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

Omar El Housni   Huseyin Topaloglu

School of Operations Research and Information Engineering, Cornell Tech, New York, NY 10044, USA
{oe46,ht88}@cornell.edu

April 19, 2021

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 to carry and a customized assortment for 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 could potentially customize the displayed assortments 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 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. Lastly, we present a fully polynomial-time approximation scheme for CAP when the number of customer types is constant. 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

Use of discrete choice models to capture the customer choice process has steadily been seeing increased attention in revenue management. Discrete choice models allow us to capture 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 a tremendous amount of browsing and purchasing customer 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 could 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 for 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, but the grocery choices of different customer segments dramatically differ from each other. Thus, they have the opportunity to adjust the assortment offered to each customer based on what is known about the segment of the customer. Often times, online grocers operate warehouses close to 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

In our work, we characterize the complexity of CAP and give tight bounds on the value of customization. As our main technical contribution, we develop approximation algorithms for CAP. Here are our contributions:

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 (Theorem 2.1). In view of this hardness result, we will focus on developing approximation algorithms for CAP.

Value of customization. We show that CAP, by customizing assortment for each customer type, can substantially increase the expected revenues obtained by MMNL, which does not customize the assortment. 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 of CAP cannot exceed the expected revenue of 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 constitutes the building block of our algorithmic framework. Augmented Greedy considers a small number of subsets of products based on the ranking of their 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 6.1). Our 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 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 hard to approximate than MMNL and our work closes this gap.

Fully polynomial-time approximation scheme. We give an FPTAS for CAP when the number of customer types m is constant (Theorem 7.2). Our FPTAS is based on using a geometric grid to guess the expected revenue obtained from each customer type and using a dynamic program to find a first stage assortment that allows us to reach the guessed expected revenues. Moreover, we show that CAP is NP-Hard even with two customer types. Thus, FPTAS is the best approximation guarantee we could aim for under a constant number of customer types.

Computational study. We use a dataset from Expedia to demonstrate the value of customization. Our computational study shows that customization allows achieving significantly higher expected revenues when compared to offering a single assortment to all customers without customization. In addition, we check the practical performance of our approximation algorithms on randomly generated problem instances. Our approximation algorithms perform remarkably well, obtaining near-optimal solutions for an overwhelming majority of instances. We also compare the performance of our approximation algorithm with a standard greedy algorithm to demonstrate that our algorithmic framework can, in practice, provide improvements over standard greedy schemes.

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 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 improves the assortment on hand by adding or removing products.

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.

Outline. 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 benefits of customization. In Section 4, we describe the Augmented Greedy algorithm. In Section 5, we show that Augmented Greedy provides $\Omega(1/\log m)$-approximation guarantee under equal arrival probabilities for the customer types. In Section 6, we extend this result for general customer arrival probabilities. In Section 7, we develop our FPTAS. In Section 8, we give computational experiments to check the value of customization on a dataset from Expedia, whereas in Section 9, we check the practical performance of our approximation algorithm.

2 Problem formulation and complexity

In this section, we give the mathematical formulation of our model and characterize its complexity. We consider a set of products $\mathcal{N} = \{1, \ldots, n\}$. 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 model (MNL). Let v_{ij} denote the preference weight of customer type j for product i. For all customer types, we normalize the preference weights of the no-purchase option to one. In particular, 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 $v_{ij}/(1 + \sum_{\ell \in S_j} v_{\ell j})$ and the expected revenue we obtain from her 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, the decision maker needs to select at most K products. We use the set variable $S \subseteq \mathcal{N}$ to denote these products. In the second stage, the decision maker observes the type of the arriving customer and offers her a personalized assortment, which is a subset of the products S carried initially. We use the set variable 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 a set 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 such that the expected revenue over all customer types is maximized. We refer to this problem as the *Customized Assortment Problem* (CAP). In particular, we want to solve the problem:

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

Note that for a given S, we can efficiently compute $f_j(S)$ for all $j \in \mathcal{M}$. This is a classical result in assortment optimization under MNL (Gallego et al., 2004; Talluri and van Ryzin, 2004).

In the next theorem, we provide an inapproximability result for CAP. 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 the customer type arrival probabilities are all equal. 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 (Computational Complexity). *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 $r_i = r_\ell, \forall i, \ell \in \mathcal{N}$ and $\theta_j = \theta_q$, $\forall j, q \in \mathcal{M}$.*

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

In this section, we show the value of customization in our model by comparing CAP to a model where the decision-maker offers the selected products in the first stage to all customer types without customization. First, let us write CAP as follows,

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

The problem without customization corresponds to the classical assortment optimization problem under the mixture of multinomial logit models with a cardinality constraint. By a slight abuse of wording, we refer to this problem as MMNL and it can be formulated as follows,

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

Note that CAP is a relaxation of MMNL and therefore the optimal expected revenue of CAP is always greater than the optimal expected revenue of MMNL. On the other hand, 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 greater than the optimal expected revenue of

MMNL (Theorem 3.1). This shows the power of customization since the revenue of our model can be significantly higher as 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, $z_{\mathsf{MMNL}} \leq z_{\mathsf{CAP}} \leq m \cdot z_{\mathsf{MMNL}}$. Moreover, there are instances such that $z_{\mathsf{CAP}} = \Omega(m) \cdot z_{\mathsf{MMNL}}$.*

Proof. The inequality $z_{\mathsf{MMNL}} \leq z_{\mathsf{CAP}}$ is straightforward because CAP is a relaxation of MMNL. To show the other inequality, 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. Let $q \in \mathcal{M}$ be the customer type with highest $\theta_j f_j(S^*)$ for $j \in \mathcal{M}$, i.e., $\theta_j f_j(S^*) \leq \theta_q f_q(S^*)$ for all $j \in \mathcal{M}$. Therefore, $z_{\mathsf{CAP}} \leq m \cdot \theta_q f_q(S^*)$. Moreover, since $S_q^* \subseteq S^*$, then $|S_q^*| \leq K$ and therefore S_q^* is a feasible solution for MMNL. Hence,

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

Therefore, $z_{\mathsf{CAP}} \leq m \cdot z_{\mathsf{MMNL}}$. To show the tightness of the bound, we present the following example. We consider an instance where the number of products n is equal to the number of customer types m and equal to the cardinality K, i.e., $n = m = K$. We will use m to denote all three parameters in the rest of the example. Let us consider the following instance,

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

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

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

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

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 $b \geq 1$.

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

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

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 in (3.1) as follows,

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

where the last inequality holds because $\sum_{i=1}^{j-1} a^i = \frac{a^j-a}{a-1} \le 2a^{j-1}$ for $a \ge 2$. We bound the second term in (3.1) as follows,

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

where ℓ_S is the smallest index in S. Moreover,

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

Therefore,

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

By choosing for instance , $b=m-1$ and $a = 2m(m-1)^m$, we get $\sum_{j=1}^m \theta_j\cdot\mathsf{Rev}_j(S)\le 3\alpha$, for any assortment $S\subseteq\mathcal{N}$. Hence, $z_{\mathsf{MMNL}} \le 3\alpha$ and therefore, $z_{\mathsf{CAP}} = \Omega(m)\cdot z_{\mathsf{MMNL}}$. □

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

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

Let us define,

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

Recall that CAP is 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 $A \subseteq B \subseteq \mathcal{N}$, since we obtain more revenue if we select more products in CAP. However, f is not submodular[1]. In fact, Nemhauser et al. (1978) show that a greedy algorithm, that picks iteratively the element providing the highest 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. However, our function f is not submodular in general and therefore this classical result does not apply to our problem. We give in Appendix B an example with only one customer type and three products to show that f is not submodular. This motivates us to develop a more general algorithm for approximating CAP.

4.1 Description of Augmented Greedy algorithm

In this section, we present our algorithm that we refer 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 $f^{\mathcal{C}}(\Delta)$. Note that 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 subset of customer types $\mathcal{C}$, a subset of products $\mathcal{P}$ and a scalar k, and tries to find an assortment of size at most k that maximizes the function $f^{\mathcal{C}}$ over the set of products $\mathcal{P}$. In particular, Greedy picks iteratively a product that provides the highest increase in the objective value until reaching the cardinality k. We use $\mathsf{Greedy}(\mathcal{C}, \mathcal{P}, k)$ to denote the solution returned by Greedy. Below, we state the details of Greedy for completeness.

Greedy

1: Input: customer types $\mathcal{C} \subseteq \mathcal{M}$, products $\mathcal{P} \subseteq \mathcal{N}$, cardinality k
2: $\Delta \leftarrow \emptyset$
3: **while** $|\Delta| < k$ and $\mathcal{P} \setminus \Delta \neq \emptyset$ **do**
4: Add to Δ a product $i \in \mathcal{P} \setminus \Delta$ that maximizes $f^{\mathcal{C}}(\Delta \cup i)$
5: **end while**
6: **return** $\mathsf{Greedy}(\mathcal{C}, \mathcal{P}, k) = \Delta$

We are ready to 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 use the notation $[n] = \{1, \ldots, n\}$. We consider the set of products $\mathcal{N}$ in descending order of revenues. For each $i \in [n]$, let $\mathcal{V}_i$ denote the subset of products with revenues greater than r_i and let $\mathcal{W}_i$ denote the subset of products with revenues between r_{i+1} and $r_{i+1}/2$, with the convention that $\mathcal{W}_n = \emptyset$. For each $i \in [n]$, let $\Delta_i = \mathsf{Greedy}(\mathcal{C}, \mathcal{V}_i, k)$ be the assortment of

[1]We say that a set function f 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 $f(A \cup \{i\}) - f(A) \geq f(B \cup \{i\}) - f(B)$.

size at most k returned by Greedy with a set of products $\mathcal{V}_i$. Then, consider the set $\mathcal{W}_i$, round up the revenues of products in $\mathcal{W}_i$ to r_{i+1} and let $\widehat{\mathcal{W}_i}$ be the resulting set of products. We run Greedy on $\widehat{\mathcal{W}_i}$ as well and let $\Pi_i = \mathsf{Greedy}(\mathcal{C}, \widehat{\mathcal{W}_i}, k)$ be the resulting assortment which has size at most k. Augmented Greedy returns the assortment that maximizes $f^{\mathcal{C}}$ over all the $2n$ candidates Δ_i and Π_i for $i \in [n]$. We summarize our algorithm below.

Augmented Greedy

Input: customer types $\mathcal{C} \subseteq \mathcal{M}$, cardinality k
for $i = 1, 2, \ldots, n$ **do**
  Let $\mathcal{V}_i = \{1, 2, \ldots, i\}$
  $\Delta_i = \mathsf{Greedy}(\mathcal{C}, \mathcal{V}_i, k)$
  Let $\mathcal{W}_i = \{\ell \in \mathcal{N} \mid r_{i+1} \geq r_\ell \geq r_{i+1}/2\}$
  Round up the revenues of products in $\mathcal{W}_i$ to r_{i+1} and let $\widehat{\mathcal{W}_i}$ be the resulting set of products
  $\Pi_i = \mathsf{Greedy}(\mathcal{C}, \widehat{\mathcal{W}_i}, k)$
end for
return $\mathsf{AugGreedy}(\mathcal{C}, k) = \mathsf{argmax}_{i \in \mathcal{N}} \{f^{\mathcal{C}}(\Delta_i), f^{\mathcal{C}}(\Pi_i)\}$

4.2 Lower bound on the performance of Augmented Greedy

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 $\mathcal{C}$. In order to present formally our structural result, let us introduce the following definitions and notations. 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 set $\Delta = \mathsf{AugGreedy}(\mathcal{C}, k)$ 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. (Structural result). *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 greater than $|\mathcal{P} \cap S^*|$ and let $\Delta = \mathsf{AugGreedy}(\mathcal{C}, k)$. Then,*

$$f^{\mathcal{C}}(\Delta) \geq \frac{1}{4}(1 - \frac{1}{e}) \cdot f^{\mathcal{C}_{\mathcal{P}}}(\mathcal{P} \cap S^*).$$

In order to prove Theorem 4.2, we first present two lemmas. The first lemma shows that for each $j \in \mathcal{M}$, f_j is submodular if the revenues r_i of all products $i \in \mathcal{N}$ are equal.

Lemma 4.3. *Suppose that $r_i = r_\ell \ \forall i, \ell \in \mathcal{N}$. Then, for each $j \in \mathcal{M}$, f_j is submodular .*

Proof. Suppose that all products in $\mathcal{N}$ have revenues equal to r. Then, it is optimal to offer all available products under an unconstrained MNL model, i.e., for $S \subseteq \mathcal{N}$,

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

It follows, for $j \in \mathcal{M}$, $S \subseteq \mathcal{N}$ and $\ell \in \mathcal{N} \setminus S$,

$$f_j(S \cup \ell) - f_j(S) = r \cdot \frac{\sum_{i \in S} v_{ij} + v_{\ell j}}{\sum_{i \in S} v_{ij} + v_{\ell j} + 1} - r \cdot \frac{\sum_{i \in S} v_{ij}}{\sum_{i \in S} v_{ij} + 1} = \frac{r \cdot v_{\ell j}}{(\sum_{i \in S} v_{ij} + v_{\ell j} + 1)(\sum_{i \in S} v_{ij} + 1)}.$$

The above term is decreasing in S, therefore f_j is submodular. $\square$

As a consequence of Lemma 4.3, we get by linearity that $f = \sum_{j \in \mathcal{M}} \theta_j f_j$ is submodular in the case where all products in $\mathcal{N}$ have the same revenue. Moreover, since f is always monotone and $f(\emptyset) = 0$, it follows from Nemhauser et al. (1978) that the greedy algorithm gives $(1 - 1/e)$-approximation to CAP in this special case of equal revenues. We would like to note that this is the best approximation we could aim for in this case since Theorem 2.1 shows that even when all the products revenues are equal, there is no polynomial time algorithm that gives an approximation to CAP better than $(1 - 1/e)$, i.e., $(1 - 1/e)$ is a tight approximation bound.

To present the second lemma, let us introduce the following notation. Consider a subset of products $\mathcal{P} \subseteq \mathcal{N}$. Let p be the cheapest product in $\mathcal{N}$ that has a revenue greater than twice the revenue of the cheapest product in $\mathcal{P} \cap S^*$, i.e.,

$$p = \mathsf{argmin}_{i \in \mathcal{N}} \left\{ r_i \ : r_i \geq 2 \cdot \min_{\ell \in \mathcal{P} \cap S^*} r_\ell \right\}. \tag{4.1}$$

Consider Augmented Greedy with inputs $\mathcal{C}$ and k. Consider the p-th iteration of that algorithm, i.e., the iteration of the *for* loop corresponding to product p, and let $\mathcal{V}_p$, $\mathcal{W}_p$, Δ_p and Π_p be the sets of products as defined in this p-th iteration. Define,

$$\mathcal{Z} = \Delta_p \cup (\mathcal{P} \cap S^* \cap \mathcal{V}_p).$$

Note that Δ_p is obtained by running Greedy on $\mathcal{V}_p$ whereas $\mathcal{P} \cap S^* \cap \mathcal{V}_p$ is the set of optimal products in $\mathcal{P} \cap \mathcal{V}_p$. Lastly, let $\mathcal{C}_\mathcal{P}$ be the set of complete customer types with respect to $\mathcal{P}$. We split the customer types $\mathcal{C}_\mathcal{P}$ into two subsets as follows,

$$\mathcal{C}_1 = \{ j \in \mathcal{C}_\mathcal{P} \mid f_j(\Delta_p) \geq f_j(\mathcal{P} \cap S^*) \},$$

$$\mathcal{C}_2 = \mathcal{C}_\mathcal{P} \setminus \mathcal{C}_1.$$

In many parts of our proofs, we will use that f_j for $j \in \mathcal{M}$ is increasing i.e., $f_j(A) \leq f_j(B)$ for any $A \subseteq B \subseteq \mathcal{N}$ and f_j is subadditive in the sense that $f_j(A \cup B) \leq f_j(A) + f_j(B)$ for any

$A, B \subseteq \mathcal{N}$. These two properties are straightforward to show. For completeness, we provide their proofs in Appendix B. Our second lemma shows that f_j for $j \in \mathcal{C}_2$ is submodular on $\mathcal{Z}$. Recall that we say that a set function f 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 $f(A \cup \{i\}) - f(A) \geq f(B \cup \{i\}) - f(B)$.

Lemma 4.4. *For each $j \in \mathcal{C}_2$, f_j is submodular on $\mathcal{Z}$.*

Proof. Consider $j \in \mathcal{C}_2$. First, let us show that

$$r_p \geq f_j(\mathcal{Z}).$$

Let q be the cheapest product in $\mathcal{P} \cap S^*$, i.e., $r_q = \min_{\ell \in \mathcal{P} \cap S^*} r_\ell$. Since $j \in \mathcal{C}_2$, we have by definition $j \in \mathcal{C}_{\mathcal{P}}$, i.e., the customer type j is complete with respect to $\mathcal{P}$ and therefore $\mathcal{P} \cap S^* \subseteq S_j^*$. This implies that $q \in S_j^*$. As a classical property of assortment optimization under MNL, we know that a product is in the optimal assortment if and only if its revenue is greater than the optimal expected revenue of the MNL problem. For completeness, we prove this property in Appendix C. This property implies that,

$$r_q \geq \frac{\sum_{i \in S_j^*} r_i v_{ij}}{1 + \sum_{i \in S_j^*} v_{ij}} = f_j(S^*).$$

We know by definition that $r_p \geq 2 r_q$ and by monotonicity of f_j, we have $f_j(S^*) \geq f_j(\mathcal{P} \cap S^*)$. Therefore,

$$r_p \geq 2 f_j(\mathcal{P} \cap S^*).$$

On the other hand, we have

$$f_j(\mathcal{Z}) = f_j(\Delta_p \cup (\mathcal{P} \cap S^* \cap \mathcal{V}_p)) \leq f_j(\Delta_p) + f_j(\mathcal{P} \cap S^* \cap \mathcal{V}_p) \leq f_j(\Delta_p) + f_j(\mathcal{P} \cap S^*) < 2 f_j(\mathcal{P} \cap S^*),$$

where the first inequality follows from the subadditivity of f_j, the second one follows from the monotonicity of f_j and the last one follows from the definition of $\mathcal{C}_2$. Hence, we conclude that $r_p \geq f_j(\mathcal{Z})$. We have $\mathcal{V}_p = \{1, 2, \ldots, p\}$, thus for all $i \in \mathcal{V}_p$, $r_i \geq r_p$. Therefore, for all $i \in \mathcal{V}_p$,

$$r_i \geq f_j(\mathcal{Z}).$$

Consider a subset $S \subseteq \mathcal{Z}$. We know that $\Delta_p \subseteq \mathcal{V}_p$ and $(\mathcal{P} \cap S^* \cap \mathcal{V}_p) \subseteq \mathcal{V}_p$. Hence, $S \subseteq \mathcal{V}_p$ and therefore, for all $i \in S$,

$$r_i \geq f_j(\mathcal{Z}) \geq f_j(S),$$

where the last inequality holds by monotonicity. This is saying that the revenues of all products in S are greater than $f_j(S)$ which is the optimal expected revenue under MNL model for customer type j where the universe of products is S. Therefore, the classical MNL property that we present in Appendix C, implies that it is optimal to offer all the products in S, i.e.,

$$f_j(S) = \frac{\sum_{i \in S} r_i v_{ij}}{1 + \sum_{i \in S} v_{ij}}, \qquad \forall S \subseteq \mathcal{Z}.$$

Therefore for all $S \subseteq \mathcal{Z}$ and $\ell \in \mathcal{Z} \setminus S$,

$$
\begin{aligned}
f_j(S \cup \ell) - f_j(S) &= \frac{r_\ell v_{\ell j} + \sum_{i \in S} r_i v_{ij}}{1 + v_{\ell j} + \sum_{i \in S} v_{ij}} - \frac{\sum_{i \in S} r_i v_{ij}}{1 + \sum_{i \in S} v_{ij}} \\
&= \frac{r_\ell v_{\ell j}}{1 + v_{\ell j} + \sum_{i \in S} v_{ij}} - \frac{v_{\ell j} \sum_{i \in S} r_i v_{ij}}{(1 + v_{\ell j} + \sum_{i \in S} v_{ij})(1 + \sum_{i \in S} v_{ij})} \\
&= \frac{v_{\ell j}}{1 + v_{\ell j} + \sum_{i \in S} v_{ij}} \cdot (r_\ell - f_j(S)).
\end{aligned}
$$

The term $\frac{v_{\ell j}}{1+v_{\ell j}+\sum_{i \in S} v_{ij}}$ is decreasing in S and since f_j is an increasing function, it follows that $r_\ell - f_j(S)$ is decreasing in S. Note that $r_\ell - f_j(S) \geq 0$ because the left hand side $f_j(S \cup \ell) - f_j(S)$ is non-negative by monotonicity. This concludes that $f_j(S \cup \ell) - f_j(S)$ is decreasing in S and therefore f_j is submodular on $\mathcal{Z}$. □

Building on Lemma 4.3 and Lemma 4.4, we now present the proof of Theorem 4.2.

Proof of Theorem 4.2 Let p be the product defined in (4.1). Recall that Δ_p is the assortment returned by Greedy with inputs customer types $\mathcal{C}$, products $\mathcal{V}_p$ and cardinality k, i.e., $\Delta_p = \mathsf{Greedy}(\mathcal{C}, \mathcal{V}_p, k)$. Let $\{u_1, u_2, \ldots, u_k\}$ be the products of Δ_p where the indices of the products are in the order they were picked in Greedy. In particular, product u_1 is the first one picked by Greedy and u_k is the last one. Let us define for all $i = 0, 1, \ldots, k$,

$$
U_i = \{u_1, u_2, \ldots, u_i\},
$$

with the convention that $U_0 = \emptyset$. Note that $U_k = \Delta_p$. Let $\{q_1, q_2, \ldots, q_\tau\}$ be the products of $\mathcal{P} \cap S^* \cap \mathcal{V}_p$ where $\tau = |\mathcal{P} \cap S^* \cap \mathcal{V}_p|$. We have $\tau \leq k$ because $|\mathcal{P} \cap S^*| \leq k$ by the assumption of our theorem. Let us define for all $t = 1, \ldots, \tau$,

$$
Q_t = \{q_1, q_2, \ldots, q_t\}.
$$

In particular, we have $Q_\tau = \mathcal{P} \cap S^* \cap \mathcal{V}_p$. Consider $j \in \mathcal{C}_2$ and fix $i \in \{0, 1, \ldots, k-1\}$. We know from Lemma 4.4 that f_j is submodular on $\mathcal{Z} = \Delta_p \cup (\mathcal{P} \cap S^* \cap \mathcal{V}_p)$. Moreover, U_k and Q_τ are subsets of $\mathcal{Z}$. Therefore,

$$
\begin{aligned}
f_j(\mathcal{P} \cap S^* \cap \mathcal{V}_p) = f_j(q_1 \cup \ldots \cup q_\tau) &\leq f_j(q_1 \cup \ldots \cup q_\tau \cup U_i) \\
&= f_j(U_i) + \sum_{t=1}^{\tau} f_j(U_i \cup q_1 \cup \ldots \cup q_t) - f_j(U_i \cup q_1 \cup \ldots \cup q_{t-1}) \\
&\leq f_j(U_i) + \sum_{t=1}^{\tau} f_j(U_i \cup q_t) - f_j(U_i),
\end{aligned}
$$

where the first inequality holds by monotonicity of f_j and the second one from its submodularity on $\mathcal{Z}$. Therefore, by summing up the above inequality over all customer types $\mathcal{C}_2$ weighted by their arrival probabilities, we get

$$
\begin{aligned}
f^{\mathcal{C}_2}(\mathcal{P}\cap S^*\cap\mathcal{V}_p) &\le f^{\mathcal{C}_2}(U_i)+\sum_{t=1}^{\tau}\left(f^{\mathcal{C}_2}(U_i\cup q_t)-f^{\mathcal{C}_2}(U_i)\right)\\
&\le f^{\mathcal{C}}(U_i)+\sum_{t=1}^{\tau}\left(f^{\mathcal{C}}(U_i\cup q_t)-f^{\mathcal{C}}(U_i)\right)\\
&\le f^{\mathcal{C}}(U_i)+\sum_{t=1}^{\tau}\left(f^{\mathcal{C}}(U_{i+1})-f^{\mathcal{C}}(U_i)\right)\\
&\le f^{\mathcal{C}}(U_i)+k\cdot\left(f^{\mathcal{C}}(U_{i+1})-f^{\mathcal{C}}(U_i)\right),
\end{aligned}
$$

where the second inequality holds by adding to the right hand side the non-negative term $f^{\mathcal{C}_1}(U_i)$ and the term $\sum_{t=1}^{\tau}\left(f^{\mathcal{C}_1}(U_i\cup q_t)-f^{\mathcal{C}_1}(U_i)\right)$ which is also non-negative by monotonicity. The third inequality follows from the construction of Greedy as $f^{\mathcal{C}}(U_{i+1})\ge f^{\mathcal{C}}(U_i\cup\{\ell\})$ for any $\ell\in\mathcal{V}_p$. In particular, $f^{\mathcal{C}}(U_{i+1})\ge f^{\mathcal{C}}(U_i\cup q_t)$. The last inequality holds simply because $\tau\le k$. Therefore, for all $i\in\{0,1,\dots,k-1\}$, we can rewrite the above inequality as follows

$$
f^{\mathcal{C}}(U_{i+1})-f^{\mathcal{C}_2}(\mathcal{P}\cap S^*\cap\mathcal{V}_p)\ge(1-\frac{1}{k})\cdot\left(f^{\mathcal{C}}(U_i)-f^{\mathcal{C}_2}(\mathcal{P}\cap S^*\cap\mathcal{V}_p)\right).
$$

We have $f(U_0)=f(\emptyset)=0$. Hence, by using the above inequality recursively, we get

$$
f^{\mathcal{C}}(U_k)\ge\left(1-\left(1-\frac{1}{k}\right)^k\right)\cdot f^{\mathcal{C}_2}(\mathcal{P}\cap S^*\cap\mathcal{V}_p)\ge(1-\frac{1}{e})\cdot f^{\mathcal{C}_2}(\mathcal{P}\cap S^*\cap\mathcal{V}_p).
$$

On the other hand, we have

$$
f^{\mathcal{C}}(\Delta_p)\ge f^{\mathcal{C}_1}(\Delta_p)\ge f^{\mathcal{C}_1}(\mathcal{P}\cap S^*)\ge f^{\mathcal{C}_1}(\mathcal{P}\cap S^*\cap\mathcal{V}_p),
$$

where the first inequality holds because $\mathcal{C}_1\subseteq\mathcal{C}$, the second one follows from the definition of $\mathcal{C}_1$ and the last one holds by monotonicity. Recall that $\Delta_p=U_k$. Therefore, from the last two inequalities we deduce that,

$$
\begin{aligned}
f^{\mathcal{C}}(\Delta_p) &\ge \frac{1}{2}\left((1-\frac{1}{e})\cdot f^{\mathcal{C}_1}(\mathcal{P}\cap S^*\cap\mathcal{V}_p)+f^{\mathcal{C}_2}(\mathcal{P}\cap S^*\cap\mathcal{V}_p)\right)\\
&\ge\frac{1}{2}(1-\frac{1}{e})\cdot f^{\mathcal{C}_{\mathcal{P}}}(\mathcal{P}\cap S^*\cap\mathcal{V}_p).
\end{aligned}
\tag{4.2}
$$

Now, consider the set $\mathcal{W}_p$. Recall that the revenues of the products in $\mathcal{W}_p$ are between r_{p+1} and $r_{p+1}/2$. These revenues were rounded up to r_{p+1} to form the set $\widehat{\mathcal{W}}_p$, which results in losing at most a multiplicative factor of 2. We know from Lemma 4.3, that f_j is submodular when all revenues are equal. Hence, $f^{\mathcal{C}}$ is submodular on $\widehat{\mathcal{W}}_p$ and monotone. Therefore, by the classical result of Nemhauser et al. (1978) Greedy gives $(1-\frac{1}{e})$-approximation to the problem of maximizing $f^{\mathcal{C}}$ over the set $\widehat{\mathcal{W}}_p$ subject to cardinality k. Thus, reverting back to the original products revenues, Greedy gives $\frac{1}{2}(1-\frac{1}{e})$-approximation to the problem of maximizing $f^{\mathcal{C}}$ over the set $\mathcal{W}_p$ subject to cardinality k. Recall that $\Pi_p=\mathsf{Greedy}(\mathcal{C},\widehat{\mathcal{W}}_p,k)$. In particular, for any subset $S\subseteq\mathcal{W}_p$ with size at most k,

$$f^{\mathcal{C}}(\Pi_p) \geq \frac{1}{2}(1-\frac{1}{e}) \cdot f^{\mathcal{C}}(S).$$

We have $\mathcal{P} \cap S^* \cap \mathcal{W}_p$ is a subset of $\mathcal{W}_p$ of size at most k because $k \geq |\mathcal{P} \cap S^*|$. Therefore,

$$f^{\mathcal{C}}(\Pi_p) \geq \frac{1}{2}(1-\frac{1}{e}) \cdot f^{\mathcal{C}}(\mathcal{P} \cap S^* \cap \mathcal{W}_p) \geq \frac{1}{2}(1-\frac{1}{e}) \cdot f^{\mathcal{C}_{\mathcal{P}}}(\mathcal{P} \cap S^* \cap \mathcal{W}_p), \tag{4.3}$$

where the last inequality holds because $\mathcal{C}_{\mathcal{P}} \subseteq \mathcal{C}$. Finally, let us show that $\mathcal{P} \cap S^* \subseteq \mathcal{V}_p \cup \mathcal{W}_p$. In fact, by definition of product p, we have $r_p \geq 2 \cdot \min_{\ell \in \mathcal{P} \cap S^*} r_\ell$ and $r_{p+1} < 2 \cdot \min_{\ell \in \mathcal{P} \cap S^*} r_\ell$. Therefore, $r_\ell > r_{p+1}/2$ for any $\ell \in \mathcal{P} \cap S^*$. Since, $\mathcal{V}_p \cup \mathcal{W}_p = \{i \in \mathcal{N} \mid r_i \geq r_{p+1}/2\}$, we get that $\mathcal{P} \cap S^* \subseteq \mathcal{V}_p \cup \mathcal{W}_p$. Therefore, we have by subadditivity,

$$f^{\mathcal{C}_{\mathcal{P}}}(\mathcal{P} \cap S^* \cap \mathcal{V}_p) + f^{\mathcal{C}_{\mathcal{P}}}(\mathcal{P} \cap S^* \cap \mathcal{W}_p) \geq f^{\mathcal{C}_{\mathcal{P}}}(\mathcal{P} \cap S^*). \tag{4.4}$$

From (4.2), (4.3) and (4.4) we deduce that,

$$\max\{f^{\mathcal{C}}(\Delta_p), f^{\mathcal{C}}(\Pi_p)\} \geq \frac{1}{4}(1-\frac{1}{e}) \cdot f^{\mathcal{C}_{\mathcal{P}}}(\mathcal{P} \cap S^*).$$

which concludes the proof since the solution of Augmented Greedy is $\mathsf{argmax}_{i \in \mathcal{N}} \{f^{\mathcal{C}}(\Delta_i), f^{\mathcal{C}}(\Pi_i)\}$. □

5 Approximation algorithm for customer types with equal arrival probabilities

In this section, we consider the case where the arrival probabilities $\{\theta_j : j \in \mathcal{M}\}$ are equal for all customer types, i.e., $\theta_j = \frac{1}{m}, \forall j \in \mathcal{M}$. We show that Augmented Greedy with inputs: customer types $\mathcal{M}$ and cardinality K, gives an $\Omega(1/\log m)$-approximation to CAP in this case. We will build on this result in Section 6 to show our main result that gives $\Omega(1/\log m)$-approximation algorithm to CAP in the general case.

Theorem 5.1. *Suppose all customer types in $\mathcal{M}$ have equal arrival probabilities. Let z_{CAP} be the optimal objective value of* CAP *and let $\Delta = \mathsf{AugGreedy}(\mathcal{M}, K)$ be the assortment returned by Augmented Greedy with inputs $\mathcal{M}$ and K. Then,*

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

Proof. 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. Suppose without loss of generality that customer type 1 has the largest $\{f_j(S^*) : j \in \mathcal{M}\}$, i.e., $f_1(S^*) \geq f_j(S^*)$ for all $j \in \mathcal{M}$. Let $\log x$ be the logarithm of x in base 2. We partition the products in S^* based on their revenues as follows,

$$S^* = P_H \cup P_1 \cup \ldots \cup P_{\lceil \log m \rceil} \cup P_L,$$

where

$$P_H = \{i \in S^* \mid r_i \geq f_1(S^*)\}, \qquad P_L = \left\{ i \in S^* \mid r_i < \frac{f_1(S^*)}{2^{\lceil \log m \rceil}} \right\},$$

and for all $\ell = 1, 2, \ldots, \lceil \log m \rceil$,

$$P_\ell = \left\{ i \in S^* \;\middle|\; \frac{f_1(S^*)}{2^\ell} \le r_i < \frac{f_1(S^*)}{2^{\ell-1}} \right\}.$$

By subadditivity of f, we get

$$f(P_H) + f(P_L) + \sum_{\ell=1}^{\lceil \log m \rceil} f(P_\ell) \ge f(S^*) = z_{\mathsf{CAP}}. \tag{5.1}$$

The classical property of assortment optimization under MNL says that a product is in the optimal assortment if and only if its revenue is greater than the optimal expected revenue of the assortment optimization under MNL (see Lemma C.1 in Appendix C). Recall that $f_j(S^*)$ is the optimal expected revenue under MNL for customer type j with universe of products S^* and the optimal solution for this problem is S_j^*. Therefore, by the preceding MNL property, we have for any $j \in \mathcal{M}$,

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

Since customer type 1 has the highest $f_j(S^*)$ among all customer types $\mathcal{M}$, for any $i \in S_1^*$ and any $j \in \mathcal{M}$, we get $r_i \ge f_1(S^*) \ge f_j(S^*)$. Therefore, $S_1^* \subseteq S_j^*$ for all $j \in \mathcal{M}$. Thus,

$$S_1^* \cap S_j^* = S_1^* = S_1^* \cap S^*, \qquad \forall j \in \mathcal{M},$$

which implies that all customer types in $\mathcal{M}$ are complete with respect to S_1^*. Note that $P_H = S_1^*$. Let Δ be the assortment returned by Augmented Greedy with inputs $\mathcal{M}$ and K, i.e., $\Delta = \mathsf{AugGreedy}(\mathcal{M}, K)$. The customer types in $\mathcal{M}$ are complete with respect to P_H and $K \ge |S^*| \ge |P_H \cap S^*|$. Therefore, from Theorem 4.2, we have

$$f(\Delta) \ge \frac{1}{4}(1 - \frac{1}{e}) \cdot f(P_H). \tag{5.2}$$

On the other hand, for any product $i \in P_L$, $r_i < \frac{f_1(S^*)}{2^{\lceil \log m \rceil}} \le \frac{f_1(S^*)}{m}$. Hence, for all $S \subseteq P_L$ and $j \in \mathcal{M}$, we get

$$\frac{\sum_{i \in S} r_i v_{ij}}{1 + \sum_{i \in S} v_{ij}} \le \frac{f_1(S^*)}{m} \cdot \frac{\sum_{i \in S} v_{ij}}{1 + \sum_{i \in S} v_{ij}} \le \frac{f_1(S^*)}{m},$$

which implies that $f_j(P_L) \le f_1(S^*)/m$. Recall that $\theta_j = \frac{1}{m}$ for all $j \in \mathcal{M}$. Therefore,

$$f(P_L) = \frac{1}{m} \cdot \sum_{j \in \mathcal{M}} f_j(P_L) \le \frac{1}{m} \cdot \sum_{j \in \mathcal{M}} \frac{f_1(S^*)}{m} = \frac{f_1(S^*)}{m}.$$

We have

$$f(P_H) = \frac{1}{m} \sum_{j \in \mathcal{M}} f_j(P_H) \ge \frac{f_1(P_H)}{m} = \frac{f_1(S^*)}{m},$$

where the last equality holds because $P_H = S_1^*$ and $f_1(S_1^*) = f_1(S^*)$ since S_1^* is the optimal MNL assortment to offer to customer type 1 among the universe of products S^*. Therefore,

$f(P_L) \leq f(P_H)$. Thus, we get from (5.1),

$$2f(P_H) + \sum_{\ell=1}^{\lceil \log m \rceil} f(P_\ell) \geq z_{\mathsf{CAP}}. \tag{5.3}$$

Finally, fix $\ell \in \{1, \ldots, \lceil \log m \rceil\}$. We will show that $f(\Delta) \geq \frac{1}{2}(1 - \frac{1}{e}) \cdot f(P_\ell)$. Let $p \in \mathcal{N}$ be the cheapest product that has a revenue greater than $f_1(S^*)/2^{\ell-1}$, i.e.,

$$p = \mathsf{argmin}_{i \in \mathcal{N}} \left\{ r_i \ : r_i \geq \frac{f_1(S^*)}{2^{\ell-1}} \right\}.$$

Product p is well defined because at least the products in S_1^* have revenues greater than $f_1(S^*)$ which is greater than $\frac{f_1(S^*)}{2^{\ell-1}}$ for any $\ell \in \{1, \ldots, \lceil \log m \rceil\}$. Recall from the description of Augmented Greedy that

$$\mathcal{W}_p = \{\ell \in \mathcal{N} \mid r_{p+1} \geq r_\ell \geq r_{p+1}/2\}.$$

We will show that $P_\ell \subseteq \mathcal{W}_p$. In fact, let $i \in P_\ell$. We have

$$\frac{f_1(S^*)}{2^\ell} \leq r_i < \frac{f_1(S^*)}{2^{\ell-1}}.$$

By definition of product p, we have that product $p+1$ is the most expensive product with revenue less than $\frac{f_1(S^*)}{2^{\ell-1}}$. Since $r_i < \frac{f_1(S^*)}{2^{\ell-1}}$, then $r_i \leq r_{p+1}$. On the other hand, we have

$$r_i \geq \frac{f_1(S^*)}{2^\ell} = \frac{1}{2} \cdot \frac{f_1(S^*)}{2^{\ell-1}} \geq \frac{1}{2} \cdot r_{p+1}.$$

Thus, $P_\ell \subseteq \mathcal{W}_p$. Recall that $\Delta = \mathsf{AugGreedy}(\mathcal{M}, K)$. We know by construction of Augmented Greedy that $f(\Delta) \geq f(\Pi_p)$ where $\Pi_p = \mathsf{Greedy}(\mathcal{M}, \widehat{\mathcal{W}_p}, K)$ and $\widehat{\mathcal{W}_p}$ is the resulting set of products after rounding up the revenues of products in $\mathcal{W}_p$ to r_{p+1}. By the classical result of Nemhauser et al. (1978), Greedy gives $(1 - \frac{1}{e})$-approximation to the problem of maximizing f over the set $\widehat{\mathcal{W}_p}$ subject to cardinality K. Thus, reverting back to the original products revenues, Greedy gives $\frac{1}{2}(1 - \frac{1}{e})$-approximation to the problem of maximizing f over the set $\mathcal{W}_p$ subject to cardinality K. In particular, for any subset $S \subseteq \mathcal{W}_p$ with size at most K, $f(\Pi_p) \geq \frac{1}{2}(1 - \frac{1}{e}) \cdot f(S)$. We have P_ℓ is a subset of $\mathcal{W}_p$ and has a size of at most K since $P_\ell \subseteq S^*$. Hence, $f(\Pi_p) \geq \frac{1}{2}(1 - \frac{1}{e}) \cdot f(P_\ell)$. Therefore, for any $\ell \in \{1, 2, \ldots, \lceil \log m \rceil\}$,

$$f(\Delta) \geq \frac{1}{2}(1 - \frac{1}{e}) \cdot f(P_\ell),$$

which implies that

$$f(\Delta) = \Omega(1/\log m) \cdot \sum_{\ell=1}^{\lceil \log m \rceil} f(P_\ell). \tag{5.4}$$

From (5.2), (5.3) and (5.4), we conclude that $f(\Delta) = \Omega(1/\log m) \cdot z_{\mathsf{CAP}}$. □

6 $\Omega(1/\log m)$-approximation algorithm to CAP in the general case

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.

6.1 Algorithm and main result

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 θ_j. 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 our assortment candidate that gives $\Omega(1/\log m)$-approximation to CAP. 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 given in Theorem 4.2. Below, we describe the steps of our algorithm.

Step 1. (Rounding of θ_j). 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 value of θ_j to $\hat{\theta}_j = \frac{\lfloor \theta_j m^{\ell+1} \rfloor}{m^{\ell+1}}$. Note that $\frac{\theta_j}{2} \leq \hat{\theta}_j \leq \theta_j$. We lose at most a factor 2 by this rounding and therefore we will focus on showing $\Omega(1/\log m)$-approximation for CAP with $\{\hat{\theta}_j : j \in \mathcal{M}\}$. For ease of notation, we simply use θ_j to denote the rounded values in the rest.

Step 2. (Tree construction for customer types). For $\ell \in \mathbb{Z}$, let,

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

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 sets 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 $i \in \{1, \ldots, L\}$, $G_i = \mathcal{M}(\ell_i)$ for some $\ell_i \in \mathbb{Z}$ and G_i is non-empty. We index the 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, the 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 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 $\{G_1, G_2, G_3, G_4\}$.

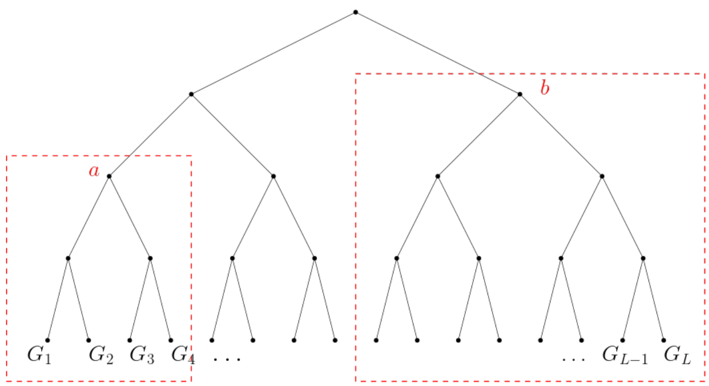

Figure 1: Tree of customer types $\mathcal{T}$

Step 3. (Augmented Greedy). Recall that K is the the bound on the number of products that we can offer in CAP. We use the notation $[K] = \{1, 2, \ldots, K\}$. For each subtree $T \in \mathcal{L}$ and each $k \in [K]$, we use Augmented Greedy to compute the following 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. (Synthesis). Finally, we solve the following maximization problem,

$$\begin{aligned}
\max \quad & \sum_{T \in \mathcal{L}} \sum_{k=1}^{K} R_{Tk} \cdot x_{Tk} \\
& \sum_{T \in \mathcal{L}} \sum_{k=1}^{K} k \cdot x_{Tk} \leq K, \\
& \sum_{k=1}^{K} (x_{Tk} + x_{T'k}) \leq 1, \qquad \forall T \in \mathcal{L}, \forall T' \subseteq T, T' \neq T \\
& x_{Tk} \in \{0, 1\}, \qquad \forall T \in \mathcal{L}, \forall k \in [K].
\end{aligned} \tag{6.1}$$

The notation $T' \subseteq T$ in the second constraint means that all the nodes of the subtree T' belong to the subtree T, or in other words the root of T' is a node in T. In problem (6.1), 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 is less than K. The second constraint ensures that the chosen subtrees are disjoint, moreover for each chosen subtree T there exists only one scalar k such that $x_{Tk} = 1$. Problem (6.1) can be solved in polynomial time using a simple tractable dynamic program.

Dynamic program for solving problem (6.1)**.** The state variable of the dynamic program is (T, k) for $T \in \mathcal{L}$ and $k \in [K]$. We compute the value functions $\{g(T, k) : T \in \mathcal{L}, k \in [K]\}$ as follows,

- If T is a leaf,
$$g(T, k) = R_{Tk}.$$
- Otherwise,
$$g(T, k) = \max\left\{R_{Tk}, \max_{k_1+k_2=k}\{g(T_1, k_1) + g(T_2, k_2)\}\right\},$$
where T_1 and T_2 are respectively the left and right subtrees of T.

The optimal objective value of problem (6.1) is given by $g(\mathcal{T}, K)$. Let x^*_{Tk} for $T \in \mathcal{L}, k \in [K]$ be the optimal solution of problem (6.1). Our candidate solution for CAP is the following:

$$\Delta = \left\{\bigcup_{T\in\mathcal{L},k\in[K]} \Delta_{Tk} \ : \ x^*_{Tk} = 1\right\}. \tag{6.2}$$

Theorem 6.1. *Let z_{CAP} be the optimal value of CAP and Δ be the assortment in* (6.2)*. Then,*

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

6.2 Proof of Theorem 6.1

In order to show Theorem 6.1, we give two preliminary lemmas, where we show structural properties that are useful in the proof of Theorem 6.1. The first lemma shows the existence of a subset of customers $\mathcal{X}$ such that the optimal revenue from the customers in $\mathcal{X}$ is a constant fraction of the optimal revenue of CAP and the customer types in $\mathcal{X}$ verify the following monotonicity property: for any $j, q \in \mathcal{X}$, if $j \in G_i$ and $q \in G_{i'}$ with $i < i'$, then $f_j(S^*) \geq f_q(S^*)$. In particular, we have the following lemma.

Lemma 6.2. *Let S^* be the optimal solution of CAP and z_{CAP} be its optimal objective value. There exists a subset of customer types $\mathcal{X} \subseteq \mathcal{M}$ such that:*

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

The proof of Lemma 6.2 is deferred to Appendix D. Before we present the second lemma, we introduce the following notation. Let S^* be the optimal solution of CAP and for each $j \in \mathcal{M}$, let

$S_j^* \subseteq S^*$ be the optimal subset to offer to customer 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,

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

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 $\mathcal{C} \subseteq \mathcal{M}$,

$$f^{\mathcal{C}}(A) \geq h^{\mathcal{C}}(A). \tag{6.3}$$

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 revenue R_{Tk} of the assortment given 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 6.3. *Consider a pair (A, T) where $A \subseteq S^*$ is a subset of the optimal products 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. Let $\mathcal{X}$ be the set of customer types defined in Lemma 6.2. Then, at least one of the following two statements is true 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., $A = A_1 \cup A_2$, $A_1 \cap A_2 = \emptyset$ such that,

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

The proof of Lemma 6.3 is deferred to Appendix E. Now we are ready to present the proof of Theorem 6.1. Let Δ be the assortment defined in (6.2). The goal is to show that Δ is a feasible assortment for CAP and $f(\Delta) = \Omega(1/\log m) \cdot z_{\mathsf{CAP}}$. We will follow the following steps in our proof: First, we will show that Δ is feasible for CAP. Second, we will show that $f(\Delta)$ is greater than the optimal objective value of problem (6.1). So, it is sufficient to show that the optimal objective value of problem (6.1) is $\Omega(1/\log m) \cdot z_{\mathsf{CAP}}$. To do that, building on Lemma 6.2 and Lemma 6.3, we construct a feasible solution for problem (6.1) and finally show that this solution has an objective value greater than $\Omega(1/\log m) \cdot z_{\mathsf{CAP}}$.

Feasibility. Let x^*_{Tk} for $T \in \mathcal{L}, k \in [K]$ be the optimal solution of problem (6.1) and Δ be the assortment defined in (6.2), i.e., $\Delta = \{\bigcup_{T \in \mathcal{L}, k \in [K]} \Delta_{Tk} \ : \ x^*_{Tk} = 1\}$. Recall $\Delta_{Tk} = \mathsf{AugGreedy}(\mathcal{C}_T, k)$

and 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 (6.1). Hence, Δ is a feasible solution for CAP.

Lower bound for $f(\Delta)$. We know by the definition of $\mathcal{L}$ that if $T, T' \in \mathcal{L}$ with $T \cap T' \neq \emptyset$, then either $T \subseteq T'$ or $T' \subseteq T$. Hence, the second constraint of (6.1) 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 $T \cap T' = \emptyset$. Therefore, the sets $\{\mathcal{C}_T \mid \sum_{k=1}^{K} x^*_{Tk} = 1\}$ 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 follows by monotonicity since $\Delta_{Tk} \subseteq \Delta$ for $x^*_{Tk} = 1$. Therefore, to prove Theorem 6.1, it is sufficient to show that the optimal objective value of problem (6.1) is $\Omega(1/\log m) \cdot z_{\mathsf{CAP}}$.

Feasible solution for problem (6.1). We construct a solution for problem (6.1) with an objective value of $\Omega(1/\log m) \cdot z_{\mathsf{CAP}}$. So, the optimal objective value of problem (6.1) is $\Omega(1/\log m) \cdot z_{\mathsf{CAP}}$. Our construction is as follows. We initialize all the variables of problem (6.1) at 0. Consider the set $\mathcal{X}$ given in Lemma 6.2. 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 6.3 with the pair $(S^*, \mathcal{T})$. We know that at least one of the two statements of Lemma 6.3 should be true. Suppose, the first statement is true, i.e., $R_{\mathcal{T}K} = \Omega(1/\log m) \cdot h^{\mathcal{X}}(S^*)$, therefore,

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

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

Otherwise, if the first statement of Lemma 6.3 for the pair $(S^*, \mathcal{T})$ is not true, then we know that the second statement should be true, i.e., there exists a partition of $S^* = A_1 \cup A_2$, $A_1 \cap A_2 = \emptyset$, such that

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

where T_1 is the left subtree of $\mathcal{T}$ and T_2 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

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

We repeat the same argument for the pairs (A_1, T_1) and (A_2, T_2). In general, each time we are at a pair (A, T), we let $\alpha_T = h^{\mathcal{X}\cap\mathcal{C}_T}(A)$. We check if the first statement of Lemma 6.3 is true for the pair (A, T) and if it is the case we do the following: we set $x_{Tk} = 1$ for $k = |A|$ and *truncate* all the descendant nodes to the root of T, i.e., we do not further explore T. We know from Lemma 6.3 that

$$R_{Tk} = \Omega(1/\log m) \cdot \alpha_T. \tag{6.4}$$

Otherwise, if the first statement is not true, then the second statement of the lemma is true and there exists a partition of $A = A_1 \cup A_2$ 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. We let $\alpha_{T_1} = h^{\mathcal{X}\cap\mathcal{C}_{T_1}}(A_1)$, $\alpha_{T_2} = h^{\mathcal{X}\cap\mathcal{C}_{T_2}}(A_2)$ and we have

$$\alpha_{T_1} + \alpha_{T_2} \geq (1 - \frac{1}{\log m}) \cdot \alpha_T. \tag{6.5}$$

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 $\mathcal{T}$, we know from the proof of Lemma 6.3 that the first statement should be true for this pair and therefore (6.4) is verified for T.

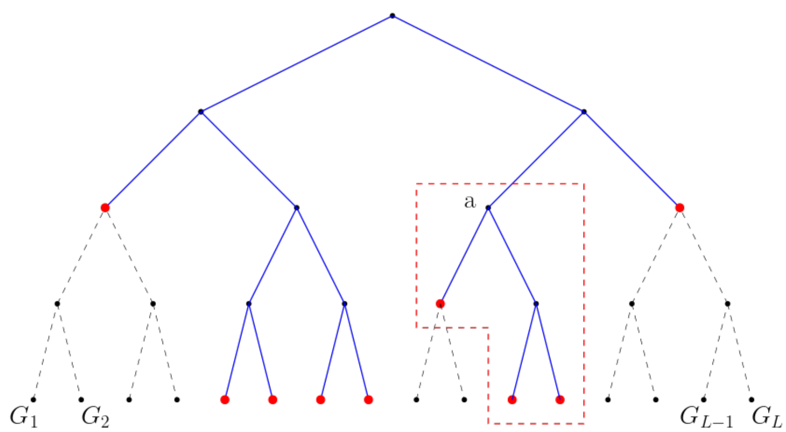

Figure 2: Truncated Tree of customer types.

Let x_{Tk} for $T \in \mathcal{L}, k \in [K]$ be the solution that we have constructed as above. Let us show that this solution is feasible for problem (6.1). 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 . Note that by construction, 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 (6.1). Moreover, if $x_{T_i k_i} = 1$ for some $i \in \mathcal{I}$, we know by our construction 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 $k \in [K]$. Furthermore, the subtrees T that correspond to the parent nodes of T_i are such that $X_{Tk} = 0$ for any $k \in [K]$. Thus, $\sum_{k=1}^{K}(x_{Tk} + x_{T'k}) \leq 1, \forall T \in \mathcal{L}, \forall T' \subseteq T, T' \neq T$, which implies that our solution is feasible for problem (6.1).

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 (6.1). 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. By definition the depth of a subtree is the distance between the 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 corresponds to the subtree that corresponds to node a. The subtree $T \in \mathcal{L}$ that corresponds to a in the original tree has four leaves. The subtree that corresponds to a in the truncated tree has 3 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

$$\sum_{T' \in \mathsf{leaves}(T)} \alpha_{T'} \geq \left(1 - \frac{1}{\log m}\right)^{\mathsf{depth}(T)} \alpha_T. \tag{6.6}$$

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 1 in the truncated tree. Hence, T has exactly two leaves and both are red nodes. Therefore, (6.6) follows directly from (6.5).
Induction. Suppose inequality (6.6) is true for the subtrees with depth d in the truncated tree. Consider a subtree $T \in \mathcal{L}$ that has a depth $d+1$ in the truncated tree. Let T_1 be its left subtree and T_2 its right subtree. From (6.5), 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, the induction hypothesis implies for $i \in \{1, 2\}$,

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

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. Let us apply now the inequality (6.6) 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,

$$\sum_{T' \in \mathsf{leaves}(\mathcal{T})} \alpha_{T'} \geq \left(1 - \frac{1}{\log m}\right)^{\log m} \alpha_{\mathcal{T}} = \Theta(e^{-1}) \cdot \alpha_{\mathcal{T}}, \tag{6.7}$$

where the last equality holds for sufficiently large m. Finally, each leaf in the truncated tree (these are red nodes in Figure 2) verifies (6.4), i.e., for all $i \in \mathcal{I}$, $R_{T_i k_i} = \Omega(1/\log m) \cdot \alpha_{T_i}$, which implies

$$\sum_{i\in\mathcal{I}} R_{T_i k_i} = \Omega(1/\log m) \sum_{T' \in \mathsf{leaves}(\mathcal{T})} \alpha_{T'}. \tag{6.8}$$

Moreover,

$$\alpha_{\mathcal{T}} = h^{\mathcal{X}}(S^*) = \sum_{j\in\mathcal{X}} \theta_j \frac{\sum_{i\in S^*\cap S_j^*} r_i v_{ij}}{1+\sum_{i\in S^*\cap S_j^*} v_{ij}} = f^{\mathcal{X}}(S^*) \geq z_{\mathsf{CAP}}/4, \tag{6.9}$$

where the last inequality follows from Lemma 6.2 . Therefore, from (6.7), (6.8) and (6.9), 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.

7 FPTAS for constant number of customer types

In this section, we develop an FPTAS for CAP in 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 $f(S) \geq (1-\delta) \cdot z_{\mathsf{CAP}}$. Let S^* be the optimal solution of CAP, and for $j \in \mathcal{M}$, let $S_j^* \subseteq S^*$ be the optimal assortment to offer to customer type j in CAP. Throughout this section, we use the notation

$$\mathrm{OPT}_j = f_j(S^*) \qquad \text{and} \qquad \mathrm{OPT} = f(S^*).$$

In particular, $\mathrm{OPT} = \sum_{j\in\mathcal{M}} \theta_j \cdot \mathrm{OPT}_j$. At a high level, our algorithm starts by guessing the value of OPT_j for each $j \in \mathcal{M}$. Assume t_j is our guess for OPT_j. Then, using a dynamic program, our algorithm tries to find a feasible assortment S for CAP such that $f_j(S) \approx t_j$ for each $j \in \mathcal{M}$. Our dynamic program is based on the observation that we have for all $j \in \mathcal{M}$,

$$\mathrm{OPT}_j = \sum_{i\in S^*} (r_i - \mathrm{OPT}_j)^+ v_{ij}. \tag{7.1}$$

Equation (7.1) follows from the definition of OPT_j and Lemma C.1. In fact, for all $j \in \mathcal{M}$,

$$\mathrm{OPT}_j = \frac{\sum_{i\in S_j^*} r_i v_i}{1+\sum_{i\in S_j^*} v_i},$$

i.e., $\mathrm{OPT}_j = \sum_{i\in S_j^*}(r_i - \mathrm{OPT}_j)v_{ij}$, which implies (7.1), since from Lemma C.1, we have

$$S_j^* = \{i \in S^* : r_i \geq \mathrm{OPT}_j\}.$$

Below, we present the details of our FPTAS.

Set of guesses. For each $j \in \mathcal{M}$, let $b_{j,\min}$ (resp. $b_{j,\max}$) denote a lower bound (resp. an upper bound) of $f_j(S)$ for any feasible assortment S of CAP. For example, we choose the upper bound to be $b_{j,\max} = \max_{i\in\mathcal{N}} r_i$ because

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

Whenever $f_j(S) > 0$, we can lower bound it by $b_{j,\min} = \min_{i\in\mathcal{N}} r_i \cdot \frac{v_{j,\min}}{1+v_{j,\min}}$ where $v_{j,\min} = \min_{i\in\mathcal{N}, v_{ij}\neq 0} v_{ij}$. Fixing $\epsilon > 0$, let us define the following geometric grid

$$\mathbf{\Gamma}_\epsilon = \Gamma_1 \times \Gamma_2 \times \ldots \times \Gamma_m,$$

where for each $j \in \mathcal{M}$,

$$\Gamma_j = 0 \cup \{b_{j,\min} \cdot (1+\epsilon)^\ell, \ell = 0, 1, \ldots, L_j\},$$

and $L_j = \log_{1+\epsilon}(b_{j,\max}/b_{j,\min})$. For $a \in [b_{j,\min}, b_{j,\max}]$, we define the round up operator $\lceil . \rceil$ that rounds up the argument a to the closest value in Γ_j. Similarly, we define the round down operator $\lfloor . \rfloor$ that rounds down the argument a to the closest value in Γ_j. Note that the set $\mathbf{\Gamma}_\epsilon$ contains all guesses that are within $(1+\epsilon)$ multiplicative factor from OPT_j for $j \in \mathcal{M}$.

Dynamic program. Fix a guess $\mathbf{t} = (t_1, \ldots, t_m) \in \mathbf{\Gamma}_\epsilon$. The state variable of the dynamic program is $(\ell, q_1, q_2, \ldots, q_m)$ for $\ell \in \{0, 1, \ldots, n\}$ and $(q_1, q_2, \ldots, q_m) \in \mathbf{\Gamma}_\epsilon$. For $i \in \{1, \ldots, n+1\}$, we compute the value function $V_i^{\mathbf{t}}(\ell, q_1, q_2, \ldots, q_m)$ as follows:

- For $i \in \mathcal{N}$ and $\ell = 0, 1, \ldots, i$,
$$V_i^{\mathbf{t}}(\ell, q_1, q_2, \ldots, q_m) = \max \begin{cases} V_{i+1}^{\mathbf{t}}(\ell+1, \lceil q_1 + v_{i1}(r_i - t_1)^+ \rceil, \ldots, \lceil q_m + v_{im}(r_i - t_m)^+ \rceil) \\ V_{i+1}^{\mathbf{t}}(\ell, q_1, \ldots, q_m) \end{cases}$$
- $V_{n+1}^{\mathbf{t}}(\ell, q_1, q_2, \ldots, q_m) = \begin{cases} -\infty & \text{if } \ell > K \text{ or } \exists j \in \mathcal{M} \text{ s.t. } q_j < t_j \\ 0 & \text{otherwise.} \end{cases}$

$$\tag{7.2}$$

In (7.2), if we did not have the round up operator $\lceil . \rceil$, then the dynamic program would find an assortment $S_{\mathbf{t}}$ such that $\sum_{i\in S_{\mathbf{t}}} v_{ij}(r_i - t_j)^+ \geq t_j$ for all $j \in \mathcal{M}$ and $|S_{\mathbf{t}}| \leq K$. Such an assortment exists if and only if $V_1^{\mathbf{t}}(0, 0, \ldots, 0) = 0$. The assortment $S_{\mathbf{t}}$ can be obtained by starting with the state $(0, 0, \ldots, 0)$ at decision epoch $i = 1$ and following the optimal state-action trajectory. The assortment $S_{\mathbf{t}}$ would include product i whenever we increase the state variable ℓ at decision epoch i under the optimal state-action trajectory. Similarly, with the round up operator $\lceil . \rceil$ in (7.2), the dynamic program finds an assortment $S_{\mathbf{t}} = (i_1, i_2, \ldots, i_k)$ such that $k \leq K$ and

$$\lceil\lceil\lceil\lceil\lceil v_{i_1 j}(r_{i_1} - t_j)^+ \rceil + v_{i_2 j}(r_{i_2} - t_j)^+ \rceil + \ldots \rceil + v_{i_k j}(r_{i_k} - t_j)^+ \rceil \geq t_j, \tag{7.3}$$

if such an assortment exists. Based on this observation, we have the following lemma.

Lemma 7.1. *For any guess $(t_1, \ldots, t_m) \in \mathbf{\Gamma}_\epsilon$, if there exists an assortment S of size at most K such that $f_j(S) \geq t_j$ for all $j \in \mathcal{M}$, then the dynamic program (7.2) finds an assortment $\tilde{S}$ of size at most K such that $f_j(\tilde{S}) \geq t_j/(1+\epsilon)^K$.*

Proof. Let $S = (i_1, i_2, \ldots, i_k)$ be an assortment of size k, $(k \leq K)$ such that $f_j(S) \geq t_j$ for all

$j \in \mathcal{M}$, i.e., $f_j(S) = \frac{\sum_{i\in S_j} r_i v_{ij}}{1+\sum_{i\in S_j} v_{ij}} \geq t_j$, where $S_j \subseteq S$ is the optimal subset to offer to customer type j out of S under MNL model. Hence,

$$\sum_{i\in S_j}(r_i - t_j)v_{ij} \geq t_j.$$

Therefore, $\sum_{i\in S_j}(r_i - t_j)^+ v_{ij} \geq t_j$, which implies that

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

Applying the round up operator to the left side above in a nested fashion, we get

$$\lceil\lceil\lceil\lceil\lceil v_{i_1 j}(r_{i_1} - t_j)^+\rceil + v_{i_2 j}(r_{i_2} - t_j)^+\rceil + \ldots\rceil + v_{i_k j}(r_{i_k} - t_j)^+\rceil \geq t_j.$$

This shows the existence of an assortment of size less than K that verifies property (7.3) which implies that $V_1^{\mathbf{t}}(0,0,\ldots,0) = 0$. Therefore, the dynamic program (7.2) finds a set $\tilde{S}$ that verifies (7.3). Therefore, by removing the round up operator and using the fact that $\lceil a\rceil \leq (1+\epsilon)\cdot a$, the inequality (7.3) applied to $\tilde{S}$, implies that for all $j \in \mathcal{M}$,

$$(1+\epsilon)^k \sum_{i\in\tilde{S}} v_{ij}(r_i - t_j)^+ \geq t_j,$$

where k is the size of $\tilde{S}$ and $k \leq K$. Denote $\hat{S}_j = \{i \in \tilde{S} : r_i \geq t_j\}$. Hence,

$$(1+\epsilon)^k \sum_{i\in\hat{S}_j} v_{ij}(r_i - t_j) \geq t_j,$$

and therefore,

$$(1+\epsilon)^k \frac{\sum_{i\in\hat{S}_j} r_i v_{ij}}{1+\sum_{i\in\hat{S}_j} v_{ij}} \geq t_j.$$

We have $\hat{S}_j \subseteq \tilde{S}$ for all $j \in \mathcal{M}$, thus,

$$f_j(\tilde{S}) \geq \frac{\sum_{i\in\hat{S}_j} r_i v_{ij}}{1+\sum_{i\in\hat{S}_j} v_{ij}}.$$

Finally, since $k \leq K$, we conclude that $(1+\epsilon)^K \cdot f_j(\tilde{S}) \geq t_j$. □

The following algorithm gives our FPTAS for CAP with constant m. The correctness proof of the algorithm and the running time analysis are presented in Theorem 7.2. In this algorithm, we set $b_{\max} = \max_{j\in\mathcal{M}} b_{j,\max}$ and $b_{\min} = \min_{j\in\mathcal{M}} b_{j,\min}$.

Theorem 7.2. *Fix $\delta > 0$. Running FPTAS with $\epsilon = (1-\delta)^{\frac{-1}{K+1}} - 1$, returns an assortment that gives $(1-\delta)$-approximation to* CAP*. The running time of the FPTAS algorithm is given by $O(n^2 K^{2m} \cdot \log^{2m}(b_{max}/b_{min})/\delta^{2m})$. This running time is polynomial in $n, K, \log(b_{max}/b_{min})$ and $1/\delta$ for constant m.*

FPTAS for constant number of customer types

1: Input: $\epsilon > 0$.
2: For all $\mathbf{t} = (t_1, \ldots, t_m) \in \mathbf{\Gamma}_\epsilon$:
3: Compute $V_1^{\mathbf{t}}(0, 0, \ldots, 0)$ using (7.2) and let $S_{\mathbf{t}}$ be the corresponding assortment
4: **return** $S_{\mathbf{t}}$ that maximizes $f(S_{\mathbf{t}})$ over $\mathbf{t} \in \mathbf{\Gamma}_\epsilon$.

Proof. Let S^* be the optimal assortment of CAP. Let $\mathbf{t} = (t_1, \ldots, t_m) \in \mathbf{\Gamma}_\epsilon$ be such that for all $j \in \mathcal{M}$,

$$t_j \leq f_j(S^*) \leq (1+\epsilon)t_j.$$

From Lemma 6.2, the dynamic program (7.2) gives a feasible assortment $\tilde{S}$ such that for all $j \in \mathcal{M}$,

$$f_j(\tilde{S}) \geq t_j/(1+\epsilon)^K \geq f_j(S^*)/(1+\epsilon)^{K+1} = (1-\delta)f_j(S^*).$$

Hence,

$$f(\tilde{S}) = \sum_{j \in \mathcal{M}} \theta_j f_j(\tilde{S}) \geq (1-\delta) \sum_{j \in \mathcal{M}} \theta_j f_j(S^*) = (1-\delta) f(S^*).$$

Let us analyze the running time of the algorithm. There are $O(\log_{1+\epsilon}^m (b_{max}/b_{min}))$ guesses in $\mathbf{\Gamma}_\epsilon$. The time to compute $V_i(\ell, q_1, q_2, \ldots, q_m)$ for all $i \in \mathcal{N}, \ell \in \{0, 1, \ldots, n+1\}$ and $(q_1, q_2, \ldots, q_m) \in \mathbf{\Gamma}_\epsilon$ is given by

$$O(n^2 \log_{1+\epsilon}^m (b_{max}/b_{min})).$$

Putting both terms together, we get a running time of

$$O(n^2 \log_{1+\epsilon}^{2m}(b_{max}/b_{min})) = O(n^2 \log^{2m}(b_{max}/b_{min})/\epsilon^{2m}) = O(n^2 K^{2m} \cdot \log^{2m}(b_{max}/b_{min})/\delta^{2m}),$$

where the last equality holds because $1 + \delta < (1-\delta)^{-1} = (1+\epsilon)^{K+1}$ implies that $\delta = O(\epsilon K)$ for small δ and ϵ, i.e., $\delta = o(1)$ and $\epsilon = o(1)$. $\square$

8 Computational study: Value of Customization

In this section, we use a dataset from Expedia, as well as randomly generated datasets, to demonstrate the value of customization.

8.1 Expedia data

We use a dataset provided by Expedia as a part of a Kaggle competition, see 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. The columns give information on the characteristics of the displayed hotels in a search query and the booking decision of the customer. We preprocessed the dataset using the same approach as in Gao et al. (2020). After removing missing and uninterpretable values, the resulting set contains 595,965 rows which represent 34,561 queries. In all the queries, the customer is looking for a single night booking. Each row 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 the search query. This corresponds to the set of hotels among which the customer needs to make a choice. The second column 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. The following eight columns show the attributes of the displayed hotel: Star rating, review score, hotel brand, location score, accessibility score, historical price, displayed price and promotion flag. The last four columns give information about the customer making the search query: Booking window of the customer, i.e., the number of days between the time of the search query and the booking date that the customer is looking for, 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 looking for a Saturday night booking. These four columns will be useful to define our customer types.

Experimental setup. We define 16 types of customers based on the following criteria: (i) whether the customer in the search query is looking for a Saturday night booking, (ii) whether the customer wants to make a booking for one adult or multiple ones, (iii) whether the customer wants to make a booking that includes children or only adults, (iv) whether the customer in the search query wants to make an *early* booking or a *late* one. We use the last four columns of the dataset to identify the type of the customer making the search query. To define the last criteria, we use the *Booking window* column and qualify a search query as late if the booking window is less than 7 days and early otherwise. These four binary criteria give us $2^4 = 16$ customer types.

In 83% of the search queries in the dataset, the customer did not make a booking. 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 such that 30% of the queries result in a no-purchase. A second dataset with 50% of no-purchase queries and a third one with 70% of no-purchase queries. For each dataset, we use the eight features of the displayed hotels mentioned earlier to estimate separately an MNL model for each customer type. We fit the MNL model for each customer type using maximum likelihood estimation. The preference weights are of the form: for a customer type j and a hotel i, $v_{ij} = \exp\left(\beta_{0j} + \sum_{\ell=1}^{8} \beta_{\ell j} x_{i\ell}\right)$ where $x_{i\ell}$ is the value of the ℓ-th attribute of hotel i and $\beta_0, \beta_{\ell j}$ for $\ell = 1, \ldots, 8$ are the estimated parameters for customer type j. In Table 1, we show an example of the estimated parameters of the MNL model for customer type 3 in the dataset that corresponds to 30% of no-purchase queries. Customer type 3 corresponds to customers making a query with the following criteria: the booking is *early* (i.e., query made at least seven days before the booking), it is not for a Saturday night, it has more than one adult and no children. The second column of Table 1 gives the total number of queries for customer type 3 and the rest of the columns show the parameters $\beta_{\ell 3}$ for $\ell = 0, 1, \ldots, 8$. For instance, the coefficient for the review score attribute is 0.15 which means that the review score affects positively the utility of the customer while the coefficient of the displayed price is -1.69 which implies that the utility of the customer is negatively correlated with the price.

Cust.	queries	β_0	stars	review	disp. price	promo	accessibility	hist. price	brand	location
3	1095	-2.45	0.49	0.15	-1.69	0.12	0.60	-0.05	0.01	-0.26

Table 1: Example of fitted MNL parameters for customer type 3 in the first dataset

To show the value of customization, we consider the following experiment. We define a mixture of multinomial logit models with 16 customer types. The arrival probability of a customer type is given by the relative frequency of this type, i.e., the number of search queries made by this type over the total number of search queries. The preference weights of each customer type are the ones we have estimated separately for each type. We compare the following two situations: (i) the platform identifies the customer type and makes an assortment offer accordingly, (ii) the platform offers the same assortment to all customers without paying attention to the customer type. Since, we do not have access to the universe of hotels that we could potentially show to the customer, we consider the set of hotels that are displayed to the customer in the search query as our universe, we refer to it by U. For each search query, we identify the type of the customer and the universe U. We solve an assortment optimization problem under the MNL model of this customer type with universe U. We use z_{MNL} to denote the optimal objective value of this problem. In particular, if the customer making the search query is of type j, then we have $z_{\mathsf{MNL}} = \max_{S \subseteq U} \frac{\sum_{i \in S} r_i v_{ij}}{1+\sum_{i \in S} v_{ij}}$. On the other hand, we solve an assortment optimization problem under the mixture of the 16 multinomial logit models with universe U. We use a standard mixed integer formulation with big M constraints to solve the MMNL problem to optimality (see for instance Méndez-Díaz et al. (2014)). Let S_{MMNL} be the optimal assortment of the latter problem. We use R_{MMNL} to denote the expected revenue from the customer type in the search query when offered assortment S_{MMNL}. In other words, if the customer making the search query is of type j, then we have

$$S_{\mathsf{MMNL}} = \arg\max_{S \subseteq U} \sum_{q \in \mathcal{M}} \theta_q \frac{\sum_{i \in S} r_i v_{iq}}{1 + \sum_{i \in S} v_{iq}} \quad \text{and} \quad R_{\mathsf{MMNL}} = \frac{\sum_{i \in S_{\mathsf{MMNL}}} r_i v_{ij}}{1 + \sum_{i \in S_{\mathsf{MMNL}}} v_{ij}}.$$

For each search query, we compute the ratio

$$\gamma_c = \frac{z_{\mathsf{MNL}} - R_{\mathsf{MMNL}}}{R_{\mathsf{MMNL}}}.$$

Here, γ_c can be interpreted as the percentage gain in revenue that is due to identifying the customer type and customizing the offered assortment as compared to showing the customer the optimal assortment of the mixture solution. We group the queries by customer types. For each customer type and each dataset, we report the mean, the 95% percentile and the maximum of γ_c over all queries made by that customer type. Our results are given in Tables 2, 3 and 4. 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 and the last three columns show the statistics of γ_c.

Results. We observe from Tables 2, 3 and 4 that there is a significant number of instances where the value of customization γ_c is high. We achieve an increase in revenue of around 1% on average over all the data due to customization. For many search queries, γ_c achieves values of 30% and higher. Moreover, the value of customization is higher for the customer types with small size which shows that that there is significant value in customizing the offered assortment for these customer types rather than offering them the same static assortment that is driven by large customer types. Note that γ_c is higher for the dataset where the fraction of the no-purchase queries is 30% as compared to the two other datasets. This is expected since there is no value of customization in the queries that ended up with a no-purchase. Our findings suggest that there is significant value in identifying the type of customers and personalizing the assortment of hotels according to the the customer type preferences.

		γ_c (%)		
Cust.	queries	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

		γ_c (%)		
Cust.	queries	Mean	95%	Max
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 2: Value of customization in the Expedia dataset (fraction of no-purchase queries is 30%)

		γ_c (%)		
Cust.	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

		γ_c (%)		
Cust.	queries	Mean	95%	Max
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 3: Value of customization in the Expedia dataset (fraction of no-purchase queries is 50%)

		γ_c (%)		
Cust.	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

		γ_c (%)		
Cust.	queries	Mean	95%	Max
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 4: Value of customization in the Expedia dataset (fraction of no-purchase queries is 70%)

8.2 Synthetic data

In this section, we present computational experiments on randomly generated problem instances to show the value of customization in CAP. In particular, we numerically compare the optimal solution of CAP and the optimal solution of the corresponding assortment optimization problem without customization. As discussed in Section 3, the latter problem corresponds to the mixed multinomial logit assortment problem which we have referred to as MMNL.

Experimental setup. We have n products. The revenue of each product is generated 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 $j \in \mathcal{M}$. The preference weights are generated independently, where for $i \in \mathcal{N}$ and $j \in \mathcal{M}$, v_{ij} is a random Bernoulli variable with parameter 1/2. We are interested in generating problem instances where customer types have strong preferences between the products. In these instances, we expect the value of customization to be particularly noticeable. Thus, we generate the preference weight v_{ij} of product i for customer type j from the Bernoulli distribution. In particular, if $v_{ij} = 1$, then customer type j is interested in product i, whereas if $v_{ij} = 0$, then the customer type is not interested in product i. 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 $K = n$, so the optimal solution of CAP is given by $S^* = \mathcal{N}$ and we offer to each customer type the optimal assortment under her MNL model. We denote z_{CAP} the optimal objective value of CAP. We solve MMNL to optimality using the standard mixed integer formulation. We use z_{MMNL} to denote the optimal objective value of MMNL. To capture the value of customization, we define the ratio

$$v_c = \frac{z_{\mathsf{CAP}} - z_{\mathsf{MMNL}}}{z_{\mathsf{MMNL}}}.$$

Here, v_c can be interpreted as the percentage gain in revenue due to customization. We consider values of $n \in \{5, 10, 15, 20\}$ and $m \in \{5, 10, 50, 100, 500\}$. Note that we restrict the number of products to at most 20 since solving a mixed integer program for MMNL becomes challenging for large size instances. For each value of n and m, we run 100 instances and report the mean, the 95% percentile and the maximum values of v_c in our 100 instances. Our results are presented in Table 5.

Results. We observe from Table 5 that the optimal expected revenue of CAP is significantly higher than the optimal expected revenue of MMNL. In particular, the average value of customization v_c is around 5% over all our test instances and the maximum value of v_c is above 20%. This shows that there is a significant value in customizing the assortments offered to each customer type as compared to offering the same assortment to all the customer types. Note that v_c increases with the number of customer types m in our numerical results and therefore there is more value in customization as the number of customer types gets larger.

		v_c (%)					v_c (%)		
n	m	Mean	95% perc.	Max	n	m	Mean	95% perc.	Max
5	5	3.53	13.47	15.76	15	5	2.48	9.67	18.95
	10	4.56	10.96	20.78		10	3.72	9.94	12.03
	50	5.92	15.29	19.42		50	4.49	10.35	19.58
	100	5.89	15.05	17.75		100	4.83	9.51	18.20
	500	6.18	15.02	17.48		500	5.70	12.76	16.84
10	5	3.30	10.84	22.91	20	5	2.47	8.02	22.06
	10	4.72	13.52	20.06		10	3.83	10.23	16.90
	50	5.69	13.38	22.41		50	3.98	9.78	15.78
	100	5.15	10.45	14.94		100	4.47	9.80	14.08
	500	5.88	13.59	16.81		500	5.15	11.21	21.21

Table 5: Value of Customization on synthetic data

9 Computational study: Performance of Augmented Greedy

We present computational experiments on randomly generated problem instances to numerically evaluate the performance of Augmented Greedy. We generate random instances of CAP in the same way as in Section 8.2. In particular, the product revenues are generated independently from the exponential distribution with parameter 1. The preference weights are generated independently such that for all $i \in \mathcal{N}$ and $j \in \mathcal{M}$, v_{ij} is a random Bernoulli variable with parameter 1/2. The customer types have equal arrival probabilities. Recall that Augmented Greedy gives $\Omega(1/\log m)$-approximation to CAP when customer types have equal arrival probabilities (Theorem 5.1).

9.1 Augmented Greedy vs. Optimal

In our first set of computational experiments, we consider small size-instances and solve CAP to optimality by enumerating over all feasible assortments of size K. We compare the optimal solution of CAP with the solution returned by Augmented Greedy. Let z_{CAP} be the optimal objective value of CAP and let z_{AG} be the expected revenue of the assortment returned by Augmented Greedy. We define the ratio

$$\gamma_1 = \frac{z_{\mathsf{AG}} - z_{\mathsf{CAP}}}{z_{\mathsf{CAP}}}.$$

Here, γ_1 captures the sub-optimality gap of the solution of Augmented Greedy. We consider values of $n \in \{5, 10, 15, 20\}$, $K \in \{2, 3, 4, 5\}$ and $m = 10$. For each value of n, m and K, we generate 100 problem instances and report our results in Table 6. The third column of Table 6 gives the number of instances where Augmented Greedy is sub-optimal. The fourth and fifth ones give the mean and the maximum values of γ_1 but only over the instances where Augmented Greedy is sub-optimal.

		instances	γ_1 (%)	
n	K	count	Mean	Max
5	2	0	0	0
	3	1	3.63	3.63
	4	0	0	0
	5	0	0	0
10	2	0	0	0
	3	1	0.05	0.05
	4	1	1.84	1.84
	5	0	0	0

		instances	γ_1 (%)	
n	K	count	Mean	Max
15	2	0	0	0
	3	1	0.14	0.14
	4	4	0.99	2.40
	5	1	0.25	0.25
20	2	0	0	0
	3	1	2.40	2.40
	4	1	4.40	4.40
	5	0	0	0

Table 6: Comparison of Augmented Greedy and Optimal

We observe from Table 6 that Augmented Greedy has a strong empirical performance. It is optimal in almost all the instances. In particular, for most of the values of n and K, there is at most one instance where Augmented Greedy is sub-optimal out of the 100 test instances. Note that the Augmented Greedy solution can be computed very efficiently with a running time less than a second even for large instances. On the other hand, enumerating over all feasible assortments to compute the optimal solution of CAP becomes challenging when we increase n or m.

9.2 Augmented Greedy vs. Greedy

In our second set of computational experiments, we consider large size-instances of CAP. Here, we can not solve CAP to optimality. Our goal is to compare Augmented Greedy with a standard greedy algorithm that iteratively adds the product providing the largest increase in the expected revenue into the assortment. Our goal is to demonstrate the practical benefit that Augmented Greedy provides over a standard greedy algorithm. Let z_{G} be the expected revenue of the assortment returned by the greedy algorithm and let z_{AG} be the expected revenue of the assortment returned by Augmented Greedy. We consider values of n from 10 to 100 with increments of 10. For each value of n we consider the cardinality $K \in \{2,3,4,5\}$. The number of customer types is $m = 10$. For each value of n, m and K, we generate 100 problem instances and focus on the instances where the solutions returned by Augmented Greedy and by the greedy algorithm are different. For the instances where these two solutions are different, we report the mean and the maximum values of

$$\gamma_2 = \frac{z_{\mathsf{AG}} - z_{\mathsf{G}}}{z_{\mathsf{G}}},$$

which captures the gap between the two solutions. Our results are summarized in Table 7. The third column gives the number of instances where z_{G} is different from z_{AG}. The fourth and fifth ones give the mean and maximum values of γ_2 over the instances where z_{G} is different from z_{AG}.

		instances	γ_2 (%)				instances	γ_2 (%)	
n	K	count	Mean	Max	n	K	count	Mean	Max
10	2	2	4.49	7.89	60	2	1	3.45	3.45
	3	1	1.46	1.46		3	4	2.14	5.24
	4	2	1.05	1.92		4	8	1.50	2.38
	5	1	0.02	0.02		5	9	1.19	3.09
20	2	2	7.01	13.97	70	2	6	2.30	6.08
	3	3	2.38	3.65		3	7	2.59	9.09
	4	5	1.66	3.80		4	7	1.18	2.56
	5	3	0.14	0.41		5	10	0.87	3.03
30	2	1	7.79	7.79	80	2	4	7.48	11.25
	3	5	2.35	6.19		3	6	4.01	14.29
	4	3	1.89	4.02		4	6	1.21	2.61
	5	2	1.25	1.58		5	8	0.68	1.97
40	2	2	0.72	1.28	90	2	5	3.96	8.73
	3	9	1.83	6.84		3	6	2.01	4.51
	4	3	1.90	3.40		4	14	2.72	6.67
	5	4	0.25	0.61		5	7	1.09	1.98
50	2	3	1.70	4.35	100	2	3	1.94	4.25
	3	5	1.71	1.93		3	3	1.30	1.98
	4	5	0.86	2.46		4	11	0.96	2.30
	5	6	0.69	1.46		5	7	0.85	2.95

Table 7: Comparison of Augmented Greedy and Greedy algorithms

We observe from Table 7 that Augmented Greedy can provide significant improvements over the standard greedy algorithm. The number of instances where there is a gap between the two algorithms gets larger as we increase the number of products n. Moreover the gap between the two solutions could go up to 14% in our test instances. Note that by construction the revenue of the solution from Augmented Greedy is always greater than or equal to the revenue of the greedy solution since the greedy solution is one of the candidate assortments in Augmented Greedy. Our results show that besides providing a theoretical performance of $\Omega(1/\log m)$, there is significant practical value in considering Augmented Greedy rather than the standard greedy algorithm, as the former improves our solutions at almost no additional computational cost.

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.

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.

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.

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.

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. 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 Hardness results

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

Maximum coverage problem. Given elements $\{1, 2, \ldots, m\}$ and sets $\{S_1, S_2, \ldots, S_n\}$, 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 NP=P, see Feige (1998).

Proof of Theorem 2.1 Consider an instance of the maximum coverage problem. Let us construct an instance of CAP. 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 $\Gamma = 1/\epsilon' - 1$. We define the preference weights as follows: for all $i \in \mathcal{N}$ and $j \in \mathcal{M}$,

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

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 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 NP=P, it is NP-hard to approximate the maximum coverage problem with a factor better than $1 - \frac{1}{e}$ (Feige (1998)), therefore 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 $\epsilon > 0$. $\square$

Next, we show that CAP is NP-hard even with two customer types (Theorem A.1). This NP-hardness result holds even when all products have the same revenue. We use a reduction from the *subset sum problem.*

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

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

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 $i \in \mathcal{N}$,

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

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 under MNL model to offer all the products in S to both customer types 1 and 2. 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 $i \in \mathcal{N}$, then we have $\sum_{i \in \mathcal{N}} y_i = K$ in an optimal solution to CAP. Let

$$X = \sum_{i \in \mathcal{N}} v_{1i} y_i \qquad \text{and} \qquad \alpha = 2 w_1 K.$$

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 $X(\alpha - X)$. Since $(X - \frac{\alpha}{2})^2 \geq 0$, we get

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

with 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., $\sum_{i \in \mathcal{N}} w_i y_i = 0$. Hence 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 Properties of the function f

In this section, we describe several properties of the functions f_j, $j \in \mathcal{M}$ and $f = \sum_{j \in \mathcal{M}} f_j$.

B.1 Monotonicity

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

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

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

B.2 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,

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

for some $\bar{S} \subseteq A \cup B$. Let us write $\bar{S}$ as $\bar{S} = S_1 \cup S_2$ where $S_1 \subseteq A$, $S_2 \subseteq B$ and $S_1 \cap S_2 = \emptyset$. We have

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

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

B.3 f is not submodular

Consider three products $n = 3$, 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 $v_1 = 1, v_2 = 1, v_3 = 100$. Consider the following sets $S = \{3\}, T = \{2, 3\}$. Let us denote the revenue function $R(S) = \frac{\sum_{i\in S} r_i v_i}{1+\sum_{i\in S} v_i}$. We have

$$\begin{aligned} &f(S) = R(\{3\}) = \frac{100}{101} \\ &f(S \cup \{1\}) = \max(R(\{1\}), R(\{1, 3\})) = R(\{1\}) = \frac{3}{2} \\ &f(T) = \max(R(\{2\}), R(\{2, 3\})) = 1 \\ &f(T \cup \{1\}) = \max(R(\{1\}), R(\{1, 2\}), R(\{1, 2, 3\})) = R(\{1, 2\}) = \frac{5}{3} \\ &f(T \cup \{1\}) - f(T) = \frac{2}{3} \\ &f(S \cup \{1\}) - f(S) = \frac{1}{2} + \frac{1}{101}. \end{aligned}$$

Hence, $f(T \cup \{1\}) - f(T) > f(S \cup \{1\}) - f(S)$ and $S \subseteq T$. Therefore, f is not submodular.

C Assortment optimization under MNL

Consider MNL model with the set of products $\mathcal{N} = \{1, \ldots, n\}$, preference weights $v_1, \ldots, v_n$ and revenues $r_1, \ldots, r_n$. The unconstrained assortment optimization problem under the MNL model is given by

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

Lemma C.1. *Let S^* be the optimal solution of problem* (C.1) *and z^* be the corresponding optimal objective value. We have*

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

The above lemma is a standard result in the literature of assortment optimization under MNL (Talluri and van Ryzin (2004); Gallego et al. (2004); Rusmevichientong and Topaloglu (2012)). It implies that we can efficiently compute the optimal MNL assortment by simply sorting the products according to their revenues. In particular, assume without loss of generality that $r_1 \geq \ldots \geq r_n$. The optimal assortment is of the form $\{1, \ldots, j\}$ for some $j \in \{1, \ldots, n\}$. So we can find the optimal assortment by checking the expected revenue for all assortments of the form $\{1, \ldots, j\}$ for $j = 1, \ldots, n$. For completeness, we give the proof of Lemma C.1.

Proof. Consider $S \subseteq \mathcal{N}$ and let us denote the revenue function $R(S) = \frac{\sum_{i \in S} r_i v_i}{1+\sum_{i \in S} v_i}$. For $S \subseteq \mathcal{N}$ and $j \in \mathcal{N} \setminus S$, we have

$$R(S \cup \{j\}) = \frac{\sum_{i \in S} r_i v_i + r_j v_j}{1 + \sum_{i \in S} v_i + v_j} = (1 - \alpha) R(S) + \alpha r_j,$$

where $\alpha = \frac{v_j}{1+\sum_{i \in S} v_i + v_j}$. Therefore, $R(S \cup \{j\}) \geq R(S)$ if and only if $r_j \geq R(S)$. Hence $j \notin S^*$ is equivalent to $R(S^* \cup \{j\}) < R(S^*)$, which is equivalent to $r_j < R(S^*)$. Therefore, $j \in S^*$ if and only if $r_j \geq R(S^*) = z^*$. □

D Proof of Lemma 6.2

Proof. Recall that $\mathcal{M} = \left\{ \bigcup_{i=1}^{L} \mathcal{M}(\ell_i) \right\}$. We split the customer types $\mathcal{M}$ as follows,

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

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 in the left hand side should be greater than $z_{\mathsf{CAP}}/2$. Let us assume without loss of generality that,

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

Let $\mathcal{I} = \{i \in [L] \mid \ell_i \text{ is even}\}$ and recall that $G_i = \mathcal{M}(\ell_i)$ for $i \in [L]$. 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 particular, G_{i^*} is the group of customer types with the highest values of θ_j among the groups of $\mathcal{M}^{\mathsf{even}}$. We construct $\mathcal{X}$ as follows. We first add all customer types in the group G_{i^*} to $\mathcal{X}$. Then, we go in a descending order from $i^* - 1$ until 1, where at each $i \in \{i^* - 1, \ldots, 2, 1\}$, we first check if i belongs to $\mathcal{I}$. If $i \in \mathcal{I}$, we consider the following set of customer types

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

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 then move to the next group of customers G_{i-1}. If $i \notin \mathcal{I}$, 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, $\mathcal{X}$ verifies the first property of our lemma.

Now, we will show the second property. Consider a customer type $j \in \mathcal{M}_{\mathsf{even}}$, hence $j \in G_i$ for some $i \in \mathcal{I}$. If $j \notin \mathcal{X}$, then by construction, there exists a customer type $q \in \mathcal{X}$ such that $f_q(S^*) > f_j(S^*)$. Moreover, q was added to $\mathcal{X}$ before we arrive at iteration i. Hence, $q \in G_{i'}$ for some $i' \in \mathcal{I}$ such that $i' > i$. By our definition of the groups G_i, this 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 $\ell_i > \ell_{i'}$. But since both ℓ_i and $\ell_{i'}$ are even, we get $\ell_i \geq \ell_{i'} + 2$. Therefore,

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

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 are at most m, we get

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

and therefore

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

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

E Proof of Lemma 6.3

Proof. First, let us address the case where T is a leaf of $\mathcal{T}$. We will show that the first statement in our lemma is verified in this case. Since T is a leaf of $\mathcal{T}$, it contains a unique group of customer types, in particular $\mathcal{C}_T = G_i$ for some $i \in \{1, \ldots, L\}$. By definition, all customer types in the group G_i have their θ_j between $1/m^{\ell_i}$ and $1/m^{\ell_i+1}$ for some $\ell_i \in \mathbb{Z}$. Moreover, our rounded values of 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 $1 \leq \theta_j m^{\ell_i+1} \leq m$, then $\mathcal{C}_T$ can be viewed 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 θ_j. 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 inputs: customer types $\mathcal{C}_T$ and cardinality k, hence for any $A \subseteq \mathcal{N}$ such that $|A| = k$,

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

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 (6.3). Therefore,

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

Now, 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 θ_j among all customer types in $\mathcal{X} \cap \mathcal{C}_{T_2}$. Let

$$q = \mathsf{argmax}\{f_j(S^*) \ : \ j \in \mathcal{X} \cap \mathcal{C}_{T_2} \cap G_\ell\}, \tag{E.1}$$

i.e., q is the customer type with the highest $f_j(S^*)$ 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 $S^*_j \subseteq S^*_q$. 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 $\ell' \in \{1, \ldots, L\}$ 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 6.2, we have for all $j \in \mathcal{X} \cap \mathcal{C}_{T_1}$,

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

The classical property of assortment optimization under MNL says that a product is in the optimal assortment if and only if its revenue is greater than the optimal expected revenue of the assortment optimization under MNL (see Lemma C.1 in Appendix C). Recall that $f_j(S^*)$ is the optimal expected revenue under MNL for customer type j with universe of products S^* and the optimal solution for this problem is S^*_j. Therefore, by the above MNL property, we have for any $j \in \mathcal{M}$,

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

In particular, for any $j \in \mathcal{X} \cap \mathcal{C}_{T_1}$ and any $i \in S^*_j$, $r_i \geq f_j(S^*) \geq f_q(S^*)$. 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 $j \in G_\ell$, then $j \in \mathcal{X} \cap \mathcal{C}_{T_2} \cap G_\ell$ and by definition (E.1) of customer type q, we have that $f_q(S^*) \geq f_j(S^*)$. If $j \notin G_\ell$, since by definition of G_ℓ, it contains the customer type with the smallest arrival probability among customer types $\mathcal{X} \cap \mathcal{C}_{T_2}$. This means that in this case $j \in G_{\ell'}$ with $\ell' > \ell$. Hence, from the first property of Lemma 6.2, 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, the classical property of MNL implies here that $S^*_q \subseteq S^*_j$ for any $j \in \mathcal{X} \cap \mathcal{C}_{T_2}$ which concludes the proof of the claim.

Let us define the following partition of A,

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

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

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

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

$$h^{\mathcal{X}\cap\mathcal{C}_{T_1}}(A_1) + h^{\mathcal{X}\cap\mathcal{C}_{T_2}}(A_1) + h^{\mathcal{X}\cap\mathcal{C}_{T_2}}(A_2) \geq h^{\mathcal{X}\cap\mathcal{C}_T}(A). \tag{E.2}$$

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

$$R_{Tk} = f^{\mathcal{C}_T}(\Delta_{Tk}) \geq \frac{1}{4}(1-\frac{1}{e}) \cdot f^{\mathcal{X}\cap\mathcal{C}_{T_2}}(A_1) \geq \frac{1}{4} \cdot (1-\frac{1}{e}) \cdot h^{\mathcal{X}\cap\mathcal{C}_{T_2}}(A_1),$$

where the last inequality follows from (6.3). 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, we get from the previous inequality that $R_{Tk} = \Omega(1/\log m) \cdot h^{\mathcal{X}\cap\mathcal{C}_T}(A)$ which means that the first property of the lemma is verified. Otherwise, if $h^{\mathcal{X}\cap\mathcal{C}_{T_2}}(A_1) < \frac{1}{\log m} \cdot h^{\mathcal{X}\cap\mathcal{C}_T}(A)$, the inequality (E.2) implies that

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

which means that the second property of the lemma is verified in this case.

$\square$