that a set function g is submodular on Ω if for every $A, B \subseteq \Omega$ with $A \subseteq B$ and every $i \in \Omega \setminus B$ we have that $g(A \cup \{i\}) - g(A) \geq g(B \cup \{i\}) - g(B)$. By our counterexample in Appendix B, the set function $S \to f_j(S)$ is not necessarily submodular on $\mathcal{V}_i$. In contrast, the set function $S \to \min(f_j(S), r_i)$ turns out to be submodular on $\mathcal{V}_i$. Throughout the paper, we will often use the fact that f_j is monotone increasing and subadditive, i.e., $f_j(A) \leq f_j(B)$ for any $A \subseteq B \subseteq \mathcal{N}$ and $f_j(A \cup B) \leq f_j(A) + f_j(B)$ for any $A, B \subseteq \mathcal{N}$. These two properties are straightforward to show. For completeness, we provide their proofs in Appendix C.

Lemma 4.3. *For each $j \in \mathcal{C}$ and $i \in \mathcal{N}$, the set function $S \to \min(f_j(S), r_i)$ is submodular on $\mathcal{V}_i$.*

We give the proof of Lemma 4.3 in Appendix D. Building on this lemma, we present the proof of Theorem 4.2.

Proof of Theorem 4.2. Let p be the product with the smallest revenue in $\mathcal{P} \cap S^*$. Execute Augmented Greedy with inputs $\mathcal{C}$ and k. Consider the p-th iteration of Augmented Greedy, i.e., the iteration of the for loop corresponding to product p. In this iteration, we have $\mathcal{V}_p = \{1, \ldots, p\}$. Let $\Delta_p \subseteq \mathcal{V}_p$ be the assortment of products returned at this p-th iteration. Let us define the function h such that, for all $S \subseteq \mathcal{V}_p$, we have

$$h(S) = \sum_{j \in \mathcal{C}} \theta_j \min(f_j(S), r_p).$$

Multiplying submodular functions by positive constants and adding them up yields a submodular function. Thus, by Lemma 4.3, h is submodular on $\mathcal{V}_p$. Moreover, since f_j is monotone increasing, h is also monotone increasing. In this case, by Nemhauser et al. (1978), the greedy algorithm provides a $(1 - 1/e)$-approximate solution to the problem of maximizing h subject to a cardinality constraint k. Recall that Δ_p is the output of Greedy to the problem of maximizing h subject to a cardinally k. Furthermore, noting that $|\mathcal{P} \cap S^*| \leq k$, $\mathcal{P} \cap S^*$ is a feasible solution to the problem of maximizing h subject to a cardinality k, in which case, we obtain $h(\Delta_p) \geq (1 - \frac{1}{e}) \cdot h(\mathcal{P} \cap S^*)$. Therefore, we have

$$f^{\mathcal{C}}(\Delta) \geq f^{\mathcal{C}}(\Delta_p) = \sum_{j \in \mathcal{C}} \theta_j f_j(\Delta_p) \geq \sum_{j \in \mathcal{C}} \theta_j \min(f_j(\Delta_p), r_p) = h(\Delta_p) \geq \left(1 - \frac{1}{e}\right) \cdot h(\mathcal{P} \cap S^*).$$

For a complete customer type $j \in \mathcal{C}^{\mathcal{P}}$, we know that $\mathcal{P} \cap S^* = \mathcal{P} \cap S_j^*$, which implies that $p \in S_j^*$. By a standard result for assortment optimization under the MNL model, a product is in the revenue-maximizing assortment if and only if the revenue of the product is greater than or equal to the optimal expected revenue of the assortment optimization problem. For completeness, we prove this property in Appendix E. Thus, by this property, for all $j \in \mathcal{C}^{\mathcal{P}}$, we have

$$r_p \geq \frac{\sum_{i \in S_j^*} r_i v_{ij}}{1 + \sum_{i \in S_j^*} v_{ij}} = f_j(S^*) \geq f_j(\mathcal{P} \cap S^*),$$

where the equality follows by the definition of S_j^*. Therefore, we get

$$h(\mathcal{P} \cap S^*) = \sum_{j \in \mathcal{C}} \theta_j \min(f_j(\mathcal{P} \cap S^*), r_p) \geq \sum_{j \in \mathcal{C}^{\mathcal{P}}} \theta_j \min(f_j(\mathcal{P} \cap S^*), r_p) = \sum_{j \in \mathcal{C}^{\mathcal{P}}} \theta_j f_j(\mathcal{P} \cap S^*) = f^{\mathcal{C}^{\mathcal{P}}}(\mathcal{P} \cap S^*),$$

which concludes the proof. □

In Augmented Greedy, rather than trying to find a maximizer of the function $\sum_{j \in \mathcal{C}} f_j(S)$ over the ground set $\mathcal{V}_i$ subject to a cardinality constraint of k, we try to find a maximizer of the function $\sum_{j \in \mathcal{C}} \theta_j \min(f_j(S), r_i)$. Note that the advantage of working with the latter problem is that the function $S \to \sum_{j \in \mathcal{C}} \theta_j \min(f_j(S), r_i)$ is submodular on $\mathcal{V}_i$. However, due the fact that the latter problem tries to find a maximizer of $\sum_{j \in \mathcal{C}} \theta_j \min(f_j(S), r_i)$, rather than $\sum_{j \in \mathcal{C}} \theta_j f_j(S)$, Augmented Greedy recovers $1 - 1/e$ fraction of the optimal expected revenue only from the complete customer types, as indicated in Theorem 4.2. As a result, Theorem 4.2 does not immediately yield an approximation guarantee for CAP, but this theorem will form an important building block when we design an approximation algorithm for CAP.

5 Approximation algorithm

In this section, we give approximation algorithms for CAP. We start by considering the case where all customer types arrive with equal probability, and show that the output of Augmented Greedy is a $\Omega(1/m)$-approximate solution to CAP. Following this result, we consider the more general case where different customer types have different arrival probabilities. We build on our result with equal arrival probabilities to give a $\Omega(1/\log m)$-approximation algorithm to CAP.

5.1 Homogenous arrival probabilities

We consider the case where the arrival probabilities $\{\theta_j : j \in \mathcal{M}\}$ are equal, i.e., $\theta_j = \frac{1}{m}$ for all $j \in \mathcal{M}$. We show that if we execute Augmented Greedy with the input $\mathcal{M}$ for the customer types and K for the cardinality constraint, then we immediately get a $\Omega(1/\log m)$-approximation to CAP, despite the fact that Augmented Greedy with the input $\mathcal{M}$ for the set of customer types focuses on the objective function $\sum_{j\in\mathcal{M}} \theta_j \min(f_j(S), r_i)$ for each $i \in \mathcal{N}$.

Theorem 5.1. *Suppose that $\theta_j = \frac{1}{m}$ for all $j \in \mathcal{M}$. Let z_{CAP} be the optimal objective value of* CAP *and $\Delta = \mathsf{AugGreedy}(\mathcal{M}, K)$. Then, we have*

$$f(\Delta) = \Omega(1/\log m) \cdot z_{\mathsf{CAP}}.$$

Proof. Let S^* be the optimal solution of CAP. For each $j \in \mathcal{M}$, let $S_j^* \subseteq S^*$ be the optimal assortment to offer to customer type j in CAP. We use $\log x$ to denote the logarithm of x in base 2. We partition the customer types as follows. For each $\ell = 1, \ldots, \lceil \log m \rceil + 1$, we set

$$G_\ell = \left\{ j \in \mathcal{M} \;\middle|\; \frac{m \cdot z_{\mathsf{CAP}}}{2^\ell} < f_j(S^*) \le \frac{m \cdot z_{\mathsf{CAP}}}{2^{\ell-1}} \right\}.$$

To capture the remaining customer types, we set

$$G_L = \left\{ j \in \mathcal{M} \;\middle|\; f_j(S^*) \le \frac{m \cdot z_{\mathsf{CAP}}}{2^{\lceil \log m \rceil + 1}} \right\}.$$

Because $z_{\mathsf{CAP}} = \frac{1}{m}\sum_{j\in\mathcal{M}} f_j(S^*)$, we have $f_j(S^*) \le m \cdot z_{\mathsf{CAP}}$, which implies that the partitions $\{G_\ell : \ell = 1, \ldots, \lceil \log m \rceil + 1\}$ along with G_L collectively include all customer types. Noting the objective function of CAP, we have

$$z_{\mathsf{CAP}} = f(S^*) = \frac{1}{m} \cdot \sum_{\ell=1}^{\lceil \log m \rceil + 1} \sum_{j \in G_\ell} f_j(S^*) + \frac{1}{m} \cdot \sum_{j \in G_L} f_j(S^*).$$

Focusing on the customer types in G_L and using the fact that $|G_L| \le m$, we get

$$\frac{1}{m} \sum_{j\in G_L} f_j(S^*) \le \frac{1}{m} \sum_{j \in G_L} \frac{m \cdot z_{\mathsf{CAP}}}{2^{\lceil \log m \rceil + 1}} \le \frac{1}{m} \cdot |G_L| \cdot \frac{m \cdot z_{\mathsf{CAP}}}{2m} \le \frac{z_{\mathsf{CAP}}}{2}.$$

Because the total expected revenue from all customer types is z_{CAP}, we get

$$\frac{1}{m} \cdot \sum_{\ell=1}^{\lceil \log m \rceil + 1} \sum_{j \in G_\ell} f_j(S^*) \geq \frac{z_{\mathsf{CAP}}}{2},$$

so there exists a group of customers $\ell^* \in \{1, 2, \ldots, \lceil \log m \rceil + 1\}$ such that

$$\frac{1}{m} \sum_{j \in G_{\ell^*}} f_j(S^*) = \Omega(1/\log m) \cdot z_{\mathsf{CAP}}. \tag{5.1}$$

For notational brevity, let $\delta = \frac{m \cdot z_{\mathsf{CAP}}}{2^{\ell^*}}$ so that $\delta < f_j(S^*) \leq 2\delta$ for all $j \in G_{\ell^*}$. Let p be the product with the smallest revenue, but no smaller than δ, i.e.,

$$p = \arg\min_{i \in \mathcal{N}} \left\{ r_i \ : r_i \geq \delta \right\}.$$

When executing Augmented Greedy with inputs $\mathcal{M}$ and K, consider the p-th iteration of this algorithm, i.e., the iteration of the for loop corresponding to product p. In this iteration, we have $\mathcal{V}_p = \{1, \ldots, p\}$. Let $\Delta_p \subseteq \mathcal{V}_p$ be the solution returned at this p-th iteration. The solution Δ_p is obtained by using Greedy on the function $S \to \frac{1}{m} \sum_{j \in \mathcal{M}} \min(f_j(S), r_p)$ over the ground set $\mathcal{V}_p$ with cardinality constraint K. Because this function is submodular over $\mathcal{V}_p$ by Lemma 4.3, Δ_p is a $(1 - 1/e)$-approximate solution to the problem $\max_{S \subseteq \mathcal{V}_p, |S| \leq K} \frac{1}{m} \sum_{j \in \mathcal{M}} \min(f_j(S), r_p)$. On the other hand, since $|S^*| \leq K$, we have $|\mathcal{V}_p \cap S^*| \leq K$, so $\mathcal{V}_p \cap S^*$ is only a feasible solution to the last problem. In this case, we obtain

$$f(\Delta) \geq f(\Delta_p) = \frac{1}{m} \sum_{j \in \mathcal{M}} f_j(\Delta_p) \geq \frac{1}{m} \sum_{j \in \mathcal{M}} \min(f_j(\Delta_p), r_p) \geq \left(1 - \frac{1}{e}\right) \frac{1}{m} \sum_{j \in \mathcal{M}} \min(f_j(\mathcal{V}_p \cap S^*), r_p).$$

As discussed in the proof of Theorem 4.2, considering the assortment optimization problem under the MNL model, in Appendix E, we show that a product is in the revenue-maximizing assortment if and only if the revenue of the product is greater than or equal to the optimal expected revenue of the assortment optimization problem. Thus, we have $S_j^* = \{i \in S^* : r_i \geq f_j(S^*)\}$. Note that $f_j(S^*) \geq \delta$ for each $j \in G_{\ell^*}$, so $S_j^* \subseteq \{i \in \mathcal{N} : r_i \geq \delta\} = \mathcal{V}_p$, where the last equality follows from the definition of product p. Therefore, using the fact that $S_j^* \subseteq S^*$, we have $S_j^* \subseteq \mathcal{V}_p \cap S^*$, which implies that $f_j(\mathcal{V}_p \cap S^*) = f_j(S^*)$. Hence, the chain of inequalities above yields

$$f(\Delta) \geq \left(1 - \frac{1}{e}\right) \frac{1}{m} \sum_{j \in G_{\ell^*}} \min(f_j(S^*), r_p).$$

For each $j \in G_{\ell^*}$, we have $\min(f_j(S^*), r_p) \geq \delta \geq f_j(S^*)/2$. Thus, by the inequality above, we get $f(\Delta) \geq (1 - \frac{1}{e}) \frac{1}{m} \sum_{j \in G_{\ell^*}} f_j(S^*)/2 = \Omega(1/\log m) \cdot z_{\mathsf{CAP}}$, where the equality is by (5.1). $\square$

5.2 Heterogenous arrival probabilities

In this section, we present our main algorithm and show that it gives a $\Omega(1/\log m)$-approximation to CAP in the general case where different customer types have different arrival probabilities. Our approximation algorithm is based on using our Augmented Greedy algorithm several times, where at each time, we compute an assortment of a certain size for a certain subset of customer types, and then, we combine these assortments using a dynamic program that can be solved in polynomial time. More specifically, we define several groups of customer types based on the values of their arrival probabilities $\{\theta_j : j \in \mathcal{M}\}$. We use a tree structure to describe these groups. For each group of customer types and a given assortment size, we compute an assortment of products of this size using Augmented Greedy. Finally, we combine all the solutions together using a tractable dynamic program to get the assortment that gives $\Omega(1/\log m)$-approximation to CAP. The performance guarantee for the algorithm exploits Theorem 5.1, which gives a $\Omega(1/\log m)$-approximation to CAP in the case of equal arrival probabilities, as well as the structural result of Augmented Greedy in Theorem 4.2. Below, we describe the steps of our algorithm.

Step 1. (Round arrival probabilities) Consider $j \in \mathcal{M}$ and let $\ell \in \mathbb{Z}$ such that

$$\frac{1}{m^{\ell+1}} \leq \theta_j < \frac{1}{m^{\ell}}.$$

We round the arrival probablity θ_j to $\hat{\theta}_j = \frac{\lfloor \theta_j m^{\ell+1} \rfloor}{m^{\ell+1}}$. Because $\theta_j\, m^{\ell+1} \geq 1$, we have $\lfloor \theta_j\, m^{\ell+1} \rfloor \geq \frac{1}{2}\, \theta_j\, m^{\ell+1}$, so $\frac{\theta_j}{2} \leq \hat{\theta}_j \leq \theta_j$. Thus, we lose at most a factor 2 by this rounding. We will focus on giving a $\Omega(1/\log m)$-approximation algorithm for CAP with the arrival probabilities $\{\hat{\theta}_j : j \in \mathcal{M}\}$. For ease of notation, we simply use $\{\theta_j : j \in \mathcal{M}\}$ to denote the rounded values in the rest.

Step 2. (Construct a tree of customer types) For $\ell \in \mathbb{Z}$, we define the group of customer types

$$\mathcal{M}(\ell) = \left\{ j \in \mathcal{M} \;\middle|\; \frac{1}{m^{\ell+1}} \leq \theta_j < \frac{1}{m^{\ell}} \right\}.$$

The customer types $\mathcal{M}$ are given by the union of $\mathcal{M}(\ell)$ for $\ell \in \mathbb{Z}$. There are at most m of the subsets $\{\mathcal{M}(\ell) : \ell \in \mathbb{Z}\}$ that are non-empty because $|\mathcal{M}| = m$. We partition the customer types $\mathcal{M}$ into non-empty customer groups of the form $\{\mathcal{M}(\ell) : \ell \in \mathbb{Z}\}$, i.e.,

$$\mathcal{M} = G_1 \cup G_2 \cup \ldots \cup G_L,$$

where, for each $i \in \{1, \ldots, L\}$, we have $G_i = \mathcal{M}(\ell_i)$ for some $\ell_i \in \mathbb{Z}$ and G_i is non-empty. We index the customer groups such that $\ell_1 > \ell_2 > \ldots > \ell_L$. Note that, if $i < i'$ then $\ell_i > \ell_{i'}$, so for any $p \in G_i$ and any $q \in G_{i'}$, we have $\theta_p < \theta_q$. In particular, customer types in G_1 have the smallest values of θ_j and customer types in G_L have the highest values of θ_j among all customer types.

Based on this order, we construct a tree $\mathcal{T}$ as follows. At the root of $\mathcal{T}$, we split the customer types into two subsets, the left subtree contains $\{G_1, \ldots, G_{\lfloor L/2 \rfloor}\}$ and the right subtree contains $\{G_{\lfloor L/2 \rfloor+1}, \ldots, G_L\}$. Similarly, at each node, we keep splitting the group of customer types into two subsets such that half of the groups go the left subtree and the other half go to the right one until we arrive to the leaves, where each leaf of $\mathcal{T}$ contains a unique group. In Figure 1, we show the tree resulting from this procedure. Note that at each split, the customer types of the left subtree have smaller values of θ_j than the customer types of the right subtree. Therefore, the leaves of $\mathcal{T}$ from left to right are in the order $G_1, \ldots, G_L$. The depth of $\mathcal{T}$ is $O(\log L)$, which is at most $O(\log m)$, since the number of non-empty groups L is smaller than m. For each node in $\mathcal{T}$, we associate a

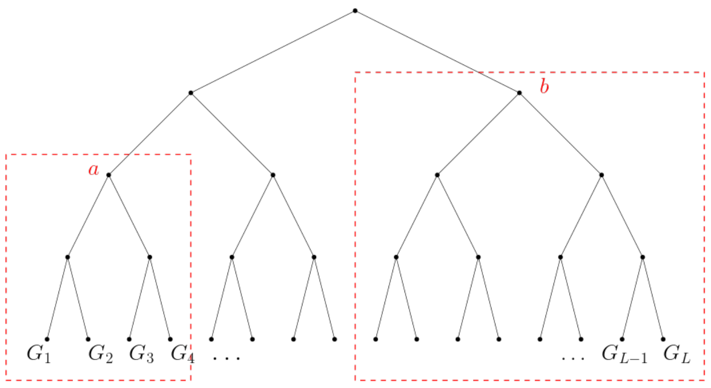

Figure 1: Tree of customer types $\mathcal{T}$

subtree that includes all descendant of that node as well as the node itself. Let $\mathcal{L}$ denote the set of all these subtrees. In Figure 1, the dashed boxes show two subtrees in $\mathcal{L}$ corresponding to nodes a and b. For each subtree $T \in \mathcal{L}$, let $\mathcal{C}_T$ denote the set of customer types that belong to the groups in the leaves of T. For instance, in Figure 1, if we denote by T the subtree that corresponds to node a, then $\mathcal{C}_T$ is the set of customer types that are in the groups $\{G_1, G_2, G_3, G_4\}$.

Step 3. (Execute Augmented Greedy) Recall that K is the the bound on the number of products that we can offer in CAP. For each subtree $T \in \mathcal{L}$ and each $k = 1, \ldots, K$, we use Augmented Greedy to compute the assortment

$$\Delta_{Tk} = \mathsf{AugGreedy}(\mathcal{C}_T, k),$$

and denote its corresponding expected revenue by

$$R_{Tk} = f^{\mathcal{C}_T}(\Delta_{Tk}).$$

Step 4. (Synthesize a solution) Finally, we solve the maximization problem

$$\begin{aligned} \max \quad & \sum_{T\in\mathcal{L}} \sum_{k=1}^{K} R_{Tk} \cdot x_{Tk} \\ \text{s.t.} \quad & \sum_{T\in\mathcal{L}} \sum_{k=1}^{K} k \cdot x_{Tk} \le K \\ & \sum_{k=1}^{K} (x_{Tk} + x_{T'k}) \le 1, \qquad \forall T \in \mathcal{L},\ T' \subseteq T,\ T' \neq T \\ & x_{Tk} \in \{0,1\}, \qquad \forall T \in \mathcal{L},\ k = 1, \ldots, K. \end{aligned} \tag{5.2}$$

In the second constraint, we write $T' \subseteq T$ if all the nodes of the subtree T' belong to the subtree T, i.e., the root of T' is a node in T. In problem (5.2), we choose a collection of subtrees to maximize the total expected revenue. The first constraint ensures that the total number of products in the assortment Δ_{Tk} for the chosen subtrees does not exceed K. The second constraint ensures that the chosen subtrees are disjoint. Furthermore, for each chosen subtree T, there exists only one capacity value k such that $x_{Tk} = 1$. We can solve problem (5.2) efficiently by using a dynamic program.

Dynamic programming formulation for problem (5.2)**.** Let $g(T, k)$ be the optimal objective value of problem (5.2) when we can choose only assortments $\Delta_{T'k}$ for the subtrees $T' \subseteq T$ and the total number of products in the chosen assortments cannot exceed k. We compute the value functions $\{g(T, k) : T \in \mathcal{L},\ k = 1, \dots, K\}$ as follows. In the boundary condition, if T is a leaf, then we set $g(T, k) = R_{Tk}$. Otherwise, we set

$$g(T, k) = \max\left\{ R_{Tk}, \max_{k_1+k_2=k} \{g(T_1, k_1) + g(T_2, k_2)\} \right\},$$

where T_1 and T_2 are respectively the left and right subtrees of T. The optimal objective value of problem (5.2) is given by $g(\mathcal{T}, K)$.

Letting $\{x^*_{Tk} : T \in \mathcal{L},\ k = 1, \dots, K\}$ be the optimal solution of problem (5.2), our candidate solution for CAP is

$$\Delta = \left\{ \bigcup_{T\in\mathcal{L},\, k=1,\dots,K} \Delta_{Tk} \ :\ x^*_{Tk} = 1 \right\}. \tag{5.3}$$

In the next theorem, we show that the solution in (5.3) is a $\Omega(1/\log m)$-approximation to CAP.

Theorem 5.2. *Let z_{CAP} be the optimal objective value of* CAP *and Δ be the assortment in* (5.3)*. Then, we have*

$$f(\Delta) = \Omega(1/\log m) \cdot z_{\mathsf{CAP}}.$$

We give the proof of Theorem 5.2 in Appendix F. The outline of the proof is as follows. First, the customer types located at a single leaf of the tree have the same arrival probabilities, so we use an argument similar to the one in the proof of Theorem 5.1 to show that Augmented Greedy, when applied to the these customer types, obtains $\Omega(1/\log m)$ fraction of the optimal expected revenue from these customer types. Second, using Theorem 4.2, we show that instead of offering some subset of products to the customers in a tree, we can partition the subset of products into two and offer the products in each partition separately to the customers in the left and right subtrees, while incurring a loss in the expected revenue of $1 - 1/\log m$. Because the depth of the tree is at most $\log m$, the total loss in this partitioning scheme is bounded by $(1 - \frac{1}{\log m})^{\log m}$ which is equivalent to $1/e$, for large m. Joining the two parts of the argument ultimately gives the desired guarantee.

6 Fully polynomial-time approximation scheme

We develop an FPTAS for CAP for the case where the number of customer types m is constant. In particular, for any desired accuracy $\delta > 0$, we design a polynomial-time algorithm that outputs an assortment S such that $f(S) \geq (1 - \delta) \cdot z_{\mathsf{CAP}}$. The idea behind our FPTAS is to guess the optimal expected revenue from each customer type. Once we guess the optimal expected revenue from each customer type, we solve a dynamic program to check whether there exists an assortment to pick in the first stage that allow us to achieve the guesses for the optimal expected revenue from each customer type. By trying guesses over a geometric grid and choosing the best achievable guess, we obtain an approximate solution with the desired accuracy.

Let S be the assortment that we pick in the first stage in CAP. There exists an assortment $S_j \subseteq S$ that provides an expected revenue of t_j or more from customer type j if and only if there exists an assortment $S_j \subseteq S$ such that

$$\frac{\sum_{i \in S_j} r_i \, v_{ij}}{1 + \sum_{i \in S_j} v_{ij}} \geq t_j.$$

The inequality above is equivalent to $\sum_{i \in S_j} v_{ij} \, (r_i - t_j) \geq t_j$. Thus, there exists an assortment $S_j \subseteq S$ that provides an expected revenue of t_j or more from customer type j if and only if there exists an assortment $S_j \subseteq S$ such that $\sum_{i \in S_j} v_{ij} \, (r_i - t_j) \geq t_j$. There exists an assortment $S_j \subseteq S$ that satisfies the last inequality if and only if $\max_{S_j \subseteq S} \{ \sum_{i \in S_j} v_{ij} \, (r_i - t_j) \} \geq t_j$. In the last maximization problem, it is optimal to include each product $i \in S$ in the solution S_j if $r_i - t_j \geq 0$. Thus, the optimal objective value of the last maximization problem is $\sum_{i \in S} v_{ij} \, (r_i - t_j)^+$. Thus, the discussion in this paragraph establishes that if we pick the assortment S in the first stage in CAP, then we can obtain an expected revenue of t_j or more from customer type j if and only if

$$\sum_{i \in S} v_{ij} \, (r_i - t_j)^+ \geq t_j.$$

Thus, by the discussion so far in this section, for each customer type j, we have $f_j(S) \geq t_j$ if and only if $\sum_{i \in S} v_{ij} \, (r_i - t_j)^+ \geq t_j$. Using this observation, our FPTAS is based on an alternative formulation for CAP. In the alternative formulation, we guess lower bounds on the optimal expected revenue from each customer type. Using these lower bounds on the optimal expected revenue for each customer type, we obtain a lower bound on the optimal objective value of CAP. In our alternative formulation, we maximize this latter lower bound on the optimal objective value of CAP. In particular, let $\boldsymbol{t} = (t_1, \ldots, t_m)$ be the vector of lower bound guesses for the optimal expected revenue from each of the m customer types. Using $S \subseteq \mathcal{N}$ to denote the assortment that we pick in the first stage in CAP, consider the problem

$$\begin{aligned} \max \quad & \sum_{j \in \mathcal{M}} \theta_j \, t_j \\ \text{s.t.} \quad & \sum_{i \in S} v_{ij} \, (r_i - t_j)^+ \geq t_j \qquad \forall \, j \in \mathcal{M} \\ & |S| \leq K \\ & S \subseteq \mathcal{N}, \; \boldsymbol{t} \in \mathbb{R}^m_+, \end{aligned} \tag{6.1}$$

where the decision variables are S and $\boldsymbol{t} = (t_1, \ldots, t_m)$. The first constraint ensures that if we pick the assortment S in the first stage, then the optimal total expected revenue from customer type j is lower bounded by t_j. The second constraint ensures that the assortment that we pick in the first stage includes no more than K products. In the objective function, $\sum_{j \in \mathcal{M}} \theta_j \, t_j$ is a lower bound on the optimal expected revenue from all customer types and we maximize this lower bound. We can show that if $(S^*, \boldsymbol{t}^*)$ is an optimal solution to problem (6.1), then S^* is also an optimal solution to CAP. We do not pursue showing this result, because CAP is NP-hard, which implies the same for the problem above. We focus on approximating problem (6.1).

Our approximate version of problem (6.1) is based on two observations. First, instead of using lower bounded guesses on the optimal expected revenues that take values over real numbers, we use guesses that take values over a geometric grid. To construct our geometric grid, let $r_{\max} =$

$\max\{r_i : i \in \mathcal{N}\}$ and $r_{\min} = \min\{r_i : i \in \mathcal{N}\}$ be the largest and smallest product revenues and $v_{\min} = \min\{v_{ij} : i \in \mathcal{N},\ j \in \mathcal{M},\ v_{ij} > 0\}$ be the smallest non-zero preference weight. In this case, the expected revenue from any customer type lies in $[r_{\min} \frac{v_{\min}}{1+v_{\min}}, r_{\max}] \cup \{0\}$. Letting $B_{\min} = r_{\min} \frac{v_{\min}}{1+v_{\min}}$ and $B_{\max} = r_{\max}$ for notational brevity, for a fixed accuracy parameter $\epsilon > 0$, we consider the grid points on the geometric grid

$$\mathsf{Grid} = \left\{(1+\epsilon)^k : k = \left\lfloor \frac{\log B_{\min}}{\log(1+\epsilon)} \right\rfloor, \ldots, \left\lceil \frac{\log B_{\max}}{\log(1+\epsilon)} \right\rceil \right\} \cup \{0\}.$$

Second, we modify the first constraint in problem (6.1) so that we can solve this problem through a dynamic program. We write the first constraint equivalently as $\sum_{i \in S} \frac{v_{ij}\,(r_i - t_j)^+}{t_j} \cdot \frac{n}{\epsilon} \geq \frac{n}{\epsilon}$. We consider an approximate version of the constraint given by

$$\sum_{i \in S} \left\lceil \frac{v_{ij}\,(r_i - t_j)^+}{t_j} \cdot \frac{n}{\epsilon} \right\rceil \geq \left\lfloor \frac{n}{\epsilon} \right\rfloor.$$

The constraint above is a relaxation of the first constraint in problem (6.1), because any (S, t_j) pair that satisfies the first constraint in problem (6.1) also satisfies the constraint above. Letting $\sigma_{ij}(t_j) = \lceil \frac{v_{ij}\,(r_i - t_j)^+}{t_j} \cdot \frac{n}{\epsilon} \rceil$ for notational brevity, we write the constraint above as $\sum_{i \in S} \sigma_{ij}(t_j) \geq \lfloor \frac{n}{\epsilon} \rfloor$. Note that $\sigma_{ij}(t_j)$ takes on integer values.

Replacing the first constraint in problem (6.1) with $\sum_{i \in S} \sigma_{ij}(t_j) \geq \lfloor \frac{n}{\epsilon} \rfloor$ and focusing on the lower bound guesses $\boldsymbol{t} = (t_1, \ldots, t_m)$ that take values in Grid^m, we consider an approximate version of problem (6.1) given by

$$\begin{aligned} \max \quad & \sum_{j \in \mathcal{M}} \theta_j\, t_j \\ \text{s.t.} \quad & \sum_{i \in S} \sigma_{ij}(t_j) \geq \left\lfloor \frac{n}{\epsilon} \right\rfloor \qquad \forall\, j \in \mathcal{M} \\ & |S| \leq K \\ & S \subseteq \mathcal{N},\ \boldsymbol{t} \in \mathsf{Grid}^m. \end{aligned} \tag{6.2}$$

Our FPTAS is based on solving the problem above by using a dynamic program. For each set of lower bound guesses $\boldsymbol{t}$, we can check whether there exists an assortment $S \subseteq \mathcal{N}$ that satisfies the first two constraints in problem (6.2) by solving the dynamic program

$$V_i(q_1, \ldots, q_m, \ell; \boldsymbol{t}) = \max\Big\{ V_{i+1}(q_1 + \sigma_{i1}(t_1), \ldots, q_m + \sigma_{im}(t_m), \ell + 1; \boldsymbol{t}), V_{i+1}(q_1, \ldots, q_m, \ell; \boldsymbol{t}) \Big\}, \tag{6.3}$$

with the boundary condition that

$$V_{n+1}(q_1, \ldots, q_m, \ell; \boldsymbol{t}) = \begin{cases} 0 & \text{if } \ell \leq K \text{ and } q_j \geq \left\lfloor \frac{n}{\epsilon} \right\rfloor \text{ for all } j \in \mathcal{M} \\ -\infty & \text{otherwise.} \end{cases} \tag{6.4}$$

The two terms in the max operator in (6.3) correspond to including and not including product i in the assortment S in problem (6.2). The state variable q_j accumulates the quantities $\{\sigma_{ij}(t_j) : i \in \mathcal{N}\}$ for the products included in the assortment S, whereas the state variable ℓ tracks the number of products in the assortment S. In the boundary condition, if $q_j \geq \lfloor \frac{n}{\epsilon} \rfloor$ for all $j \in \mathcal{M}$ and $\ell \leq K$, then there exists an assortment S with $|S| \leq K$ such that $\sum_{i \in \mathcal{N}} \sigma_{ij}(t_j) \geq \lfloor \frac{n}{\epsilon} \rfloor$. Noting that $\sigma_{ij}(t_j)$ is an integer, all components of the state variable take on integer values.

Observe that there are $O\big(\frac{\log(B_{\max}/B_{\min})}{\epsilon}\big)$ points in Grid, yielding a total of $O\big(\frac{\log^m(B_{\max}/B_{\min})}{\epsilon^m}\big)$ lower bound guesses $\boldsymbol{t} \in \mathsf{Grid}^m$. In the next theorem, accounting for the running time to solve the dynamic program in (6.3) for each $\boldsymbol{t} \in \mathsf{Grid}^m$ so that we can check whether there exists an assortment S that satisfies the two constraints in problem (6.2), along with tracking the error introduced by using a geometric grid for the lower bound guesses, we give an FPTAS for CAP for constant m. We give the proof of the theorem in Appendix G.

Theorem 6.1. *There exists an algorithm, where, for any $\delta \in (0,1)$, the algorithm returns a $(1-\delta)$-approximate solution to CAP in running time $O\big(\log^m(\frac{B_{\max}}{B_{\min}})\,\frac{n^{m+2}}{\delta^{2m}}\big)$, which is polynomial in the input size and δ for constant m.*

Another useful aspect of our FPTAS is that we can build on the approach for developing our FPTAS to give an integer programming formulation for CAP. In particular, just like our FPTAS is motivated by guessing a lower bound t_j for the optimal expected revenue from each customer type j, we can give an integer programming formulation that is based on guessing such a lower bound. The running time for our FPTAS can get excessive when the number of customer types is large. The integer programming formulation we give can be interpreted as a practically appealing version of our FPTAS. Moreover, the integer programming formulation will serve as a benchmark to evaluate numerically the performance of our algorithms.

Integer programming formulation. In our integer programming formulation, we build on problem (6.1), but focus on the lower bound guesses that take place over the geometric grid. For notational brevity, we denote the elements of the grid by $\{\tau^1, \ldots, \tau^L\}$ with $L = |\mathsf{Grid}|$. We have two sets of binary decision variables. The decision variable x_i takes value 1 if we pick product i to include in the assortment in the first stage of CAP, whereas the decision variable $z_{j\ell}$ takes value 1 if we chose the ℓ-th point in Grid as the lower bound on the optimal expected revenue from customer type j. In this case, we consider the integer program

$$\begin{aligned}
\max \quad & \sum_{j \in \mathcal{M}} \theta_j \sum_{\ell=1}^{L} \tau^\ell \, z_{j\ell} && (6.5)\\
\text{s.t.} \quad & \sum_{i \in \mathcal{N}} v_{ij}\,(r_i - \tau^\ell)^+ \, x_i \geq \tau^\ell \, z_{j\ell} \qquad \forall\, j \in \mathcal{M},\ \ell = 1, \ldots, L \\
& \sum_{i \in \mathcal{N}} x_i \leq K \\
& \sum_{\ell=1}^{L} z_{j\ell} = 1 \qquad \forall\, j \in \mathcal{M} \\
& x_i,\ z_{j\ell} \in \{0,1\} \qquad \forall\, i \in \mathcal{N},\ j \in \mathcal{M},\ \ell = 1, \ldots, L.
\end{aligned}$$

In the problem above, the first two constraints are analogues of the first two constraints in problem (6.1). The right side of the first constraint above takes value τ^ℓ if we choose the ℓ-th point in Grid as the lower bound on the optimal expected revenue from customer type j. In this case, if the assortment that we pick in the first stage of CAP is characterized by the decision variables $\{x_i : i \in \mathcal{N}\}$, then we know that the optimal expected revenue from customer type j is lower bounded by τ^ℓ if and only if $\sum_{i\in\mathcal{N}} v_{ij}\,(r_i - \tau^\ell)^+\, x_i \geq \tau^\ell$, which is the condition imposed by the first constraint. The second constraint limits the size of the assortment that we pick in the first stage of CAP to K. The third constraint ensures that we pick one lower bound on the optimal expected revenue from each customer type. In the objective function, we maximize the lower bound on the optimal expected revenue over all customer types.

The first constraint in (6.5) is driven by the same idea we used in our FPTAS. In Appendix H, noting that ϵ is the accuracy parameter for Grid, we show that if $\{x_i^* : i \in \mathcal{N}\}$ is an optimal solution to problem (6.5), then the assortment $S^* = \{i \in \mathcal{N} : x_i^* = 1\}$ is a $(1-\epsilon)$-approximation to CAP and we can obtain an upper bound on the optimal objective value of CAP by using problem (6.5). Our integer programming formulation returns a $(1-\epsilon)$-approximate solution, but one advantage of our formulation is that existing integer programming formulations for assortment optimization problems under the MNL model with multiple customer types usually require big-M constraints (Bront et al., 2009). In contrast, problem (6.5) does not require such constraints.

7 Extension to cardinality constraints in the second stage

We can consider a variety of constraints on the assortments offered in the first and second stages of CAP. In our original formulation, we used a cardinality constraint on the assortment offered in the first stage. One can also consider constraints on the assortment offered to each customer type in the second stage. Our $\Omega(1/\log m)$-approximation exploits the properties of the unconstrained assortment optimization problem with a single customer type under the MNL model, as discussed in Appendix E. We lose these properties when there is a constraint on the assortment offered to each customer type in the second stage. However, considering the case where we can include at most K products in the assortment offered in the first stage assortment and p products in assortment offered to each customer type in the second stage, we can give a $\Omega(1/\sqrt{m \log m})$-approximation. In particular, we have the following theorem.

Theorem 7.1. *There exists a $\Omega(1/\sqrt{m\log m})$-approximation to CAP with cardinality constraints in the second stage.*

Proof. Throughout the proof, we assume that $p \leq K < mp$. In particular, if $p \geq K$, then the cardinality constraint in the second stage is redundant. On the other hand, if $K \geq mp$, then we can focus on each customer type separately and find an assortment with no more than p products to maximize the expected revenue from each customer type. In this case, the union of these assortments would include at most mp products, but since $K \geq mp$, we can offer this union in the first stage. Assuming that $p \leq K < mp$, we can proceed as follows to get a $\Omega(1/\sqrt{m\log m})$-approximation. Using the subadditivity property of the expected revenue function discussed in Appendix C, we can show that if we can offer at most $K' \leq K$ products in the first stage, instead of K, then the optimal expected revenue of CAP changes by at most a factor of $\Omega(K'/K)$. With this observation in place, we construct two solutions. First, we solve CAP with a cardinality constraint of p, instead of K in the first stage. By doing so, the loss in the optimal expected revenue is by a factor of $\Omega(p/K)$, but since we can solve CAP only with $\Omega(1/\log m)$ guarantee, we obtain a $\Omega(p/(K\log m))$-approximate solution. Second, we focus on each customer type one by one and find an assortment with no more than p products that maximizes the expected revenue from each customer type. We can find these assortments in polynomial time by maximizing the expected revenue from a single customer type under a cardinality constraint (Rusmevichientong et al., 2010). The union of these assortments would include $O(mp)$ products. By using the subadditivity property of the expected revenue function, we can split this assortment into $O(mp/K)$ assortments each with at most K products, so that the sum of the expected revenues from the $O(mp/K)$ assortments is at least as large as the optimal objective value of CAP, which means that one of the $O(mp/K)$ assortments is a $\Omega(K/(mp))$-approximation to CAP. Considering the two solutions that we construct, we get an approximation guarantee of

$$\Omega\left(\max\left\{\frac{p}{K}\cdot\frac{1}{\log m},\frac{K}{p}\cdot\frac{1}{m}\right\}\right).$$

For $a \geq 0$, the minimum of $\max\{a\frac{1}{\log m}, \frac{1}{a}\frac{1}{m}\}$ occurs at $a = \sqrt{\log m/m}$. Thus, the expression above is at least $\Omega(1/\sqrt{m\log m})$. $\square$

The approximation guarantee of $\Omega(1/\sqrt{m\log m})$ is better than $O(1/m^{1-\epsilon})$ for $\epsilon > 0$, which is the best possible approximation without customization. We were able to get this guarantee by exploiting the results that we already have, but it may be possible to use an entirely new argument to give an even better approximation guarantee for CAP under a cardinality constraint for the second stage assortment.

8 Computational study: Value of customization

We use a dataset from Expedia, as well as synthetically generated datasets, to demonstrate the value of customization, when compared to offering the same assortment to all customers.

8.1 Expedia data

We use a dataset provided by Expedia as a part of a Kaggle competition (Kaggle, 2013).

Description of the dataset. The dataset gives the results of search queries for hotels on Expedia. The rows of the dataset correspond to different hotels that are displayed in different search queries by different customers. The columns give information on the characteristics of the displayed hotels in a search query, the characteristics of the customer making the search query, and the booking decision of the customer. In Table 1, we demonstrate the structure of the dataset by giving a small excerpt that includes five rows from the dataset. There are two queries in the excerpt. The customer making the first booking query was shown three hotels, whereas the customer making the second booking query was shown two hotels. Each row in the data corresponds to a displayed hotel in a search query. The columns in the dataset have the following interpretation: The first column is the unique code of the search query in which the hotel was displayed. Using this column, we have access to all the displayed hotels in a search query, providing the set of hotels among which the customer needed to make a choice. The second column gives the name of the hotel, but we do not use this information. The following eight columns show the attributes of the displayed hotel: Star rating, review score, an indicator showing whether the hotel is a part of a chain, location score, accessibility score, average historical price, displayed price for the current customer, and an

Qry. No	Hotel Name	Star	Rvw. Scr.	Chain?	Loc. Scor.	Acc. Scor.	Past Price	Curr. Price	Prom?	Early?	No Adlt.	No Chld.	Sat?	Booked?
1	SGE	4	5.8	Y	8.3	6.2	140	116	Y	Y	2	2	Y	N
1	PSC	5	9.0	N	7.8	9.9	146	179	N	Y	2	2	Y	Y
1	MQU	3	7.3	N	5.2	6.6	253	223	N	Y	2	2	Y	N
2	UCA	5	6.6	N	9.6	6.4	286	270	Y	N	1	0	N	N
2	OKB	2	9.9	N	8.0	5.8	228	245	N	N	1	0	N	N

Table 1: Excerpt from the Expedia dataset.

indicator showing whether the hotel was on promotion. The last four columns give information about the characteristics of the customer making the search query: An indicator showing whether the customer booked early, the number of adults in the intended booking, the number of children in the intended booking, and lastly, an indicator of whether the customer is making a Saturday night booking. These four columns will be useful to define our customer types. A booking is classified as an early booking as long as it is made more than a week before the night of stay. The last column in the dataset is an indicator of whether the customer booked the hotel in the search query. This corresponds to the purchase decision of the customer. A customer can book at most one hotel or leave without making any booking. For example, the first customer in the table booked the second displayed hotel, whereas the second customer did not book any hotels.

We preprocessed the dataset to remove missing and uninterpretable values by using the same approach in Gao et al. (2020). For booking queries for multiple nights, the price was sometimes quoted for the whole duration of the stay and was sometimes quoted on a per night basis. It was not possible to distinguish in the dataset which approach is used for which booking queries. To avoid ambiguity, we dropped all booking queries for multiple nights of stay, focusing only on single-night queries. The resulting dataset contains 595,965 rows, representing 34,561 queries.

Customer types and preference weights. We define 16 types of customers based on the following four criteria: Whether the customer makes an early booking request, whether the customer makes a booking request for a single adult, whether the customer makes a booking request that includes children, and lastly, whether the customer makes a booking request for a Saturday night stay. We use the 11-th to 14-th columns of the dataset to identify the type of the customer making the booking query. These four binary criteria give us $2^4 = 16$ customer types.

The customer did not make a booking in 83% of the search queries in the dataset. To enrich our experiments, we generate three datasets with different fractions of the no-purchase outcome by randomly dropping some rows that correspond to a no-purchase. In particular, we construct a first dataset with 30% of the queries resulting in a no-purchase, a second dataset with 50% of the queries resulting in a no-purchase, and a third one with 70% of the queries resulting in a no-purchase. For each dataset, we use the eight features of the displayed hotels to estimate an MNL model separately for each customer type. We fit the MNL model for each customer type using maximum likelihood estimation. The preference weight a customer type of j associates with hotel i has the form $v_{ij} = \exp(\beta_{0j} + \sum_{\ell=1}^{8} \beta_{\ell j} x_{i\ell})$, where $x_{i\ell}$ is the value of the ℓ-th attribute of hotel i and the parameters $\beta_{\ell j}$ for $\ell = 0, 1, \ldots, 8$ are estimated parameters for customer type j.

Considering the dataset with 30% of the search queries resulting in a no-purchase, in Table 2, we show an example of the estimated parameters of the MNL model for customer type 3. Customer type 3 corresponds to customers making an early booking query with more than one adult and no children for a non-Saturday night stay. We had 1095 customers of this type making booking queries in our dataset. For instance, we observe from the table that the coefficient for the review score is

	Star	Rvw. Scr.	Chain?	Loc. Scor.	Acc. Scor.	Past Price	Curr. Price	Prom?
β_{0j}	β_{1j}	β_{2j}	β_{3j}	β_{4j}	β_{5j}	β_{6j}	β_{7j}	β_{8j}
-2.45	0.49	0.15	0.01	-0.26	0.60	-0.05	-1.69	0.12

Table 2: Example of fitted MNL parameters for customer type 3.

0.15 which means that higher review score positively affects the preference weight that customer type 3 attaches to a hotel, while the coefficient for the displayed price is -1.69, which implies that higher price negatively affects the preference weight.

Experimental setup. To demonstrate the value of customization, we compare the expected revenues obtained in the following two situations: (i) The platform identifies the type of the customer and makes a customized assortment offer accordingly, (ii) the platform offers the same assortment to all customers without paying attention to the customer type. By the discussion in the previous two paragraphs, we can estimate the preference weight v_{ij} that a customer of each type j attaches to each hotel i. We do not have access to the universe of hotels that we could potentially show to the customer, so we consider the set of hotels that are displayed to a customer in her search query as our universe. Let U_t be the set of hotels displayed to the customer making the t-th search query in the dataset and j_t be the type of the customer making the t-th search query. The maximum and average size of U_t in our data are, respectively, 34 and 17. If the platform identifies the type of the customer making the search query and makes a customized assortment offer, then the assortment that maximizes the expected revenue from the customer making the t-th query is

$$S_t^{\mathsf{cust}} = \arg\max_{S \subseteq U_t} \frac{\sum_{i \in S} r_i \, v_{i,j_t}}{1 + \sum_{i \in S} v_{i,j_t}},$$

where r_i is the price of hotel $i \in U_t$ in the t-th search query. Here, S_t^{cust} stands for the optimal customized assortment for the customer making the t-th query.

On the other hand, if the platform offers the same assortment to all customers without paying attention to the customer type, then it can maximize the expected revenue from a customer visit by solving an assortment optimization problem under a mixture of multinomial logit models. In this problem, a customer making a search query is of type j with probability θ_j, where θ_j is the fraction of type j customers in the dataset. Thus, the non-customized assortment that we offer to the customer making the t-th search query is obtained by solving the problem

$$S_t^{\mathsf{non\text{-}cust}} = \arg\max_{S \subseteq U_t} \sum_{j \in \mathcal{M}} \theta_j \frac{\sum_{i \in S} r_i \, v_{ij}}{1 + \sum_{i \in S} v_{ij}},$$

where $\mathcal{M}$ corresponds to the set of 16 customer types in our dataset. Here, $S_t^{\mathsf{non\text{-}cust}}$ stands for the optimal non-customized assortment for the customer making the t-th search query. We can formulate the problem above as an integer program with big-M constraints (Méndez-Díaz et al., 2014). We use $\mathsf{Rev}_j(S) = \frac{\sum_{i \in S} r_i \, v_{ij}}{1+\sum_{i \in S} v_{ij}}$ to denote the expected revenue that we obtain from a customer of type j when we offer the assortment S to this customer. In this case, the percent improvement in the expected revenue by offering a customized assortment to the customer making the t-th search

Cust. No	Queries	γ_t (%) Mean	95%	Max
1	354	0.11	0.46	5.26
2	499	2.08	7.17	24.75
3	1095	0.73	2.59	12.65
4	815	0.10	0.47	4.67
5	137	1.93	7.82	32.58
6	117	4.52	15.91	52.57
7	350	0.62	2.49	24.20
8	170	1.27	5.09	34.62
9	324	1.16	5.42	22.17
10	405	1.03	4.01	11.16
11	1667	0.24	1.12	7.43
12	1319	0.08	0.39	6.58
13	151	1.20	4.09	8.97
14	143	3.55	14.07	51.57
15	459	0.96	4.49	25.62
16	349	0.30	1.10	35.42

Table 3: Value of customization in the Expedia dataset with 30% fraction of no-purchase queries.

query, as opposed to a non-customized assortment, is given by

$$\gamma_t = 100 \cdot \frac{\mathsf{Rev}_{j_t}(S_t^{\mathsf{cust}}) - \mathsf{Rev}_{j_t}(S_t^{\mathsf{non\text{-}cust}})}{\mathsf{Rev}_{j_t}(S_t^{\mathsf{non\text{-}cust}})}.$$

We group the queries in our datasets by customer types. For each customer type and each dataset, we report the mean, 95-th percentile and maximum value of γ_t over all search queries made by that customer type. Our results are given in Tables 3, 4 and 5. Each table corresponds to one of our three datasets with different fractions of no-purchase outcome. The first column in our tables indicates the customer type. The second column gives the number of queries that correspond to the customer type. The last three columns show the statistics of γ_t.

Results. We observe from Tables 3, 4 and 5 that there is a significant number of instances where the value of customization γ_t is high. There are customer types for which the value of customization can reach 15% on average. Over all search queries in the dataset, on average, customization achieves an expected revenue increase of about 1%. We emphasize that even a few percentage point increase in the expected revenue can translate into a substantial increase in the expected profit of a firm, so even a few percentage points is significant in revenue management applications. The value of customization is higher for the customer types with small market size, indicating that there is significant value in customizing the offered assortment for these customer types rather than offering them the same assortment that is driven by the majority of the market. Note that γ_t tends to be large for the dataset where the fraction of the no-purchase queries is 30% as compared to the two other datasets. This observation is expected since there is no value of customization in the queries that ended up with a no-purchase. Overall, our findings suggest that there is significant value in identifying the type of customers and personalizing the assortment of hotels according to the customer type preferences.

8.2 Synthetic data

We test the value of customization by also using synthetically generated datasets, which allow us to vary the number of products and customer types in consideration.

Experimental setup. We generate our test problems as follows. There are n products and we vary n. We generate the revenue of each product independently from the exponential distribution with parameter 1. We consider m customer types with equal arrival probabilities, i.e., $\theta_j = 1/m$ for all $j \in \mathcal{M}$. We vary the number of customer types as well. We generate the preference weight for each

		γ_t (%)		
Cust. No	Queries	Mean	95%	Max
1	504	0.04	0.19	1.44
2	594	2.59	8.90	29.56
3	1611	0.43	1.84	9.97
4	1125	0.07	0.31	15.69
5	181	1.02	3.74	15.71
6	141	4.52	20.07	47.12
7	476	0.23	0.82	11.66
8	231	0.55	2.74	13.29
9	481	0.44	2.17	10.08
10	548	0.25	1.04	3.20
11	2448	0.16	0.85	3.34
12	1866	0.04	0.15	4.68
13	206	0.52	2.62	5.93
14	182	1.99	9.58	36.32
15	647	0.30	1.38	15.53
16	454	0.10	0.58	2.76

Table 4: Value of customization in the Expedia dataset with 50% fraction of no-purchase queries.

		γ_t (%)		
Cust. No	Queries	Mean	95%	Max
1	792	0.01	0.05	0.50
2	928	0.45	2.19	8.93
3	2775	0.07	0.45	3.11
4	1807	0.01	0.04	1.65
5	276	0.44	2.41	12.91
6	209	1.09	6.41	22.33
7	774	0.04	0.18	2.00
8	358	0.12	0.60	7.58
9	829	0.07	0.45	3.06
10	849	0.07	0.41	2.32
11	4363	0.05	0.27	4.80
12	3169	0.01	0.01	3.41
13	329	0.10	0.63	2.13
14	260	0.59	3.67	12.71
15	1088	0.05	0.24	4.60
16	687	0.03	0.15	1.72

Table 5: Value of customization in the Expedia dataset with 70% fraction of no-purchase queries.

product and customer type independently. The preference weight that a customer of type j attaches to product i is of the form $v_{ij} = B_{ij}\, X_{ij}$, where B_{ij} is sampled from the Bernoulli distribution with parameter 1/2 and X_{ij} is sampled from the folded standard normal distribution. Recall that folded standard normal random variable is the absolute value of a standard normal random variable. In this way, B_{ij} captures whether customers of type j are interested in product i. Conditional on customers of type j being interested in product i, X_{ij} captures the attractiveness that customers of this type attach to product i. Our goal is to generate problem instances where customers of different types have strong preferences between the different products. The samples $\{B_{ij} : i \in \mathcal{N}\}$ indicate whether customers of type j would even be interested in purchasing each product i. If $B_{ij} = 0$, then product i is not even considered for purchase by customer type j. We also tried exponential and uniform distributions, rather than the folded standard normal, for the samples $\{X_{ij} : i \in \mathcal{N}\}$ and our results were qualitatively the same. Our focus in this section is to characterize the value of customization, rather than finding near-optimal solutions for CAP. Thus, we focus on the case where the limit on the number of products in the first stage of CAP is $K = n$, so the optimal solution of CAP is given by $S^* = \mathcal{N}$. Once we generate a problem instance by using the approach described in this paragraph, for this problem instance, we solve CAP and MMNL as formulated in Section 3. Letting z_{CAP} and z_{MMNL}, respectively, be the optimal objective values of CAP and MMNL, for each problem instance t, we define the ratio

$$v_t = 100 \cdot \frac{z_{\mathsf{CAP}} - z_{\mathsf{MMNL}}}{z_{\mathsf{MMNL}}}.$$

Here, v_t can be interpreted as the percentage gain in the optimal expected revenue for problem instance t due to customization.

We vary the number of products over $n \in \{5, 10, 15, 20\}$ and the number of customer types over

		v_t (%)		
n	m	Mean	95%	Max
5	5	2.86	10.17	23.42
	10	4.65	12.22	18.09
	50	5.06	12.36	18.78
	100	5.64	11.38	20.01
	500	5.51	13.46	16.94
10	5	2.76	7.74	23.45
	10	4.72	10.89	20.32
	50	5.99	14.65	17.98
	100	5.76	11.79	15.77
	500	5.95	10.61	19.10
15	5	2.81	8.97	15.28
	10	3.94	11.04	17.74
	50	5.14	9.95	18.98
	100	5.99	11.73	18.02
	500	6.61	12.73	26.54
20	5	2.79	8.83	14.31
	10	4.00	8.98	20.94
	50	5.09	10.54	16.33
	100	6.06	14.24	21.59
	500	5.28	10.74	16.10

Table 6: Value of customization on synthetic datasets.

$m \in \{5, 10, 50, 100, 500\}$. For each value of n and m, we generate 100 problem instances by using the approach in the previous paragraph and report the mean, 95-th percentile and maximum value of v_t over all 100 instances. We restrict the number of products to 20, because for large problem instances, it is computationally challenging to solve MMNL by using the integer programming formulation with big-M constraints in Méndez-Díaz et al. (2014), especially considering that we solve 100 problem instance for each m and n combination. Thus, the computational bottleneck here is MMNL, rather than CAP. Our results are presented in Table 6.

Results. We observe from Table 6 that the optimal expected revenue of CAP can be significantly larger than the optimal expected revenue of MMNL. The average value of customization v_t is around 5% over all our test instances and the maximum value of v_t is above 20%. Thus, there is significant value in customizing the assortments offered to each customer type, when compared to offering the same assortment to all the customer types. Our numerical results indicate that the value of v_t tends to increase with the number of customer types m, so customization becomes more beneficial as the number of customer types gets larger.

9 Computational study: Solution quality

We present computational experiments on synthetically generated problem instances to numerically evaluate the performance of Augmented Greedy and our integer programming formulation.

Experimental setup. We generate random instances of CAP in the same way we did in Section 8.2. In particular, we sample the revenue of each product independently from the exponential distribution with parameter 1. The preference weight that a customer of type j associates with product i has the form $v_{ij} = B_{ij}\, X_{ij}$, where we sample B_{ij} from the Bernoulli distribution with parameter 1/2 and we sample X_{ij} from the folded standard normal distribution. The customer types have equal arrival probabilities. We vary the number of customer types over $m \in \{10, 50, 100\}$ and the number of products over $n \in \{50, 100\}$. We vary the upper bound on the number of products that we can pick in the first stage over $K \in \{\frac{1}{10}\, n, \frac{1}{2}\, n\}$. Practical values of m, n and K are highly dependent on the application setting. For example, mid-size online grocers stock $K = 10$ to 50 varieties in product categories such as yoghurt, olive oil and ice cream, out of a total of $n = 50$ to 500 product varieties available to them. Going more towards the larger end of the spectrum, large outdoor equipment suppliers offer $K = 100$ to 200 varieties in product categories such as tents, sleeping bags and hiking boots, out of a total of $n = 250$ to 1000 varieties available to them. Going towards the smaller end of the spectrum, cellphone service providers offer $K = 10$ to 50 phone varieties to their customers out of $n = 50$ to 200 viable phone models in the market. The number of customer types depends on how the customers are segmented. Segmenting the customers by the boroughs or zip codes in an urban area, one ends up with 10 to 20 customer types. Considering additional demographic factors, such as age and gender, increases the number of customer types to 50 or 100. One can work with more customer types depending on the amount of available data, but as the number of customer segments increases, the amount of data to estimate the choice model governing the behavior of each customer segment shrinks.

For each combination of m, n and K, we generate 50 problem instances of CAP. For each problem instance, we use Augmented Greedy to obtain a solution. By Theorem 5.1, Augmented Greedy gives $\Omega(1/\log m)$-approximate solution to CAP when customer types have equal arrival probabilities. Also, for each problem instance, we use the integer programming formulation in (6.5) to obtain a solution. By the discussion in Appendix H, this integer programming formulation provides a $(1-\epsilon)$-approximate solution to CAP, where ϵ is the accuracy parameter of Grid. We use $\epsilon = 0.01$ in our numerical experiments. Furthermore, letting z_{IP} be the optimal objective value of the integer program in (6.5), in the same appendix, we show that $(1+\epsilon)\, z_{\mathsf{IP}}$ is an upper bound on the optimal objective value of CAP. For each problem instance t, let $\mathsf{Rev}_t^{\mathsf{AUG}}$ be the objective value of CAP evaluated at the solution provided by Augmented Greedy, $\mathsf{Rev}_t^{\mathsf{IP}}$ be the objective value of CAP evaluated at the solution provided by the integer programming formulation, and UB_t be the upper bound on the optimal objective value of CAP provided by the integer programming formulation. For each problem instance t that we generate, we compute the two ratios

$$\gamma_{1t} = 100 \cdot \frac{\mathsf{Rev}_t^{\mathsf{AUG}}}{\mathsf{UB}_t} \qquad \text{and} \qquad \gamma_{2t} = 100 \cdot \frac{\mathsf{Rev}_t^{\mathsf{IP}}}{\mathsf{UB}_t}.$$

Note that γ_{1t} and γ_{2t} are upper bounds on the optimality gaps of the solutions from Augmented Greedy and the integer program. These quantities are upper bounds on the optimality gaps because we compare the expected revenues of the solutions with an upper bound on the optimal objective value of CAP, rather than the optimal objective value itself.

We consider values of $n \in \{50, 100\}$, $m \in \{10, 50, 100\}$ and $K \in \{\frac{1}{10}n, \frac{1}{2}n\}$. For each value of n, m and K, we generate 50 problem instances and report the mean, 5% percentile and minimum of the ratios γ_{1t} and γ_{2t} in our numerical experiments. Our results are shown in Table 7.

Results. We observe from Table 7 that Augmented Greedy has strong empirical performance. In an overwhelming majority of instances, the expected revenues of the solutions obtained by Augmented Greedy are within 1% of the upper bound on the optimal expected revenue. Moreover, the numerical performance of Augmented Greedy gets better as m increases. This suggests that our worst-case theoretical bound $\Omega(1/\log m)$ is not tight and the empirical performance of Augmented Greedy is much better than its theoretical guarantees. By the discussion in Appendix H, we know that the expected revenue of the assortment provided by the integer programming formulation is at least $(1-\epsilon) = 99\%$ fraction of the upper bound on the optimal expected revenue. The performance of the integer program is noticeably better than this 99% benchmark. Over all of our instances, on average, the assortment from the integer programming formulation obtains 99.5% of the upper bound on the optimal expected revenue. The running time for Augmented greedy scales roughy linearly with the number of customer types m, quadratically with the number of products n, and linearly with

Augmented Greedy

			γ_{1t} (%)			Time (s)
m	n	K	Mean	5%	Min	
10	50	5	99.4	99.3	96.3	1.4
		25	98.7	97.0	94.0	6.0
	100	10	99.0	97.7	95.0	15.7
		50	98.9	96.9	95.5	77.9
50	50	5	99.5	99.4	99.2	5.7
		25	99.4	99.1	98.8	26.2
	100	10	99.5	99.4	99.2	85.8
		50	99.5	99.3	99.3	325.5
100	50	5	99.5	99.5	99.5	11.6
		25	99.5	99.3	99.1	52.1
	100	10	99.5	99.5	99.5	179.1
		50	99.4	99.3	99.3	646.4

Integer Program

			γ_{2t} (%)			Time (s)
m	n	K	Mean	5%	Min	
10	50	5	99.5	99.3	99.3	11.4
		25	99.5	99.4	99.3	0.3
	100	10	99.5	99.4	99.3	3.6
		50	99.5	99.4	99.2	2.9
50	50	5	99.5	99.5	99.4	921.4
		25	99.5	99.4	99.4	2.1
	100	10	99.5	99.4	99.4	1612.0
		50	99.5	99.4	99.4	19.4
100	50	5	99.5	99.5	99.5	18032
		25	99.5	99.4	99.4	5.1
	100	10	99.5	99.5	99.5	4296.0
		50	99.5	99.4	99.4	34

Table 7: Comparison of Augmented Greedy and IP

the limit on the number of offered products K. In our numerical experiments, the running time for Augmented Greedy ranges from few seconds for small size instances to few minutes for large size instances. On the other hand, the running time for solving the integer programming formulation is somewhat unpredictable. For some instances, solving the integer program can be faster than running Augmented Greedy and takes few seconds. For other instances, solving the integer program can be time consuming and could take minutes or even hours to get a $(1-\epsilon)$-approximate solution. For instance, for our largest test problems with $m = 100$ and $n = 100$, the solution times for the integer programming formulation can range from 30 seconds to 4200 seconds. Furthermore, an important advantage of Augmented Greedy is that we can effortlessly parallelize this algorithm, because each iteration in the *for* loop of the algorithm is independent of the other iterations. Indeed, our implementation of Augmented Greedy is on an 8-core computer using Python 3.8.1 as the development environment and Gurobi 9.1.1 as the linear programming solver. We distributed the iterations in the *for* loop to the eight cores. Naturally, having access to more cores would yield improvements in the running time of Augmented Greedy roughly linear in the number of cores.

We also tested Augmented Greedy to solve CAP with heterogeneous arrival probabilities. In our theoretical results, the algorithm that gives $\Omega(1/\log m)$-approximation for this case uses Augmented Greedy recursively as a subroutine and finally solves a dynamic program. Here, we directly apply Augmented Greedy to test its empirical performance for this case of heterogeneous arrival probabilities, although we do not have theoretical guarantees under heterogeneous arrival probabilities. We randomly sample the arrival probabilities from a uniform distribution $[0, 1]$ and normalize them such that their sum is equal to 1. We present our results in Table 8 in Appendix I. Augmented Greedy continues to provide near-optimal assortments for an overwhelming majority of instances and the same conclusions we have in this section for the case of equal probabilities carry out to the case of heterogeneous probabilities.

10 Conclusions

We considered a joint assortment optimization and customization problem under a mixture of MNL models, where the firm commits to an assortment in the first stage, but can customize this assortment based on the type of the customer arriving in the second stage. This problem models the situation faced by many online retailers that can customize the assortment displayed to a customer. Despite its ubiquitous nature, however, to our knowledge, our joint assortment optimization and customization problem setup was unexplored until our work and it opens a number of research directions. An interesting research direction is to study CAP under general type of constraints that can be expressed as matroid or totally unimodular constraints. Our development exploited the properties of the revenue-maximizing assortment under the MNL model that we discussed in Appendix E. Thus, extensions to more general constraints appear to require different lines of attack. Another research direction is to study joint assortment optimization and customization problems under other choice models. For example, our efforts to extend our results to the case where each customer type follows a nested logit model, which is closely related to the MNL model, were not successful and it also appears that we need different lines of attack.

References

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

A. Aouad, J. Feldman, D. Segev, and D. J. Zhang. Click-based MNL: Algorithmic frameworks for modeling click data in assortment optimization. Technical report, Washington University, St. Louis, MO, 2019.

A. Aouad, A. N. Elmachtoub, K. J. Ferreira, and R. McNellis. Market segmentation trees. Technical report, Columbia University, New York, NY, 2020.

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

G. Berbeglia, A. Flores, and G. Gallego. Refined assortment optimization. Technical report, University of Melbourne, Melbourne, Australia, 2021.

J. J. M. Bront, I. Méndez-Díaz, and G. Vulcano. A column generation algorithm for choice-based network revenue management. *Operations research*, 57(3):769–784, 2009.

A. Désir, V. Goyal, and J. Zhang. Near-optimal algorithms for capacity constrained assortment optimization. *Available at SSRN 2543309*, 2014.

U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

J. Feldman and H. Topaloglu. Bounding optimal expected revenues for assortment optimization under mixtures of multinomial logits. *Production and Operations Management*, 24(10):1598–1620, 2015.

J. Feldman and H. Topaloglu. Technical note: Capacitated assortment optimization under the multinomial logit model with nested consideration sets. *Operations Research*, 66(2):380–391, 2018.

G. Gallego, G. Iyengar, R. Phillips, and A. Dubey. Managing flexible products on a network. Computational Optimization Research Center Technical Report TR-2004-01, Columbia University, 2004.

P. Gao, Y. Ma, N. Chen, G. Gallego, A. Li, P. Rusmevichientong, and H. Topaloglu. Assortment optimization and pricing under the multinomial logit model with impatient customers: Sequential recommendation and selection. *Operations Research*, (to appear), 2020.

S. Jagabathula. Assortment optimization under general choice. Technical report, NYU, New York, NY, 2016.

S. Jagabathula and G. Vulcano. A partial-order-based model to estimate individual preferences using panel data. *Management Science*, 64(4):1609–1628, 2018.

S. Jagabathula, D. Mitrofanov, and G. Vulcano. Personalized retail promotions through a DAG-based representation of customer preferences. *Operations Research*, (to appear), 2020.

Kaggle. Personalize expedia hotel searches. last checked: August 5, 2019., 2013. `https://www.kaggle.com/c/expedia-personalized-sort`.

R. M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

W. Ma. When is assortment optimization optimal? *Management Science*, (to appear), 2022.

I. Méndez-Díaz, J. J. M. Bront, G. Vulcano, and P. Zabala. A branch-and-cut algorithm for the latent-class logit assortment problem. *Discrete Applied Mathematics*, 164:246–263, 2014.

G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions-i. *Mathematical programming*, 14(1):265–294, 1978.

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

P. Rusmevichientong, Z.-J. M. Shen, and D. B. Shmoys. Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Operations Research*, 58(6):1666–1680, 2010.

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

A. Sen, A. Atamturk, and P. Kaminsky. Technical note – A conic integer optimization approach to the constrained assortment problem under the mixed multinomial logit model. *Operations Research*, 66(4):994–1003, 2018.

M. Sumida, G. Gallego, P. Rusmevichientong, H. Topaloglu, and J. M. Davis. Revenue-utility tradeoff in assortment optimization under the multinomial logit model with totally unimodular constraints. *Management Science*, (to appear), 2020.

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

R. Udwani. Submodular order functions and assortment optimization. Technical report, University of California, Berkeley, CA, 2021.

R. Wang. Capacitated assortment and price optimization under the multinomial logit model. *Operations Research Letters*, 40(6):492–497, 2012.

R. Wang and O. Sahin. The impact of consumer search cost on assortment planning and pricing. *Management Science*, 64(8):3649–3666, 2018.

A Appendix: Hardness results

We start by presenting the maximum coverage problem that we use in the proof of Theorem 2.1.

Maximum coverage problem. Given elements $\{1, 2, \ldots, m\}$ and sets $\{S_1, S_2, \ldots, S_n\}$ with $S_i \subseteq \{1, 2, \ldots, m\}$ for each $i = 1, \ldots, n$, we say that set S_i covers element j if $j \in S_i$. For a given K, the goal of the maximum coverage problem is to find at most K sets such that the total number of covered elements is maximized. This problem is NP-hard to approximate within a factor better than $(1 - \frac{1}{e})$ unless P = NP (Feige, 1998).

Proof of Theorem 2.1. Consider an instance of the maximum coverage problem. We construct an instance of CAP as follows. The products $\mathcal{N}$ correspond to the sets $\{S_1, S_2, \ldots, S_n\}$ and the customer types $\mathcal{M}$ correspond to the elements $\{1, 2, \ldots, m\}$. Fix $\epsilon > 0$ and let $\epsilon' = (1 - 1/e)\epsilon$. Let $\Gamma = 1/\epsilon' - 1$. In CAP, for all $i \in \mathcal{N}$ and $j \in \mathcal{M}$, the preference weights are given by

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

The product revenues are given by $r_i = 1$ for all $i \in \mathcal{N}$ and the arrival probabilities of the customer types are given by $\theta_j = 1/m$ for all $j \in \mathcal{M}$.

Suppose there exists a maximum coverage solution with an objective value z, i.e., there are z covered elements. We construct a solution for CAP using exactly the K products corresponding to the K sets in this maximum coverage solution. There are z customer types that are covered by these sets. Hence, for each customer type j among these z customer types, there exists a product i such that $v_{ij} = \Gamma$. Therefore, we get at least an expected revenue of $\frac{\Gamma}{1+\Gamma}$ from each of these customer types. Let R be the objective value of our solution for CAP. We have

$$R \geq \frac{1}{m} \cdot \frac{\Gamma}{1 + \Gamma} \cdot z = \frac{1}{m} \cdot (1 - \epsilon')z.$$

Now, let us consider a solution of CAP. Without loss of generality, the solution has K products and let R be its objective value. We construct a feasible solution for the maximum coverage problem by choosing exactly the sets corresponding to the K products in the solution of CAP. Let z be the resulting total number of covered elements, and for each $j \in \mathcal{M}$, let q_j be the number of sets that cover element j. If an element j is not covered, then $q_j = 0$ and the expected revenue of the customer type j is 0. Otherwise, if $q_j \geq 1$, then the expected revenue from customer type j is given by $\frac{q_j \Gamma}{1 + q_j \Gamma} \leq 1$. Therefore, the objective value of our solution for CAP is given by

$$R = \frac{1}{m} \sum_{j=1}^{m} \frac{q_j \Gamma}{1 + q_j \Gamma} \leq \frac{1}{m} \cdot z.$$

We know that unless P = NP, it is NP-hard to approximate the maximum coverage problem with a factor better than $(1 - \frac{1}{e})$ (Feige, 1998). Thus, it is NP-hard to approximate CAP within a factor better than $(1 - \frac{1}{e})(1 - \epsilon') = 1 - \frac{1}{e} - \epsilon$ for any $\epsilon > 0$. □

In the next theorem, we show that CAP is NP-hard even with two customer types. This

NP-hardness result holds even when all products have the same revenue. We use a reduction from the subset sum problem, which is defined as follows.

Subset sum problem. Given weights $w_1, w_2, \ldots, w_n$ and a cardinality K, the goal of the subset sum problem is to find a subset $S \subseteq \{1, \ldots, n\}$ of size K such that $\sum_{i \in S} w_i = 0$. This problem is NP-complete. (Karp, 1972).

Theorem A.1. CAP *is NP-hard even with two customer types and $r_i = 1$ for all $i \in \mathcal{N}$.*

Proof. Consider an instance $\{w_1, w_2, \ldots, w_n\}$ of the subset sum problem and assume without loss of generality that $w_1 = \max_{i=1,\ldots,n} w_i$. We define an instance of CAP with two customer types such that $\theta_1 = \theta_2 = 1/2$, and n products such that, for all $i \in \mathcal{N}$, we have

$$r_i = 1, \qquad v_{i1} = w_1 + w_i, \qquad v_{i2} = w_1 - w_i.$$

For $i \in \mathcal{N}$, let y_i be a binary variable that captures whether product i is in the optimal assortment of CAP. Let $S = \{i \in \mathcal{N} : y_i = 1\}$. Since all the products have the same revenue, it is optimal to offer all the products in S to both customer types. Hence, CAP is equivalent to maximizing

$$\frac{1}{2} \cdot \frac{\sum_{i \in \mathcal{N}} v_{i1} y_i}{1 + \sum_{i \in \mathcal{N}} v_{i1} y_i} + \frac{1}{2} \cdot \frac{\sum_{i \in \mathcal{N}} v_{i2} y_i}{1 + \sum_{i \in \mathcal{N}} v_{i2} y_i},$$

subject to the constraint $\sum_{i \in \mathcal{N}} y_i \leq K$. Since $\frac{\sum_{i \in \mathcal{N}} v_{i1} y_i}{1+\sum_{i \in \mathcal{N}} v_{i1} y_i}$ and $\frac{\sum_{i \in \mathcal{N}} v_{i2} y_i}{1+\sum_{i \in \mathcal{N}} v_{i2} y_i}$ are both increasing in y_i for any $i \in \mathcal{N}$, we have $\sum_{i \in \mathcal{N}} y_i = K$ in an optimal solution to CAP. Let

$$X = \sum_{i \in \mathcal{N}} v_{i1}\, y_i \qquad \text{and} \qquad \alpha = 2 w_1 K.$$

After simple algebraic manipulations, the objective function of CAP becomes

$$\frac{1}{2} \cdot \frac{X}{1+X} + \frac{1}{2} \cdot \frac{\alpha - X}{1 + \alpha - X} = 1 - \frac{1 + \alpha/2}{1 + \alpha + X(\alpha - X)}.$$

Maximizing the above objective function is equivalent to maximizing $X(\alpha - X)$. In this case, since $(X - \frac{\alpha}{2})^2 \geq 0$, we get

$$X(\alpha - X) \leq \frac{\alpha^2}{4},$$

with the inequality above holding as equality if and only if $X = \alpha/2$, which is equivalent to $\sum_{i \in \mathcal{N}} v_{i1} y_i = w_1 K$, i.e., $\sum_{i \in \mathcal{N}} w_i y_i = 0$. Therefore, solving CAP implies finding whether there exists a subset S of size K such that $\sum_{i \in S} w_i = 0$. Since the subset sum problem is NP-hard, it follows that CAP is NP-hard. $\square$

B Appendix: Counterexample to submodularity of the expected revenue

We give a counterexample to demonstrate that f is not submodular. Consider an instance of CAP with three products $n = 3$, along with revenues $r_1 = 3, r_2 = 2, r_3 = 1$, and one customer type $m = 1$. Dropping the index for the single customer type, the preference weights of the customer type are given by $v_1 = 1, v_2 = 1, v_3 = 100$. Let us denote the expected revenue function as

$R(S) = \frac{\sum_{i\in S} r_i v_i}{1+\sum_{i\in S} v_i}$, in which case, $f(S) = \max_{Q\subseteq S} R(Q)$. Consider the following sets $S = \{3\}$ and $T = \{2, 3\}$. We obtain the expected revenues

$$\begin{aligned}
&f(S) = R(\{3\}) = \frac{100}{101}\\
&f(S\cup\{1\}) = \max_{Q\subseteq S\cup\{1\}} R(Q) = \frac{3}{2}\\
&f(T) = \max_{Q\subseteq T} R(Q) = 1\\
&f(T\cup\{1\}) = \max_{Q\subseteq T\cup\{1\}} R(Q) = \frac{5}{3},
\end{aligned}$$

which yields $f(T\cup\{1\}) - f(T) = \frac{2}{3} > \frac{1}{2} + \frac{1}{101} = f(S\cup\{1\}) - f(S)$. Thus, f is not submodular.

C Appendix: Properties of the expected revenue

In this section, we show that the function f_j is monotone and subadditive for each $j \in \mathcal{M}$. Because f is obtained by a positive linear combination of these functions, monotonicity and subadditivity properties transfer to f as well.

Monotonicity. For any $j \in \mathcal{M}$, f_j is increasing, i.e., for any $A \subseteq B \subseteq \mathcal{N}$, we have $f_j(A) \le f_j(B)$. In particular, for $A \subseteq B \subseteq \mathcal{N}$, we have

$$f_j(A) = \max_{S\subseteq A} \frac{\sum_{i\in S} r_i v_{ij}}{1+\sum_{i\in S} v_{ij}} \le \max_{S\subseteq B} \frac{\sum_{i\in S} r_i v_{ij}}{1+\sum_{i\in S} v_{ij}} = f_j(B).$$

Therefore, $f = \sum_{j\in\mathcal{M}} \theta_j f_j$ is increasing as well.

Subadditivity. For any $j \in \mathcal{M}$, f_j is subadditive, i.e., $f_j(A\cup B) \le f_j(A)+f_j(B)$ for any $A, B \subseteq \mathcal{N}$. In fact, for $A, B \subseteq \mathcal{N}$, we have

$$f_j(A\cup B) = \max_{S\subseteq A\cup B} \frac{\sum_{i\in S} r_i v_{ij}}{1+\sum_{i\in S} v_{ij}} = \frac{\sum_{i\in \overline{S}} r_i v_{ij}}{1+\sum_{i\in \overline{S}} v_{ij}},$$

for some $\overline{S} \subseteq A \cup B$. Let us write $\overline{S}$ as $\overline{S} = S_1 \cup S_2$, where $S_1 \subseteq A$, $S_2 \subseteq B$ and $S_1 \cap S_2 = \varnothing$. In this case, we obtain

$$\begin{aligned}
f_j(A\cup B) &= \frac{\sum_{i\in S_1} r_i v_{ij} + \sum_{i\in S_2} r_i v_{ij}}{1+\sum_{i\in S_1} v_{ij} + \sum_{i\in S_2} v_{ij}} \le \frac{\sum_{i\in S_1} r_i v_{ij}}{1+\sum_{i\in S_1} v_{ij}} + \frac{\sum_{i\in S_2} r_i v_{ij}}{1+\sum_{i\in S_2} v_{ij}}\\
&\le \max_{S\subseteq A} \frac{\sum_{i\in S} r_i v_{ij}}{1+\sum_{i\in S} v_{ij}} + \max_{S\subseteq B} \frac{\sum_{i\in S} r_i v_{ij}}{1+\sum_{i\in S} v_{ij}} = f_j(A) + f_j(B).
\end{aligned}$$

Therefore, $f = \sum_{j\in\mathcal{M}} \theta_j f_j$ is subadditive as well.

D Appendix: Proof of Lemma 4.3

In this section, we give a proof for Lemma 4.3. Fix $j \in \mathcal{C}$ and $i \in \mathcal{N}$. Let $g(S) = \min(f_j(S), r_i)$. Note that g is monotone increasing because f_j is monotone increasing by the discussion in Appendix C.

Let $S \subseteq T \subseteq \mathcal{V}_i$ and $p \in \mathcal{V}_i \setminus T$. The goal is to show that $g(S \cup \{p\}) - g(S) \geq g(T \cup \{p\}) - g(T)$. We are going to distinguish three cases in our proof.

Case 1. Suppose that $f_j(T) \geq r_i$. In this case, by monotonicity of f_j, we get $f_j(T \cup \{p\}) \geq f_j(T) \geq r_i$. Therefore, we have $g(T \cup \{p\}) - g(T) = r_i - r_i = 0 \leq g(S \cup \{p\}) - g(S)$, where the last inequality holds by monotonicity of g.

Case 2. Suppose that $f_j(T \cup \{p\}) \geq r_i \geq f_j(T)$. Hence, we have $g(T \cup \{p\}) - g(T) = r_i - f_j(T)$. By monotonicity of f_j, we get $f_j(S) \leq f_j(T) \leq r_i$, so $g(S) = f_j(S)$. If we have $g(S \cup \{p\}) = r_i$, then having $g(S \cup \{p\}) - g(S) \geq g(T \cup \{p\}) - g(T)$ is equivalent to having $f_j(S) \leq f_j(T)$, which holds by the monotonicity of f_j and the desired result follows. Therefore, throughout the rest of this case, we assume that $g(S \cup \{p\}) = f_j(S \cup \{p\})$.

Having $g(S \cup \{p\}) = f_j(S \cup \{p\})$ implies that $f_j(S \cup \{p\}) \leq r_i$. Since product i has the smallest revenue among the products in $\mathcal{V}_i$, we get $f_j(S \cup \{p\}) \leq r_s$ for all $s \in \mathcal{V}_i$. A standard property of the assortment optimization problem under the MNL model is that a product is in the revenue-maximizing assortment if and only if the revenue of the product is greater than or equal to the optimal expected revenue of the assortment optimization problem. For completeness, we show this property in Appendix E.

By the property in the previous paragraph, having $r_s \geq f_j(S \cup \{p\})$ for all $s \in \mathcal{V}_i \supseteq S \cup \{p\}$ implies that if the assortment in the first stage of CAP is $S \cup \{p\}$, then it is optimal to offer all products in $S \cup \{p\}$ to customer type j, i.e.,

$$f_j(S \cup \{p\}) = \frac{\sum_{\ell \in S \cup \{p\}} r_\ell \, v_{\ell j}}{1 + \sum_{\ell \in S \cup \{p\}} v_{\ell j}}.$$

Similarly, because we have $f_j(S) \leq f_j(T) \leq r_i$, by the same property of the assortment optimization problem under the MNL model, we get

$$f_j(S) = \frac{\sum_{\ell \in S} r_\ell \, v_{\ell j}}{1 + \sum_{\ell \in S} v_{\ell j}}, \qquad f_j(T) = \frac{\sum_{\ell \in T} r_\ell \, v_{\ell j}}{1 + \sum_{\ell \in T} v_{\ell j}}.$$

Therefore, we obtain

$$\begin{aligned} g(S \cup \{p\}) - g(S) \;&=\; f_j(S \cup \{p\}) - f_j(S) \;=\; \frac{v_{pj}}{1 + v_{pj} + \sum_{\ell \in S} v_{\ell j}} \cdot (r_p - f_j(S)) \\ &\geq\; \frac{v_{pj}}{1 + v_{pj} + \sum_{\ell \in T} v_{\ell j}} \cdot (r_p - f_j(T)) \;=\; \frac{\sum_{\ell \in T \cup \{p\}} r_\ell \, v_{\ell j}}{1 + \sum_{\ell \in T \cup \{p\}} v_{\ell j}} - \frac{\sum_{\ell \in T} r_\ell \, v_{\ell j}}{1 + \sum_{\ell \in T} v_{\ell j}} \\ &=\; \frac{\sum_{\ell \in T \cup \{p\}} r_\ell \, v_{\ell j}}{1 + \sum_{\ell \in T \cup \{p\}} v_{\ell j}} - g(T), \end{aligned}$$

where the first two equalities use algebraic manipulations and the inequality holds because we have $f_j(S) \leq f_j(T)$, $\sum_{\ell \in S} v_{\ell j} \leq \sum_{\ell \in T} v_{\ell j}$ and $r_p - f_j(T) \geq 0$. Hence, it is sufficient to show that

$$\frac{\sum_{\ell \in T \cup \{p\}} r_\ell \, v_{\ell j}}{1 + \sum_{\ell \in T \cup \{p\}} v_{\ell j}} \geq r_i, \tag{D.1}$$

in which case, noting that we have $g(T \cup \{p\}) = r_i$, the previous chain of inequalities would yield

$g(S \cup \{p\}) - g(S) \geq g(T \cup \{p\}) - g(T)$, as desired. Thus, we finish this case by establishing that (D.1) holds. Let $f_j(T \cup \{p\}) = \frac{\sum_{\ell \in A} r_\ell \, v_{\ell j}}{1 + \sum_{\ell \in A} v_{\ell j}}$ for some $A \subseteq T \cup \{p\}$ and let $B = T \cup \{p\} \setminus A$. Note that $r_i = \min\{r_\ell : \ell \in \mathcal{V}_i\}$ by the definition of $\mathcal{V}_i$, but since $T \cup \{p\} \subseteq \mathcal{V}_i$, we have $r_i \leq r_\ell$ for all $\ell \in B$. In this case, we get

$$\begin{aligned} &\frac{\sum_{\ell \in T \cup \{p\}} r_\ell \, v_{\ell j}}{1 + \sum_{\ell \in T \cup \{p\}} v_{\ell j}} \\ &\qquad = \frac{1 + \sum_{\ell \in A} v_{\ell j}}{1 + \sum_{\ell \in A} v_{\ell j} + \sum_{\ell \in B} v_{\ell j}} \cdot \frac{\sum_{\ell \in A} r_\ell \, v_{\ell j}}{1 + \sum_{\ell \in A} v_{\ell j}} + \frac{\sum_{\ell \in B} v_{\ell j}}{1 + \sum_{\ell \in A} v_{\ell j} + \sum_{\ell \in B} v_{\ell j}} \cdot \frac{\sum_{\ell \in B} r_\ell \, v_{\ell j}}{\sum_{\ell \in B} v_{\ell j}} \\ &\qquad \geq \frac{1 + \sum_{\ell \in A} v_{\ell j}}{1 + \sum_{\ell \in A} v_{\ell j} + \sum_{\ell \in B} v_{\ell j}} \cdot r_i + \frac{\sum_{\ell \in B} v_{\ell j}}{1 + \sum_{\ell \in A} v_{\ell j} + \sum_{\ell \in B} v_{\ell j}} \cdot r_i \;=\; r_i, \end{aligned}$$

where the inequality holds because $f_j(T \cup \{p\}) \geq r_i$ in the current case we consider and $r_i \leq r_\ell$ for all $\ell \in B$ so that $r_i \leq \frac{\sum_{\ell \in B} r_\ell \, v_{\ell j}}{\sum_{\ell \in B} v_{\ell j}}$.

Case 3. Suppose that $r_i \geq f_j(T \cup \{p\})$. By the monotonicity of f_j we get $r_i \geq f_j(T \cup \{p\}) \geq f_j(T)$ and $r_i \geq f_j(T \cup \{p\}) \geq f_j(S \cup \{p\}) \geq f_j(S)$. By using the standard property of the assortment optimization problem under the MNL model, these two chains of inequalities implies that if we pick the assortment $T \cup \{p\}$, $S \cup \{p\}$, S or T in the first stage of CAP, then it would be optimal to offer the full available assortment $T \cup \{p\}$, $S \cup \{p\}$, S or T to customer type j, i.e.,

$$f_j(S \cup \{p\}) = \frac{\sum_{\ell \in S \cup \{p\}} r_\ell \, v_{\ell j}}{1 + \sum_{\ell \in S \cup \{p\}} v_{\ell j}}, \quad f_j(T \cup \{p\}) = \frac{\sum_{\ell \in T \cup \{p\}} r_\ell \, v_{\ell j}}{1 + \sum_{\ell \in T \cup \{p\}} v_{\ell j}}, \quad f_j(S) = \frac{\sum_{\ell \in S} r_\ell \, v_{\ell j}}{1 + \sum_{\ell \in S} v_{\ell j}},$$
$$f_j(T) = \frac{\sum_{\ell \in T} r_\ell \, v_{\ell j}}{1 + \sum_{\ell \in T} v_{\ell j}}. \tag{D.2}$$

Therefore, we get

$$\begin{aligned} g(S \cup \{p\}) - g(S) &= f_j(S \cup \{p\}) - f_j(S) = \frac{v_{pj}}{1 + v_{pj} + \sum_{\ell \in S} v_{\ell j}} \cdot (r_p - f_j(S)) \\ &\geq \frac{v_{pj}}{1 + v_{pj} + \sum_{\ell \in T} v_{\ell j}} \cdot (r_p - f_j(T)) = f_j(T \cup \{p\}) - f_j(T) = g(T \cup \{p\}) - g(T), \end{aligned}$$

where the second and third equalities follow by using (D.2) along with algebraic manipulations, whereas the inequality holds because $f_j(S) \leq f_j(T)$, $\sum_{\ell \in S} v_{\ell j} \leq \sum_{\ell \in T} v_{\ell j}$ and $r_p - f_j(T) \geq 0$.

E Appendix: Assortment optimization under the multinomial logit model

We consider the assortment optimization problem under the MNL model and give a characterization of the revenue-maximizing assortment. We use this characterization at several places throughout the paper. We use $\mathcal{N} = \{1, \ldots, n\}$ to denote the set of products. There is a single customer type. A customer associates the preference weight v_i with product i. The revenue of product i is r_i. Thus, if we offer the assortment $S \subseteq \mathcal{N}$, then we obtain an expected revenue of $\frac{\sum_{i \in \mathcal{N}} r_i \, v_i}{1 + \sum_{i \in \mathcal{N}} v_i}$. We want to solve the problem

$$\max_{S \subseteq \mathcal{N}} \frac{\sum_{i\in S} r_i\, v_i}{1 + \sum_{i \in S} v_i}. \tag{E.1}$$

In the next lemma, we give a characterization of an optimal solution to the problem above.

Lemma E.1. *Letting z^* be the optimal objective value of problem* (E.1)*, there exists an optimal solution S^* to this problem that satisfies*

$$S^* = \{i \in \mathcal{N} \;:\; r_i \geq z^*\}.$$

Proof. Let $R(S) = \frac{\sum_{i\in S} r_i\, v_i}{1+\sum_{i\in S} v_i}$ be the objective function of problem (E.1) and let $W(S) = \frac{\sum_{i\in S} r_i\, v_i}{\sum_{i\in S} v_i}$ for notational brevity. For any $S, A \subseteq \mathcal{N}$ with $S \cap A = \varnothing$, we have

$$\begin{aligned} R(S \cup A) &= \frac{\sum_{i\in S} r_i\, v_i + \sum_{i\in A} r_i\, v_i}{1 + \sum_{i\in S} v_i + \sum_{i\in A} v_i} \\ &= \frac{1 + \sum_{i\in S} v_i}{1 + \sum_{i\in S} v_i + \sum_{i\in A} v_i} \cdot \frac{\sum_{i\in S} r_i v_i}{1 + \sum_{i\in S} v_i} + \frac{\sum_{i\in A} v_i}{1 + \sum_{i\in S} v_i + \sum_{i\in A} v_i} \cdot \frac{\sum_{i\in A} r_i\, v_i}{\sum_{i\in A} v_i} \\ &= \frac{1 + \sum_{i\in S} v_i}{1 + \sum_{i\in S} v_i + \sum_{i\in A} v_i} \cdot R(S) + \frac{\sum_{i\in A} v_i}{1 + \sum_{i\in S} v_i + \sum_{i\in A} v_i} \cdot W(A), \end{aligned}$$

so $R(S \cup A)$ is a convex combination of $R(S)$ and $W(A)$ and the latter quantity can be viewed as a weighted average of the revenues of the products in A.

Let $\overline{S}$ be any optimal solution to problem (E.1) and $\hat{S} = \{i \in \mathcal{N} : r_i \geq z^*\}$. Assume that $\overline{S} \setminus \hat{S} \neq \varnothing$. Note that $r_i < z^*$ for each $i \in \overline{S} \setminus \hat{S}$, so we have $W(\overline{S} \setminus \hat{S}) < z^*$. By the equality above, $R(\overline{S})$ is a convex combination of $R(\overline{S} \cap \widehat{S})$ and $W(\overline{S} \setminus \hat{S})$, so since $R(\overline{S}) = z^* > W(\overline{S} \setminus \hat{S})$, it must be the case that $R(\overline{S} \cap \hat{S}) \geq R(\overline{S})$. Thus, $\overline{S} \cap \hat{S}$ is an optimal solution to problem (E.1) as well, so there exists an optimal solution to problem (E.1) that is a subset of $\hat{S} = \{i \in \mathcal{N} : r_i \geq z^*\}$.

Let $\tilde{S}$ be an optimal solution to problem (E.1) that is a subset of $\hat{S} = \{i \in \mathcal{N} : r_i \geq z^*\}$. Assume that $\hat{S} \setminus \tilde{S} \neq \varnothing$. Note that we have $r_i \geq z^*$ for each $i \in \hat{S} \setminus \tilde{S}$, so $W(\hat{S} \setminus \tilde{S}) \geq z^*$. Using the fact that $\tilde{S} \subseteq \hat{S}$, by the equality above, $R(\hat{S})$ is a convex combination of $R(\tilde{S})$ and $W(\hat{S} \setminus \tilde{S})$, so because $W(\hat{S} \setminus \tilde{S}) \geq z^* = R(\tilde{S})$, it must be the case that $R(\hat{S}) \geq R(\tilde{S})$, which implies that $\hat{S} = \{i \in \mathcal{N} : r_i \geq z^*\}$ is an optimal solution to problem (E.1), as desired. □

The above lemma is a standard result in assortment optimization under the MNL model and it is stated in various forms in Talluri and van Ryzin (2004), Gallego et al. (2004) and Rusmevichientong and Topaloglu (2012).

F Appendix: Proof of Theorem 5.2

In this section, we give a proof for Theorem 5.2. We will use two auxiliary lemmas in the proof. We start by stating and proving these two lemmas. Following the two lemmas, we conclude this section with a proof for Theorem 5.2.

F.1 Auxiliary lemmas

The first lemma shows the existence of a subset of customers $\mathcal{X}$ such that the optimal expected revenue from the customers in $\mathcal{X}$ is a constant fraction of the optimal expected revenue of CAP and the expected revenues from the customer types in $\mathcal{X}$ satisfy a certain monotonicity property based on where they lie on the leaf nodes of the tree.

Lemma F.1. *Let S^* be the optimal solution of CAP and z_{CAP} be its optimal objective value. There exists a subset of customer types $\mathcal{X} \subseteq \mathcal{M}$ that satisfies the following properties.*

(i) For all $j, q \in \mathcal{X}$, if $j \in G_i$ and $q \in G_{i'}$ with $i < i'$, then $f_j(S^*) \geq f_q(S^*)$.

(ii) $f^{\mathcal{X}}(S^*) \geq \frac{1}{4} \cdot z_{\mathsf{CAP}}$.

Proof. Recall that the set of all customer types is given by $\mathcal{M} = \left\{\bigcup_{i=1}^{L} \mathcal{M}(\ell_i)\right\}$. We partition the customer types $\mathcal{M}$ as

$$\mathcal{M}^{\mathsf{even}} = \left\{\bigcup_{i=1}^{L} \mathcal{M}(\ell_i) \;\middle|\; \ell_i \text{ is even}\right\} \qquad \text{and} \qquad \mathcal{M}^{\mathsf{odd}} = \left\{\bigcup_{i=1}^{L} \mathcal{M}(\ell_i) \;\middle|\; \ell_i \text{ is odd}\right\}.$$

Hence,

$$f^{\mathcal{M}_{\mathsf{even}}}(S^*) + f^{\mathcal{M}_{\mathsf{odd}}}(S^*) = f^{\mathcal{M}}(S^*) = z_{\mathsf{CAP}}.$$

One of the two terms on the left hand side should be at least $z_{\mathsf{CAP}}/2$. Let us assume without loss of generality that

$$f^{\mathcal{M}_{\mathsf{even}}}(S^*) \geq \frac{z_{\mathsf{CAP}}}{2}.$$

Let $\mathcal{I} = \{i = 1, \ldots, L \mid \ell_i \text{ is even}\}$ and recall that $G_i = \mathcal{M}(\ell_i)$. In particular,

$$\mathcal{M}^{\mathsf{even}} = \bigcup_{i \in \mathcal{I}} G_i.$$

Let $i^* \in \mathcal{I}$ be the largest index in $\mathcal{I}$. In other words, G_{i^*} is the group of customer types with the highest values of θ_j among the groups in $\mathcal{M}^{\mathsf{even}}$. We construct the subset of customer types $\mathcal{X}$ as follows. We first add all customer types in the group G_{i^*} to $\mathcal{X}$. Then, we go in descending order from $i^* - 1$ to 1, where at each $i \in \{i^* - 1, \ldots, 2, 1\}$, we first check if i belongs to $\mathcal{I}$. If we have $i \in \mathcal{I}$, then we consider the set of customer types

$$\mathcal{C} = \left\{j \in G_i : f_j(S^*) \geq \max_{q \in \mathcal{X}} \; f_q(S^*)\right\},$$

where the max is taken over the set $\mathcal{X}$ constructed up to the current iteration. We update $\mathcal{X}$ by adding the customer types in $\mathcal{C}$ and move on to the next group of customers G_{i-1}. If we have $i \notin \mathcal{I}$, then we move directly to the next group of customers G_{i-1}. By construction, when we are at a group G_i and $i \in \mathcal{I}$, we add a customer type $j \in G_i$ to $\mathcal{X}$ only if $f_j(S^*)$ is greater than $f_q(S^*)$ for all customer types q in the groups that we have already checked, i.e., the groups $G_{i'}$ with $i' > i$. Therefore, $\mathcal{X}$ satisfies the first property in the lemma.

In the remainder of the proof, we will establish that the subset of customers $\mathcal{X}$ satisfies the second property. Consider a customer type $j \in \mathcal{M}_{\mathsf{even}}$, so $j \in G_i$ for some $i \in \mathcal{I}$. If we have $j \notin \mathcal{X}$, then by construction, there exists a customer type $q \in \mathcal{X}$ such that $f_q(S^*) > f_j(S^*)$. Moreover, the customer type q was added to $\mathcal{X}$ before we arrive at iteration i. Hence, we have $q \in G_{i'}$ for some $i' \in \mathcal{I}$ such that $i' > i$. By our definition of the groups G_i, this observation implies that

$$\frac{1}{m^{\ell_{i'}+1}} \leq \theta_q < \frac{1}{m^{\ell_{i'}}} \qquad \text{and} \qquad \frac{1}{m^{\ell_i+1}} \leq \theta_j < \frac{1}{m^{\ell_i}},$$

where $\ell_i > \ell_{i'}$. However, since both ℓ_i and $\ell_{i'}$ are even, we get $\ell_i \geq \ell_{i'} + 2$. Therefore, we have

$$\theta_q \geq \frac{1}{m^{\ell_{i'}+1}} \geq \frac{1}{m^{\ell_i - 1}} \geq m \cdot \theta_j.$$

Since, $f_q(S^*) > f_j(S^*)$ and $q \in \mathcal{X}$, it follows that

$$\theta_j f_j(S^*) \leq \frac{1}{m} \cdot \theta_q f_q(S^*) \leq \frac{1}{m} \cdot f^{\mathcal{X}}(S^*).$$

Hence, by summing over all customer types in $\mathcal{M}_{\mathsf{even}} \setminus \mathcal{X}$ which, number at most m, we get

$$f^{\mathcal{M}_{\mathsf{even}} \setminus \mathcal{X}}(S^*) \leq f^{\mathcal{X}}(S^*).$$

Thus, we get

$$2 f^{\mathcal{X}}(S^*) \geq f^{\mathcal{X}}(S^*) + f^{\mathcal{M}_{\mathsf{even}} \setminus \mathcal{X}}(S^*) = f^{\mathcal{M}_{\mathsf{even}}}(S^*) \geq \frac{z_{\mathsf{CAP}}}{2}.$$

The same argument holds if we assume that $f^{\mathcal{M}_{\mathsf{odd}}}(S^*) \geq \frac{z_{\mathsf{CAP}}}{2}$. □

Before we present the second lemma, we introduce some additional notation. Let S^* be the optimal solution of CAP. For each $j \in \mathcal{M}$, let $S_j^* \subseteq S^*$ be the optimal subset to offer to customer type j in CAP. For a subset of customer types $\mathcal{C} \subseteq \mathcal{M}$ and a subset of products $A \subseteq \mathcal{N}$, we define the expected revenue function

$$h^{\mathcal{C}}(A) = \sum_{j \in \mathcal{C}} \theta_j \frac{\sum_{i \in A \cap S_j^*} r_i v_{ij}}{1 + \sum_{i \in A \cap S_j^*} v_{ij}}.$$

Note that $h^{\mathcal{C}}(A)$ corresponds to the total expected revenue from customer types $\mathcal{C}$ when we offer assortment $A \cap S_j^*$ to customer type $j \in \mathcal{C}$. Recall that

$$f^{\mathcal{C}}(A) = \sum_{j \in \mathcal{C}} \theta_j \cdot \max_{S_j \subseteq A} \frac{\sum_{i \in S_j} r_i v_{ij}}{1 + \sum_{i \in S_j} v_{ij}}.$$

Therefore, for any $A \subseteq \mathcal{N}$ and any $\mathcal{C} \subseteq \mathcal{M}$,

$$f^{\mathcal{C}}(A) \geq h^{\mathcal{C}}(A). \tag{F.1}$$

Our second lemma shows that for any subset of products A of size k and any subtree of customer types $T \in \mathcal{L}$, we either give a lower bound on the expected revenue R_{Tk} of the assortment provided by Augmented Greedy or show the existence of a partition of products A that verifies a certain inequality. In particular, we have the following lemma.

Lemma F.2. *Consider a pair (A, T) where $A \subseteq S^*$ is a subset of the optimal set of products to offer in* CAP *and $T \in \mathcal{L}$ is a subtree of customer types. Let $|A| = k$, T_1 be the left subtree of T and T_2 be the right subtree of T. Lastly, let $\mathcal{X}$ be the set of customer types defined in Lemma F.1. Then, at least one of the following two statements holds for the pair (A, T).*

(i) $R_{Tk} = \Omega(1/\log m) \cdot h^{\mathcal{X} \cap \mathcal{C}_T}(A)$.

(ii) There exists a partition of A, i.e., $A = A_1 \cup A_2$, $A_1 \cap A_2 = \varnothing$ such that,

$$h^{\mathcal{X} \cap \mathcal{C}_{T_1}}(A_1) + h^{\mathcal{X} \cap \mathcal{C}_{T_2}}(A_2) \geq (1 - \frac{1}{\log m}) \cdot h^{\mathcal{X} \cap \mathcal{C}_T}(A).$$

Proof. First, let us address the case where T is a leaf of $\mathcal{T}$. We will show that the first statement in the lemma is verified in this case. Since T is a leaf of $\mathcal{T}$, it contains a unique group of customer types. In particular, we have $\mathcal{C}_T = G_i$ for some $i \in \{1, \ldots, L\}$. By definition, each customer type j in the group G_i has its arrival probability θ_j between $1/m^{\ell_i}$ and $1/m^{\ell_i+1}$ for some $\ell_i \in \mathbb{Z}$. Moreover, the rounded values of the arrival probabilities are such that $\theta_j m^{\ell_i+1}$ is an integer between 1 and m. Therefore, a customer type $j \in \mathcal{C}_T$ is equivalent to $\theta_j m^{\ell_i+1}$ copies of customer types with the same preference weights $\{v_{\ell j} : \ell \in \mathcal{N}\}$ such that each one of them has arrival probability $1/m^{\ell_i+1}$. Since $1 \leq \theta_j m^{\ell_i+1} \leq m$, we can view $\mathcal{C}_T$ as a group of at most m^2 customer types, where each customer type has the same arrival probability. We know from Theorem 5.1 that Augmented Greedy gives $\Omega(1/\log m)$-approximation to CAP when the number of customer types is m and they all have the same arrival probability. Therefore, Augmented Greedy gives $\Omega(1/\log(m^2))$-approximation to CAP with the input customer types $\mathcal{C}_T$. Recall that $\Delta_{Tk} = \mathsf{AugGreedy}(\mathcal{C}_T, k)$ is the assortment returned by Augmented Greedy with the inputs $\mathcal{C}_T$ for the set of customer types and k for the cardinality constraint. Thus, for any $A \subseteq \mathcal{N}$ such that $|A| = k$, we have

$$f^{\mathcal{C}_T}(\Delta_{Tk}) = \Omega(1/\log(m^2)) \cdot f^{\mathcal{C}_T}(A) = \Omega(1/\log m) \cdot f^{\mathcal{C}_T}(A).$$

Moreover,

$$f^{\mathcal{C}_T}(A) \geq f^{\mathcal{X} \cap \mathcal{C}_T}(A) \geq h^{\mathcal{X} \cap \mathcal{C}_T}(A),$$

where the last inequality follows from (F.1). Therefore,

$$R_{Tk} = f^{\mathcal{C}_T}(\Delta_{Tk}) = \Omega(1/\log m) \cdot h^{\mathcal{X} \cap \mathcal{C}_T}(A).$$

Next, suppose that T is not a leaf. Let G_ℓ be the group among $\{G_1, \ldots, G_L\}$ that contains the customer type j that has the smallest arrival probability θ_j among those in $\mathcal{X} \cap \mathcal{C}_{T_2}$. Let

$$q = \arg\max\{f_j(S^*) \;:\; j \in \mathcal{X} \cap \mathcal{C}_{T_2} \cap G_\ell\}, \tag{F.2}$$

i.e., q is the customer type with the highest $f_j(S^*)$ value among all customer types in G_ℓ that belong to $\mathcal{X} \cap \mathcal{C}_{T_2}$. Let us show the following claim.

Claim. For any $j \in \mathcal{X} \cap \mathcal{C}_{T_1}$, we have $S^*_j \subseteq S^*_q$, whereas for any $j \in \mathcal{X} \cap \mathcal{C}_{T_2}$, we have $S^*_q \subseteq S^*_j$.

Let us start with the first inclusion. Consider $j \in \mathcal{X} \cap \mathcal{C}_{T_1}$, and let $\ell' \in \{1, \ldots, L\}$ be such that $j \in G_{\ell'}$. We know that $q \in \mathcal{X} \cap \mathcal{C}_{T_2} \cap G_\ell$. Since T_1 is on the left of T_2, we get that $\ell' < \ell$. Hence, from the first property of Lemma F.1, for all $j \in \mathcal{X} \cap \mathcal{C}_{T_1}$, we have

$$f_j(S^*) \geq f_q(S^*).$$

By Lemma E.1 in Appendix E, a product is in the optimal assortment that we offer to a customer type if and only if its revenue is greater than the optimal expected revenue from the customer type. Recall that $f_j(S^*)$ is the optimal expected revenue for customer type j with universe of products S^* and the optimal solution for this problem is S_j^*. Therefore, for any $j \in \mathcal{M}$, we have

$$S_j^* = \{i \in S^* \ : \ r_i \geq f_j(S^*)\}.$$

In particular, for any $j \in \mathcal{X} \cap \mathcal{C}_{T_1}$ and any $i \in S_j^*$, we get $r_i \geq f_j(S^*) \geq f_q(S^*)$. Therefore, $i \in S_q^*$, which implies that $S_j^* \subseteq S_q^*$.

Now, let us show the second inclusion in the claim. Let $j \in \mathcal{X} \cap \mathcal{C}_{T_2}$. If we have $j \in G_\ell$, then $j \in \mathcal{X} \cap \mathcal{C}_{T_2} \cap G_\ell$, and by definition (F.2) of customer type q, we have $f_q(S^*) \geq f_j(S^*)$. On the other hand, consider the case $j \notin G_\ell$. By its definition, G_ℓ contains the customer type with the smallest arrival probability among customer types $\mathcal{X} \cap \mathcal{C}_{T_2}$. In this case, we get $j \in G_{\ell'}$ with $\ell' > \ell$. Hence, from the first property in Lemma F.1, we get that $f_q(S^*) \geq f_j(S^*)$. Therefore, for all $j \in \mathcal{X} \cap \mathcal{C}_{T_2}$, we have $f_q(S^*) \geq f_j(S^*)$. Similar to the proof of the first part of the claim, by Lemma E.1, we get $S_q^* \subseteq S_j^*$ for any $j \in \mathcal{X} \cap \mathcal{C}_{T_2}$ which concludes the proof of the claim.

We partition A as

$$A_1 = A \cap S_q^* \qquad \text{and} \qquad A_2 = A \setminus A_1.$$

For $j \in \mathcal{X} \cap \mathcal{C}_{T_1}$, from the above claim, we have $S_q^* \cap S_j^* = S_j^*$. Hence, we get $A_1 \cap S_j^* = A \cap S_q^* \cap S_j^* = A \cap S_j^*$. Therefore, it follows that

$$h^{\mathcal{X} \cap \mathcal{C}_{T_1}}(A) = \sum_{j \in \mathcal{X} \cap \mathcal{C}_{T_1}} \theta_j \frac{\sum_{i \in A \cap S_j^*} r_i \, v_{ij}}{1 + \sum_{i \in A \cap S_j^*} v_{ij}} = \sum_{j \in \mathcal{X} \cap \mathcal{C}_{T_1}} \theta_j \frac{\sum_{i \in A_1 \cap S_j^*} r_i \, v_{ij}}{1 + \sum_{i \in A_1 \cap S_j^*} v_{ij}} = h^{\mathcal{X} \cap \mathcal{C}_{T_1}}(A_1).$$

We have $\mathcal{C}_T = \mathcal{C}_{T_1} \cup \mathcal{C}_{T_2}$ with $\mathcal{C}_{T_1} \cap \mathcal{C}_{T_2} = \varnothing$, thus

$$h^{\mathcal{X} \cap \mathcal{C}_T}(A) = h^{\mathcal{X} \cap \mathcal{C}_{T_1}}(A) + h^{\mathcal{X} \cap \mathcal{C}_{T_2}}(A).$$

On the other hand, by subadditivity of the function $h^{\mathcal{C}}$, we get

$$h^{\mathcal{X} \cap \mathcal{C}_{T_2}}(A_1) + h^{\mathcal{X} \cap \mathcal{C}_{T_2}}(A_2) \geq h^{\mathcal{X} \cap \mathcal{C}_{T_2}}(A).$$

Therefore, putting all together,

$$h^{\mathcal{X} \cap \mathcal{C}_{T_1}}(A_1) + h^{\mathcal{X} \cap \mathcal{C}_{T_2}}(A_1) + h^{\mathcal{X} \cap \mathcal{C}_{T_2}}(A_2) \geq h^{\mathcal{X} \cap \mathcal{C}_T}(A). \tag{F.3}$$

For $j \in \mathcal{X} \cap \mathcal{C}_{T_2}$, from the above claim, we have $S_q^* \cap S_j^* = S_q^*$, so $A_1 \cap S_j^* = A \cap S_q^* \cap S_j^* = A \cap S_q^* = A_1 = A_1 \cap S^*$ where the last equality holds simply because $A_1 \subseteq S^*$. Therefore, the customer types in $\mathcal{X} \cap \mathcal{C}_{T_2}$ are complete with respect to A_1. Moreover, $|A_1| \leq |A| = k$. Hence, by Theorem 4.2, it follows that

$$R_{Tk} = f^{\mathcal{C}_T}(\Delta_{Tk}) \geq \left(1 - \frac{1}{e}\right) \cdot f^{\mathcal{X} \cap \mathcal{C}_{T_2}}(A_1) \geq \left(1 - \frac{1}{e}\right) \cdot h^{\mathcal{X} \cap \mathcal{C}_{T_2}}(A_1),$$

where the last inequality uses (F.1). Lastly, if $h^{\mathcal{X} \cap \mathcal{C}_{T_2}}(A_1) \geq \frac{1}{\log m} \cdot h^{\mathcal{X} \cap \mathcal{C}_T}(A)$, then the inequality yields $R_{Tk} = \Omega(1/\log m) \cdot h^{\mathcal{X} \cap \mathcal{C}_T}(A)$, which establishes that the first property in the lemma is verified. Otherwise, if $h^{\mathcal{X} \cap \mathcal{C}_{T_2}}(A_1) < \frac{1}{\log m} \cdot h^{\mathcal{X} \cap \mathcal{C}_T}(A)$, then (F.3) implies that

$$h^{\mathcal{X} \cap \mathcal{C}_{T_1}}(A_1) + h^{\mathcal{X} \cap \mathcal{C}_{T_2}}(A_2) \geq \left(1 - \frac{1}{\log m}\right) \cdot h^{\mathcal{X} \cap \mathcal{C}_T}(A),$$

which establishes that the second property in the lemma is verified. □

F.2 Proof of the main result

We use the two auxiliary lemmas earlier in this section to give a proof for Theorem 5.2. Let Δ be the assortment defined in (5.3). The goal is to show that the solution Δ is feasible to CAP and its expected revenue satisfies $f(\Delta) = \Omega(1/\log m) \cdot z_{\mathsf{CAP}}$. We will follow the following steps in our proof. First, we will show that the solution Δ is feasible for CAP. Second, we will show that the expected revenue $f(\Delta)$ corresponding to the solution is at least as large as the optimal objective value of problem (5.2). Thus, it is sufficient to show that the optimal objective value of problem (5.2) is $\Omega(1/\log m) \cdot z_{\mathsf{CAP}}$. To achieve the latter goal, building on Lemma F.1 and Lemma F.2, we will construct a feasible solution for problem (5.2) and establish that this feasible solution has an objective value of at least $\Omega(1/\log m) \cdot z_{\mathsf{CAP}}$.

Feasibility. Let $\{x^*_{Tk} : T \in \mathcal{L}, k = 1, \ldots, K\}$ be the optimal solution of problem (5.2) and Δ be the assortment defined in (5.3). Recall Δ_{Tk} in the definition of Δ in (5.3) is given by $\mathsf{AugGreedy}(\mathcal{C}_T, k)$, so Δ_{Tk} has at most k products. Therefore,

$$|\Delta| \leq \sum_{T \in \mathcal{L}} \sum_{k=1}^{K} |\Delta_{Tk}| \cdot x^*_{Tk} \leq \sum_{T \in \mathcal{L}} \sum_{k=1}^{K} k \cdot x^*_{Tk} \leq K,$$

where the last inequality follows from the first constraint of problem (5.2). Hence, Δ is a feasible solution for CAP.

Lower bound for $f(\Delta)$. By the definition of $\mathcal{L}$, if $T, T' \in \mathcal{L}$ with $T \cap T' \neq \varnothing$, then either $T \subseteq T'$ or $T' \subseteq T$. Hence, the second constraint of (5.2) ensures that the subtrees $\{T \in \mathcal{L} : \sum_{k=1}^{K} x^*_{Tk} = 1\}$ are disjoint, i.e., if there exists $T, T' \in \mathcal{L}$ such that $\sum_{k=1}^{K} x^*_{Tk} = 1$ and $\sum_{k=1}^{K} x^*_{T'k} = 1$, then $T \cap T' = \varnothing$. Therefore, the sets $\{\mathcal{C}_T : \sum_{k=1}^{K} x^*_{Tk} = 1\}$ for $T \in \mathcal{L}$ are disjoint subsets of $\mathcal{M}$. Thus,

$$f(\Delta) = f^{\mathcal{M}}(\Delta) \geq \sum_{T \in \mathcal{L}} f^{\mathcal{C}_T}(\Delta) \cdot \sum_{k=1}^{K} x^*_{Tk} \geq \sum_{T \in \mathcal{L}} \sum_{k=1}^{K} f^{\mathcal{C}_T}(\Delta_{Tk}) \cdot x^*_{Tk} = \sum_{T \in \mathcal{L}} \sum_{k=1}^{K} R_{Tk} \cdot x^*_{Tk},$$

where the second inequality uses the monotonicity of $f^{\mathcal{C}_T}$ and $\Delta_{Tk} \subseteq \Delta$ for $x^*_{Tk} = 1$. Thus, to prove Theorem 5.2, we can show that the optimal objective value of (5.2) is $\Omega(1/\log m) \cdot z_{\mathsf{CAP}}$.

Feasible solution for problem (5.2)**.** We construct a solution for problem (5.2) with an objective value of $\Omega(1/\log m) \cdot z_{\mathsf{CAP}}$, which implies that the optimal objective value of problem (5.2) is $\Omega(1/\log m) \cdot z_{\mathsf{CAP}}$. Our construction is as follows. We initialize all the variables of problem (5.2) at 0. Consider the set $\mathcal{X}$ given in Lemma F.1. Let us start at the root of the tree $\mathcal{T}$. For ease of notation, let $\alpha_{\mathcal{T}} = h^{\mathcal{X} \cap \mathcal{C}_{\mathcal{T}}}(S^*) = h^{\mathcal{X}}(S^*)$. Initially, we apply Lemma F.2 with the pair $(S^*, \mathcal{T})$. We know that at least one of the two statements of Lemma F.2 should be true. Suppose, the first statement is true, i.e., $R_{\mathcal{T}K} = \Omega(1/\log m) \cdot h^{\mathcal{X}}(S^*)$, in which case, we get

$$R_{\mathcal{T}K} = \Omega(1/\log m) \cdot \alpha_{\mathcal{T}}.$$

In that case, we let $x_{\mathcal{T}K} = 1$, truncate all the descendant nodes after the root and stop.

If the first statement of Lemma F.2 for the pair $(S^*, \mathcal{T})$ is not true, then the second statement must be true, i.e., there exists a partition of $S^* = A_1 \cup A_2$ and $A_1 \cap A_2 = \varnothing$ such that

$$h^{\mathcal{X} \cap \mathcal{C}_{T_1}}(A_1) + h^{\mathcal{X} \cap \mathcal{C}_{T_2}}(A_2) \geq \left(1 - \frac{1}{\log m}\right) \cdot h^{\mathcal{X}}(S^*),$$

where T_1 is the left subtree of $\mathcal{T}$ and T_2 is the right subtree of $\mathcal{T}$. We let $\alpha_{T_1} = h^{\mathcal{X} \cap \mathcal{C}_{T_1}}(A_1)$ and $\alpha_{T_2} = h^{\mathcal{X} \cap \mathcal{C}_{T_2}}(A_2)$. Hence, we have

$$\alpha_{T_1} + \alpha_{T_2} \geq \left(1 - \frac{1}{\log m}\right) \cdot \alpha_{\mathcal{T}}.$$

We repeat the same argument for the pairs (A_1, T_1) and (A_2, T_2). In general, each time we consider a pair (A, T), we let $\alpha_T = h^{\mathcal{X} \cap \mathcal{C}_T}(A)$. If the first statement of Lemma F.2 is true for the pair (A, T), then we set $x_{Tk} = 1$ for $k = |A|$ and *truncate* all the descendant nodes of the root of T, i.e., we do not further explore T. By Lemma F.2, we know that

$$R_{Tk} = \Omega(1/\log m) \cdot \alpha_T. \tag{F.4}$$

Otherwise, if the first statement is not true, then the second statement of the lemma must be true, so there exists a partition of $A = A_1 \cup A_2$ and $A_1 \cap A_2 = \varnothing$ such that $h^{\mathcal{X} \cap \mathcal{C}_{T_1}}(A_1) + h^{\mathcal{X} \cap \mathcal{C}_{T_2}}(A_2) \geq (1 - \frac{1}{\log m}) \cdot h^{\mathcal{X} \cap \mathcal{C}_T}(A)$, where T_1 is the left subtree of T and T_2 is the right subtree of T. Letting $\alpha_{T_1} = h^{\mathcal{X} \cap \mathcal{C}_{T_1}}(A_1)$ and $\alpha_{T_2} = h^{\mathcal{X} \cap \mathcal{C}_{T_2}}(A_2)$, we have

$$\alpha_{T_1} + \alpha_{T_2} \geq \left(1 - \frac{1}{\log m}\right) \cdot \alpha_T. \tag{F.5}$$

We repeat the same argument again for (A_1, T_1) and (A_2, T_2). Note that at each node, we either truncate the tree or move to the next level. If we arrive at a pair (A, T) where the subtree T is simply a leaf of $\mathcal{T}$, then we know from the proof of Lemma F.2 that the first statement should be true for this pair and therefore (F.4) is verified for T.

Let $\{x_{Tk} : T \in \mathcal{L},\ k = 1, \ldots, K\}$ be the solution that we have constructed as above. Let us show that this solution is feasible for problem (5.2). Consider the pairs (A, T) for which we set $x_{Tk} = 1$ where $k = |A|$. Each pair (A, T) among these pairs is such that T correspond to a red node in Figure 2. Let us index them with (A_i, T_i) for $i \in \mathcal{I}$, where $\mathcal{I}$ is an index set. By our construction,

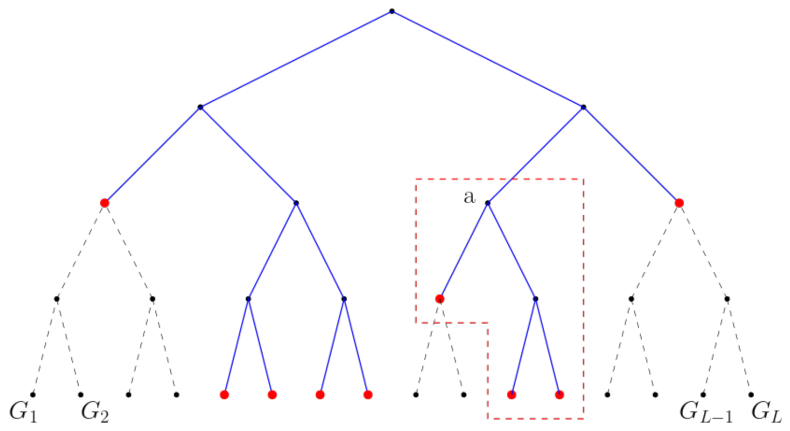

Figure 2: Truncated Tree of customer types.

the subsets A_i for $i \in \mathcal{I}$ form a partition of S^*. In particular, we have

$$\sum_{k=1}^{K} \sum_{T \in \mathcal{L}} k \cdot x_{Tk} = \sum_{i \in \mathcal{I}} |A_i| = |S^*| \leq K.$$

Therefore, our solution verifies the first constraint of problem (5.2). Moreover, if $x_{T_i k_i} = 1$ for some $i \in \mathcal{I}$, then our construction implies that $x_{T_i k} = 0$ for any $k \neq |A_i|$. The tree was truncated after the node that corresponds to the subtree T_i, hence $x_{Tk} = 0$ for any subtree T that correspond to a descendant node of T_i and for any $k = 1, \ldots, K$. Furthermore, the subtrees T that correspond to the parent nodes of T_i are such that $X_{Tk} = 0$ for any $k = 1, \ldots, K$. Thus, it follows that $\sum_{k=1}^{K}(x_{Tk} + x_{T'k}) \leq 1$ for all $T \in \mathcal{L}$, $T' \subseteq T$, $T' \neq T$, which implies that our solution satisfies the second constraint of problem (5.2) as well.

Performance guarantee. To complete our proof, let us show that the solution constructed above provides an expected revenue of $\Omega(1/\log m) \cdot z_{\mathsf{CAP}}$ for problem (5.2). Consider a node in the truncated tree and let T be the corresponding subtree in $\mathcal{L}$. Let $\mathsf{depth}(T)$ be the depth of T in the truncated tree and $\mathsf{leaves}(T)$ be the leaves of T in the truncated tree. The depth of a subtree is defined as the distance between its root and the farthest leaf. By convention, the distance between two consecutive levels is 1. Note that $\mathsf{leaves}(T)$ must be among the red nodes in Figure 2. For instance, in Figure 2, the dashed box shows the subtree that corresponds to node a. The subtree $T \in \mathcal{L}$ that corresponds to node a in the original tree has four leaves. The subtree that corresponds to node a in the truncated tree has three leaves labeled as red nodes, i.e., $|\mathsf{leaves}(T)| = 3$. The depth of T in the truncated tree is $\mathsf{depth}(T) = 2$. First, let us show that

$$\sum_{T' \in \mathsf{leaves}(T)} \alpha_{T'} \geq \left(1 - \frac{1}{\log m}\right)^{\mathsf{depth}(T)} \alpha_T. \tag{F.6}$$

We show the above inequality using induction on $\mathsf{depth}(T)$.

Base case. Consider a node in the truncated tree and let $T \in \mathcal{L}$ be the corresponding subtree. Assume that T has depth one in the truncated tree. Hence, T has exactly two leaves and both are red nodes. Therefore, (F.6) follows directly from (F.5).

Induction. Suppose that the inequality in (F.6) holds for the subtrees with depth d in the truncated tree. Consider a subtree $T \in \mathcal{L}$ that has depth $d+1$ in the truncated tree. Let T_1 be its left subtree and T_2 be its right subtree. By (F.5), we have

$$\alpha_{T_1} + \alpha_{T_2} \geq (1 - \frac{1}{\log m}) \cdot \alpha_T.$$

Since, T_1 and T_2 have depth d, by the induction hypothesis, for $i \in \{1, 2\}$, we have

$$\sum_{T' \in \mathsf{leaves}(T_i)} \alpha_{T'} \geq \left(1 - \frac{1}{\log m}\right)^d \alpha_{T_i}.$$

Therefore,

$$\sum_{T' \in \mathsf{leaves}(T)} \alpha_{T'} = \sum_{T' \in \mathsf{leaves}(T_1)} \alpha_{T'} + \sum_{T' \in \mathsf{leaves}(T_2)} \alpha_{T'} \geq \left(1 - \frac{1}{\log m}\right)^d (\alpha_{T_1} + \alpha_{T_2}) \geq \left(1 - \frac{1}{\log m}\right)^{d+1} \alpha_T,$$

which concludes the induction.

Applying the inequality in (F.6) to the full tree $\mathcal{T}$, we get

$$\sum_{T' \in \mathsf{leaves}(\mathcal{T})} \alpha_{T'} \geq \left(1 - \frac{1}{\log m}\right)^{\mathsf{depth}(\mathcal{T})} \alpha_{\mathcal{T}}.$$

We know that the depth of $\mathcal{T}$ is at most $\log m$. Hence,

$$\sum_{T' \in \mathsf{leaves}(\mathcal{T})} \alpha_{T'} \geq \left(1 - \frac{1}{\log m}\right)^{\log m} \alpha_{\mathcal{T}} = \Theta(e^{-1}) \cdot \alpha_{\mathcal{T}}, \tag{F.7}$$

where the last equality holds for sufficiently large m. Finally, each leaf node in the truncated tree, which are the red nodes in Figure 2, verifies (F.4), i.e., for all $i \in \mathcal{I}$, $R_{T_i,k_i} = \Omega(1/\log m) \cdot \alpha_{T_i}$. Thus, we have

$$\sum_{i \in \mathcal{I}} R_{T_i k_i} = \Omega(1/\log m) \sum_{T' \in \mathsf{leaves}(\mathcal{T})} \alpha_{T'}. \tag{F.8}$$

Moreover, we have

$$\alpha_{\mathcal{T}} = h^{\mathcal{X}}(S^*) = \sum_{j \in \mathcal{X}} \theta_j \frac{\sum_{i \in S^* \cap S_j^*} r_i v_{ij}}{1 + \sum_{i \in S^* \cap S_j^*} v_{ij}} = f^{\mathcal{X}}(S^*) \geq \frac{1}{4} \cdot z_{\mathsf{CAP}}, \tag{F.9}$$

where the last inequality follows from Lemma F.1 . Therefore, from (F.7), (F.8) and (F.9), we get

$$\sum_{i \in \mathcal{I}} R_{T_i k_i} = \Omega(1/\log m) \cdot z_{\mathsf{CAP}},$$

i.e., our solution provides an expected revenue of $\Omega(1/\log m) \cdot z_{\mathsf{CAP}}$. This concludes our proof.

G Appendix: Proof of Theorem 6.1

In this section, we give a proof for Theorem 6.1. In the next lemma, we give a performance guarantee for CAP that is provided by an optimal solution to problem (6.2).

Lemma G.1. *Let z_{CAP} be the optimal objective value of* CAP *and $(\hat{S}, \hat{\boldsymbol{t}})$ be an optimal solution to problem (6.2). Then, we have*

$$\sum_{j \in \mathcal{M}} \theta_j \, f_j(\hat{S}) \geq (1 - 3\epsilon) \cdot z_{\mathsf{CAP}}.$$

Proof. Let S^* be an optimal solution to CAP. By the definition of Grid, for each $j \in \mathcal{M}$, there exists $t_j^* \in \mathsf{Grid}$ such that $t_j^* \leq f_j(S^*) \leq (1+\epsilon)\, t_j^*$. We claim that $(S^*, \boldsymbol{t}^*)$ with $\boldsymbol{t}^* = (t_1^*, \ldots, t_m^*)$ is a feasible solution to problem (6.2). In particular, by the discussion at the beginning of Section 6, we have $f_j(S) \geq t_j$ if and only if $\sum_{i \in S} v_{ij}\,(r_i - t_j)^+ \geq t_j$. Thus, since $f_j(S^*) \geq t_j^*$, it follows that $\sum_{i \in S^*} v_{ij}\,(r_i - t_j^*)^+ \geq t_j^*$, which we write equivalently as

$$\sum_{i \in S^*} \frac{v_{ij}\,(r_i - t_j^*)^+}{t_j^*} \cdot \frac{n}{\epsilon} \geq \frac{n}{\epsilon},$$

in which case, the inequality above yields

$$\left\lfloor \frac{n}{\epsilon} \right\rfloor \leq \sum_{i \in S^*} \left\lceil \frac{v_{ij}\,(r_i - t_j^*)^+}{t_j^*} \cdot \frac{n}{\epsilon} \right\rceil = \sum_{i \in S^*} \sigma_{ij}(t_j^*).$$

Thus, the solution $(S^*, \boldsymbol{t}^*)$ satisfies the first constraint in problem (6.2). Since S^* is an optimal solution to CAP, we have $|S^*| \leq K$, which implies that the second constraint in problem (6.2) is also satisfied. By our choice of t_j^*, we have $\boldsymbol{t}^* \in \mathsf{Grid}^m$ as well. Therefore, the claim follows and $(S^*, \boldsymbol{t}^*)$ is feasible to problem (6.2).

Next, recalling that $(\hat{S}, \hat{\boldsymbol{t}})$ is an optimal solution to problem (6.2), they satisfy the first constraint in this problem. In this case, using the fact that $\lfloor a \rfloor \geq a - 1$ and $\lceil a \rceil \leq a + 1$, we have

$$\begin{aligned}
\frac{n}{\epsilon} - 1 \leq \left\lfloor \frac{n}{\epsilon} \right\rfloor \leq \sum_{i \in \hat{S}} \sigma_{ij}(\hat{t}_j) = \sum_{i \in \hat{S}} \left\lceil \frac{v_{ij}\,(r_i - \hat{t}_j)^+}{\hat{t}_j} \cdot \frac{n}{\epsilon} \right\rceil \\
\leq \sum_{i \in \hat{S}} \left(\frac{v_{ij}\,(r_i - \hat{t}_j)^+}{\hat{t}_j} \cdot \frac{n}{\epsilon} + 1 \right) \leq \sum_{i \in \hat{S}} \left(\frac{v_{ij}\,(r_i - \hat{t}_j)^+}{\hat{t}_j} \cdot \frac{n}{\epsilon} \right) + n,
\end{aligned}$$

where the last inequality holds because $|\hat{S}| \leq n$. Arranging the terms in the inequality above, we obtain

$$\sum_{i \in \hat{S}} \frac{v_{ij}\,(r_i - \hat{t}_j)^+}{\hat{t}_j} \geq \frac{\epsilon}{n} \left(\frac{n}{\epsilon} - 1 - n \right) \geq 1 - 2\epsilon.$$

Thus, we obtain $(1 - 2\epsilon)\,\hat{t}_j \leq \sum_{i \in \hat{S}} v_{ij}\,(r_i - \hat{t}_j)^+ \leq \sum_{i \in \hat{S}} v_{ij}\,(r_i - (1-2\epsilon)\,\hat{t}_j)^+$. Once again, by the discussion at the beginning of Section 6, we have $f_j(S) \geq t_j$ if and only if $\sum_{i \in S} v_{ij}\,(r_i - t_j)^+ \geq t_j$.

Thus, the last inequality implies that $f_j(\hat{S}) \geq (1-2\epsilon)\,\hat{t}_j$ for each $j \in \mathcal{M}$. In this case, we obtain the chain of inequalities

$$\sum_{j\in\mathcal{M}} \theta_j\, f_j(\hat{S}) \geq (1-2\epsilon) \sum_{j\in\mathcal{M}} \theta_j\, \hat{t}_j \geq (1-2\epsilon) \sum_{j\in\mathcal{M}} \theta_j\, t_j^* \geq \frac{1-2\epsilon}{1+\epsilon} \sum_{j\in\mathcal{M}} \theta_j\, f_j(S^*) = \frac{1-2\epsilon}{1+\epsilon} z_{\mathsf{CAP}},$$

where the second inequality holds because $(\hat{S}, \hat{\boldsymbol{t}})$ is is an optimal solution to problem (6.2), but $(S^*, \boldsymbol{t}^*)$ is only a feasible solution as shown in the claim at the beginning of the proof, and the third inequality holds by noting that $(1+\epsilon)\, t_j^* \geq f_j(S^*)$ by our choice of t_j^*. Finally, we get the desired result because $\frac{1-2\epsilon}{1+\epsilon} \geq 1 - 3\epsilon$. □

By the lemma above, for any given $\delta \in (0,1)$, if we construct the geometric grid in problem (6.2) with an accuracy parameter of $\epsilon = \delta/3$, then an optimal solution to this problem is a $(1-\delta)$-approximation to CAP. We use this observation to give a proof for Theorem 6.1.

Proof of Theorem 6.1. For any $\delta \in (0,1)$, we construct the geometric grid in problem (6.2) with an accuracy parameter of $\epsilon = \delta/3$. By the discussion in the previous paragraph, an optimal solution to problem (6.2) is a $(1-\delta)$-approximation to CAP.

To show Theorem 6.1, it remains to argue that we can obtain an optimal solution to problem (6.2) in running time $O\big(\log^m(\frac{B_{\max}}{B_{\min}})\, \frac{n^{m+2}}{\delta^{2m}}\big)$. We obtain an optimal solution to problem (6.2) as follows. By the definition of Grid, the number of points in Grid is

$$O\left(\frac{\log(B_{\max}/B_{\min})}{\log(1+\epsilon)}\right) = O\left(\frac{\log(B_{\max}/B_{\min})}{\epsilon}\right),$$

which yields a total of $O\big(\frac{\log^m(B_{\max}/B_{\min})}{\epsilon^m}\big)$ lower bound guesses $\boldsymbol{t} \in \mathsf{Grid}^m$. For each value of $\boldsymbol{t} \in \mathsf{Grid}^m$, we can use the dynamic program in (6.3) to check whether there exists an assortment S that satisfies the constraints in problem (6.2).

We proceed to accounting for the running time for solving the dynamic program. All components of the state variable $(q_1, \ldots, q_m, \ell)$ in (6.3) take on integer values. The value of the state variable q_j only increases from one decision epoch to the next. Thus, if $q_j \geq \lfloor \frac{n}{\epsilon} \rfloor$ at any decision epoch, then we can set the value of this state variable to $\lfloor \frac{n}{\epsilon} \rfloor$ at all future decision epochs because the boundary condition only depends on whether q_j exceeds $\lfloor \frac{n}{\epsilon} \rfloor$. This discussion indicates that there are $O(\frac{n}{\epsilon})$ possible values for the state variable q_j. Moreover, the state variable ℓ takes on $O(n)$ possible values. Thus, there are $O((\frac{n}{\epsilon})^m n)$ possible values for the state variable $(q_1, \ldots, q_m, \ell)$, so noting that there are n decision epochs in the dynamic program, for fixed $\boldsymbol{t} \in \mathsf{Grid}^m$, we can solve the dynamic program in (6.3) in $O((\frac{n}{\epsilon})^m\, n^2)$ operations.

Putting the discussion in the previous two paragraphs together, we can obtain an optimal solution to problem (6.2) by checking each value of $\boldsymbol{t} \in \mathsf{Grid}^m$, and for each $\boldsymbol{t} \in \mathsf{Grid}^m$, using the dynamic program in (6.3) to see whether there exists an assortment S that satisfies the constraints of problem (6.2). The total number of operations to do so is

$$O\left(\frac{\log^m(B_{\max}/B_{\min})}{\epsilon^m} \left(\frac{n}{\epsilon}\right)^m n^2\right) = O\left(\log^m\!\left(\frac{B_{\max}}{B_{\min}}\right) \frac{n^{m+2}}{\epsilon^{2m}}\right).$$

Evaluating the running time above with $\epsilon = \delta/3$ yields the desired result. □

H Appendix: Integer programming formulation

Let $\{x_i^* : i \in \mathcal{N}\}$ and $\{z_{j\ell}^* : j \in \mathcal{M},\ \ell = 1, \ldots, L\}$ be an optimal solution to problem (6.5). Using this optimal solution, define the assortment $S^* = \{i \in \mathcal{N} : x_i^* = 1\}$. Noting the second constraint in problem (6.5), we have $|S^*| \leq K$. Letting z_{IP} be the optimal objective value of problem (6.5), in this section, we show that

$$\sum_{j \in \mathcal{M}} \theta_j \, f_j(S^*) \ \geq \ z_{\mathsf{IP}} \ \geq \ \frac{1}{1+\epsilon} \, z_{\mathsf{CAP}}.$$

Thus, the solution S^* is a $(1-\epsilon)$-approximation to CAP and $(1+\epsilon)\, z_{\mathsf{IP}}$ is an upper bound on the optimal objective value of CAP.

We start by arguing that $\sum_{\in \mathcal{M}} \theta_j \, f_j(S^*) \geq z_{\mathsf{IP}}$. For each $j \in \mathcal{M}$, let $k_j = 1, \ldots, L$ be such that $z_{j,k_j}^* = 1$. Thus, the first constraint in problem (6.5) implies that $\sum_{i \in \mathcal{N}} v_{ij} \, (r_i - \tau^{k_j})^+ \, x_i^* \geq \tau^{k_j}$, which, by using the definition of S^*, is equivalent to

$$\sum_{i \in S^*} v_{ij} \, (r_i - \tau^{k_j})^+ \geq \tau^{k_j}.$$

By the discussion at the beginning of Section 6, $f_j(S) \geq t_j$ if and only if $\sum_{i \in S} v_{ij} \, (r_i - t_j)^+ \geq t_j$. Thus, the inequality above implies that $f_j(S^*) \geq \tau^{k_j}$ for each $j \in \mathcal{M}$. In this case, we obtain the chain of inequalities

$$\sum_{j \in \mathcal{M}} \theta_j \, f_j(S^*) \geq \sum_{j \in \mathcal{M}} \theta_j \, \tau^{k_j} = \sum_{j \in \mathcal{M}} \theta_j \sum_{\ell=1}^{L} \tau^{\ell} \, z_{j\ell}^* = z_{\mathsf{IP}},$$

where the first equality uses the definition of k_j.

Next, we argue that $z_{\mathsf{IP}} \geq \frac{1}{1+\epsilon} \, z_{\mathsf{CAP}}$. Let $\hat{S}$ be an optimal solution to CAP. For each $i \in \mathcal{N}$, define $\hat{x}_i \in \{0, 1\}$ as $\hat{x}_i = 1$ if and only if $i \in \hat{S}$. For each $j \in \mathcal{M}$, let $\hat{k}_j = 1, \ldots, L$ be such that

$$\tau^{\hat{k}_j} \leq f_j(\hat{S}) \leq (1+\epsilon) \, \tau^{\hat{k}_j}.$$

By the definition of Grid, such $\hat{k}_j$ is guaranteed to exist. Lastly, for each $j \in \mathcal{M}$ and $\ell = 1, \ldots, L$, define $\hat{z}_{j\ell} \in \{0, 1\}$ as $\hat{z}_{j\ell} = 1$ if and only if $\ell = \hat{k}_j$. If we can show that $\{\hat{x}_i : i \in \mathcal{N}\}$ and $\{\hat{z}_{j\ell} : j \in \mathcal{M},\ \ell = 1, \ldots, L\}$ is a feasible solution to problem (6.5), then it follows that

$$z_{\mathsf{IP}} \geq \sum_{j \in \mathcal{M}} \theta_j \sum_{\ell=1}^{L} \tau^{\ell} \hat{z}_{j\ell} = \sum_{j \in \mathcal{M}} \theta_j \, \tau^{\hat{k}_j} \geq \frac{1}{1+\epsilon} \sum_{j \in \mathcal{M}} \theta_j \, f_j(\hat{S}) = \frac{1}{1+\epsilon} \, z_{\mathsf{CAP}},$$

where the first inequality holds because $\{\hat{x}_i : i \in \mathcal{N}\}$ and $\{\hat{z}_{j\ell} : j \in \mathcal{M},\ \ell = 1, \ldots, L\}$ is a feasible, but not necessarily an optimal, solution to problem (6.5), whereas the second inequality holds because we chose $\hat{k}_j$ to satisfy $f_j(\hat{S}) \leq (1+\epsilon) \, \tau^{\hat{k}_j}$. The chain of inequalities yields $z_{\mathsf{IP}} \geq \frac{1}{1+\epsilon} \, z_{\mathsf{CAP}}$, which is the desired result.

It remains to show that $\{\hat{x}_i : i \in \mathcal{N}\}$ and $\{\hat{z}_{j\ell} : j \in \mathcal{M},\ \ell = 1, \ldots, L\}$ is a feasible solution to problem (6.5). Once again, by the discussion at the beginning of Section 6, $f_j(S) \geq t_j$ if and only if $\sum_{i\in S} v_{ij}\,(r_i - t_j)^+ \geq t_j$. Therefore, noting that $f_j(\hat{S}) \geq \tau^{\hat{k}_j}$ by our choice of $\hat{k}_j$, we get $\sum_{i\in\hat{S}} v_{ij}\,(r_i - \tau^{\hat{k}_j})^+ \geq \tau^{\hat{k}^j}$. We equivalently write the last inequality as

$$\sum_{i\in\mathcal{N}} v_{ij}\,(r_i - \tau^{\hat{k}_j})^+\,\hat{x}_i = \sum_{i\in\hat{S}} v_{ij}\,(r_i - \tau^{\hat{k}_j})^+ \geq \tau^{\hat{k}^j} = \tau^{\hat{k}^j}\,\hat{z}_{j,\hat{k}_j},$$

where the first equality uses the definition of $\hat{x}_i$ and the last equality uses the definition of $\hat{z}_{j\ell}$. Thus, the solution $\{\hat{x}_i : i\in\mathcal{N}\}$ and $\{\hat{z}_{j\ell} : j\in\mathcal{M},\ \ell = 1,\ldots,L\}$ satisfies the first constraint in problem (6.5) for each $j\in\mathcal{M}$ and $\ell = \hat{k}_j$. When $\ell \neq \hat{k}_j$, $\hat{z}_{j\ell} = 0$, in which case, the right side of the first constraint takes value zero. Therefore, the solution $\{\hat{x}_i : i\in\mathcal{N}\}$ and $\{\hat{z}_{j\ell} : j\in\mathcal{M},\ \ell = 1,\ldots,L\}$ satisfies the first constraint in problem (6.5) for each $j\in\mathcal{M}$ and $\ell\neq\hat{k}_j$ as well.

Since $\hat{S}$ is an optimal solution to CAP, we have $|\hat{S}|\leq K$, but by the definition of $\hat{x}_i$, we have $\sum_{i\in\mathcal{N}}\hat{x}_i = |\hat{S}|$. Thus, $\{\hat{x}_i : i\in\mathcal{N}\}$ and $\{\hat{z}_{j\ell} : j\in\mathcal{M},\ \ell = 1,\ldots,L\}$ satisfies the second constraint in problem (6.5). Lastly, by the definition of $\hat{z}_{j\ell}$, we have $\sum_{\ell=1}^{L}\hat{z}_{j\ell} = \hat{z}_{j,\hat{k}_j} = 1$, so the third constraint in problem (6.5) is satisfied as well.

I Appendix: Numerical experiments for heterogeneous arrival probabilities

In this section, we provide computational experiments under heterogeneous arrival probabilities for different customer types. Our results are in Table 8. The format of this table is identical to that of Table 7. Augmented Greedy continues to provide near-optimal assortments with less than 1% optimality gaps in an overwhelming majority of our problem instances.

Augmented Greedy

			γ_{1t} (%)		
m	n	K	Mean	5%	Min
10	50	5	98.6	97.1	96.9
		25	98.5	96.2	95.5
	100	10	99.3	99.1	99.0
		50	99.3	99.2	99.2
50	50	5	99.5	99.4	99.4
		25	99.4	99.2	99.1
	100	10	99.5	99.5	99.4
		50	99.5	99.4	99.4
100	50	5	99.5	99.5	99.4
		25	99.5	99.5	99.5
	100	10	99.5	99.5	99.5
		50	99.5	99.5	99.5

Integer Program

			γ_{2t} (%)		
m	n	K	Mean	5%	Min
10	50	5	99.4	99.4	99.4
		25	99.5	99.4	99.4
	100	10	99.5	99.4	99.4
		50	99.5	99.4	99.4
50	50	5	99.5	99.4	99.4
		25	99.5	99.4	99.4
	100	10	99.5	99.5	99.5
		50	99.5	99.4	99.4
100	50	5	99.5	99.5	99.5
		25	99.5	99.5	99.5
	100	10	99.5	99.5	99.5
		50	99.5	99.5	99.5

Table 8: Quality of the solutions from Augmented Greedy and the integer programming formulation for problem instances with heterogeneous arrival probabilities.