Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Maximum Load Assortment Optimization: Approximation Algorithms and Adaptivity Gaps

Omar El Housni, Marouane Ibn Brahim, Danny Segev

informs
https://pubsonline.informs.org/journal/opre

OPERATIONS RESEARCH
Vol. 74, No. 1, January–February 2026, pp. 408–429
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Methods

Maximum Load Assortment Optimization: Approximation Algorithms and Adaptivity Gaps

Omar El Housni,[a,*] **Marouane Ibn Brahim,**[a] **Danny Segev**[b]

[a] School of Operations Research and Information Engineering, Cornell Tech, Cornell University, New York, New York 10044; [b] School of Mathematical Sciences and Coller School of Management, Tel Aviv University, Tel Aviv 69978, Israel

*Corresponding author

Contact: oe46@cornell.edu, https://orcid.org/0000-0003-3097-8571 (OEH); mi262@cornell.edu, https://orcid.org/0009-0001-7415-5836 (MIB); segevdanny@tauex.tau.ac.il, https://orcid.org/0000-0003-4684-2185 (DS)

Received: September 13, 2023
Revised: June 10, 2024; November 15, 2024; February 7, 2025
Accepted: February 19, 2025
Published Online in Articles in Advance: April 8, 2025

Area of Review: Optimization

https://doi.org/10.1287/opre.2023.0495

Abstract. Motivated by modern-day applications such as attended home delivery and preference-based group scheduling, where decision makers wish to steer a large number of customers toward choosing the exact same alternative, we introduce a novel class of assortment optimization problems, referred to as *maximum load assortment optimization*. In such settings, given a universe of substitutable products, we are facing a stream of customers, each choosing between either selecting a product out of an offered assortment or opting to leave without making a selection. Assuming that these decisions are governed by the multinomial logit choice model, we define the random *load* of any underlying product as the total number of customers who select it. Our objective is to offer an assortment of products to each customer so that the expected maximum load across all products is maximized. We consider both static and dynamic formulations of the maximum load assortment optimization problem. In the static setting, a single offer set is carried throughout the entire process of customer arrivals, whereas in the dynamic setting, the decision maker offers a personalized assortment to each customer, based on the entire information available at that time. As can only be expected, both formulations present a wide range of computational challenges and analytical questions. The main contribution of this paper resides in proposing efficient algorithmic approaches for computing near-optimal static and dynamic assortment policies. In particular, we develop a polynomial time approximation scheme for the static problem formulation. Additionally, we demonstrate that an elegant policy utilizing weight-ordered assortments yields a $1/2$ approximation. Concurrently, we prove that such policies are sufficiently strong to provide a $1/4$ approximation with respect to the dynamic formulation, establishing a constant factor bound on its adaptivity gap. Finally, we design an adaptive policy whose expected maximum load is within factor $1-\epsilon$ of optimal, admitting a quasi-polynomial time implementation.

Funding: This work was supported by the National Science Foundation [Grant CMMI-2226900 to O. El Housni] and the Israel Science Foundation [Grant 1407/20 to D. Segev].
Supplemental Material: All supplemental materials, including the code, data, and files required to reproduce the results, are available at https://doi.org/10.1287/opre.2023.0495.

Keywords: assortment optimization • maximum load • approximation schemes • adaptivity gap • balls and bins • multinomial logit model

1. Introduction

Assortment optimization forms one of the most fundamental problems in revenue management, arising in a wide spectrum of application domains such as retailing and online advertising. At a high level, in such settings, the decision maker wishes to decide on a subset of products, picked out of a given universe, that will be offered to arriving customers in order to optimize a certain objective function. At least traditionally, each customer either chooses a single product from the offered assortment or decides to leave without making any purchase, with choice probabilities that are captured by a *discrete choice model*. The vast majority of assortment optimization models are guided by having either revenue maximization or sales maximization as their objective function. In the former case, each underlying product is associated with a fixed selling price, and the goal is to identify an assortment that maximizes the expected revenue due to a single representative customer, where the price of each product within this assortment is weighted by its corresponding choice probability. Problems of this form arise, for instance, when an online retailer displays a subset of products from a large universe in order to maximize the expected revenue. On the other hand, in sales maximization, our goal is to determine an assortment that maximizes the expected market share, given by the probability that a customer would purchase a product from the offered set. For instance, publishers such as Google Ads or Microsoft Ads may wish to select a subset of online ads to display, aiming to maximize the probability that customers will click on one of these ads. For a comprehensive overview of classical assortment optimization problems and their applications, we refer the reader to related surveys and books (Kök et al. 2009, Gallego and Topaloglu 2019, Phillips 2021).

1.1. Informal Model Description

In this paper, we introduce and study a new class of assortment optimization problems where, informally speaking, our goal is to identify, either statically or adaptively, assortments that would steer a large number of customers toward choosing the exact same product. Deferring the formal model formulation to be discussed in Section 2, given a universe of substitutable products, we are facing a finite stream of customers, each choosing between either selecting a product out of an offered assortment or opting to leave without making a selection. Assuming that these decisions are governed by the multinomial logit (MNL) choice model, we define the random *load* of any underlying product as the total number of customers who select it along the arrival sequence. This way, the number of customers who choose the most selected product corresponds to the maximum load across all products. Our objective is to offer an assortment of products to each customer so that the expected maximum load across all products is maximized. We refer to problem formulations along these lines as *maximum load assortment optimization*. Specifically, we consider both the static formulation (static-MLA), where a single offer set should be kept unchanged for the entire sequence of customer arrivals, and the dynamic setting (dynamic-MLA), in which the decision maker offers a personalized assortment to each customer, based on the entire information available at that time, taking into account the choices of all previously arriving customers. As we proceed to show next, the above-mentioned objective function is motivated by real-life applications in e-commerce such as attended home delivery (AHD), where customers have the option to select their delivery slot, as well as in scheduling platforms, where users select a common time slot to meet.

1.2. Attended Home Delivery

Online supermarket chains such as WholeFoods, FreshDirect, and AmazonFresh provide customers with various delivery time slots to choose from, based on their individual preferences. Similarly, e-retailers such as Amazon, Wayfair, and Walmart allow their customers to select an appealing delivery time among the available options. The dominant business model in the grocery delivery sector is known as AHD (Manerba et al. 2018). This model entails the customer's presence during the delivery process, necessitating an agreement on a specific time slot between the e-grocer and the customer. To optimize delivery costs, e-commerce platforms are interested in *packing* as many customers as possible from the same geographical area into the same time slot. In their survey on this topic, Waßmuth et al. (2023, section 2.2) highlight the importance of managing customer demand: "Demand management aims to manage the resulting tradeoffs between captured demand (revenue) and assembly and delivery efficiency (costs)."

In the context of AHD, there are two primary strategies for managing customer demand: offering and pricing. In the offering strategy, decision makers determine which delivery time slots will be presented to customers and which slots will be hidden (Casazza et al. 2016, Truden et al. 2022, Waßmuth et al. 2023, van der Hagen et al. 2024). For example, Mackert (2019) introduces a dynamic time slot management framework under the generalized attraction model (Gallego et al. 2015), which starts by approximating the opportunity cost of a given customer request, and then employs this approximation to formulate a nonlinear integer program and its linearization, in order to determine the time slot assortment. On the other hand, in the pricing strategy, prices are assigned to each delivery slot in order to influence customers choices (Campbell and Savelsbergh 2006). For instance, Yang and Strauss (2017) propose a dynamic programming framework for the delivery time slot pricing problem, and show in MNL-based simulation that these pricing policies can improve profitability by more than 2% compared with simple fixed price policies. In this paper, we focus on exploiting the offering strategy to steer customers toward selecting the same time slot. To this end, while booking their delivery times, the platform can guide customers by strategically determining the assortment of time slots to offer. This objective seamlessly aligns with our framework, where each delivery time slot can be viewed as a product, meaning that the "load" of each product represents the number of customers who select its corresponding time slot. Consequently, our aim is to determine an assortment of time slots that maximizes the expected maximum load across all available time slots. Interestingly, Amorim et al. (2024) have recently investigated the effect of time slot management in the context of AHD. In particular, their MNL-based study concludes that retailers with the ability to tailor their time slots offering to specific customer segments enjoy a 9% increase in shipping revenue. These conclusions further emphasize the importance of introducing and studying frameworks that align with time slot management and highlight its significant managerial implications.

1.3. Preference-Based Group Scheduling

When scheduling a group meeting, the overarching goal is to identify the most suitable time slot from a given set of options, that is, one that accommodates the maximum number of attendees. Platforms such as Doodle and When2meet often rely on users selecting a preferred time slot from the available choices. However, the individual choice made by each user is influenced by the available options, due to substitution effects. Therefore, to maximize the likelihood of users selecting the same time slot, decision makers can carefully curate the assortment of offered time slots. This approach is known as preference-based group scheduling (Brzozowski et al. 2006, Berry et al. 2007). That said, to our knowledge, previous studies have not approached such questions from the perspective of assortment optimization, nor have they utilized discrete choice models to deal with customer preferences. Our framework effectively captures this scenario, by viewing each time slot as a product, again implying that the load of each product represents the number of users who select that time slot. Thus, our objective would be to determine an assortment of time slots that maximizes the expected maximum load among all available options.

1.4. Fundamental Challenges

As readers would quickly find out by examining our model formulations, whether one considers static or adaptive settings, coming up with efficient algorithmic approaches that can be rigorously analyzed appears to be a very challenging goal. To better understand where some hurdles are emerging from, we should bear in mind the conceptual tradeoff between offering an extensive set of products versus a more focused set, due to two competing effects. On one hand, providing a wide array of products grants customers more choices, reducing their likelihood to leave the market without making a selection, and potentially increasing the maximum load. On the other hand, offering too many products may disperse customer demand across all available choices. As our objective is to guide customers toward selecting the same product, this dispersion can potentially diminish the maximum load.

Let us proceed by briefly highlighting some fundamental challenges in addressing both problem formulations. In the static setting, the first and foremost challenge revolves around the highly nonlinear nature of the objective function. Unlike revenue or sales maximization, we are considering a novel objective function, appearing to be very different from classical settings in the assortment optimization literature. Among other missing pieces, we are not aware of any integer programming formulations or linear relaxations for the problem in question. In fact, even computing the expected maximum load for a given static assortment is very much unclear at first glance. In the dynamic setting, the state space of every conceivable dynamic program describing this problem is exponential in size. Therefore, by directly solving natural dynamic programs, we would not end up with efficient algorithmic approaches. Moreover, as we explain in the sequel, the Bellman equations associated with such dynamic programs include optimizing over the seemingly unstructured collection of all relevant assortments, which generally poses a complex challenge by itself. On top of these obstacles, we will discuss additional challenges in subsequent sections, as soon as they can be better digested.

1.5. Main Contributions

The primary contribution of this paper resides in developing a unified optimization framework with provably near-optimal performance guarantees for both formulations of the maximum load assortment optimization problem. In the static setting, we first present a polynomial time evaluation oracle to compute the expected maximum load of a given assortment. Then, by uncovering well-hidden structural properties of the objective function, we provide an elegant constant factor approximation for static-MLA. As our main result for the static setting, we present a polynomial time approximation scheme (PTAS). For the dynamic formulation, by developing novel coupling arguments in this context, we first establish a constant factor bound on its adaptivity gap. Moreover, we devise a $(1-\epsilon)$-approximate adaptive policy that can be computed in quasi-polynomial time. We proceed by providing refined details on our main contributions.

1.5.1. Static Maximum Load Assortment Optimization

• *Polynomial time evaluation oracle for the expected maximum load.* The first challenge in the static formulation consists of the seemingly-simple question of evaluating the objective function of static-MLA for a given assortment, that is, computing its expected maximum load. In fact, even though the latter admits a closed-form expression, it requires summing over exponentially many terms that arise from the multinomial distribution. Our first contribution is to design a polynomial time evaluation oracle for computing the expected maximum load of a given static assortment. Our algorithm, whose specifics are given in Section 3.1, builds on the work of Frey (2009) who designed polynomial time procedures to evaluate *rectangular probabilities* for the multinomial distribution. In essence, we show that the expected maximum load function can be computed through polynomially many external calls to evaluate rectangular probabilities.

• *A* $1/2$ *approximation via preference weight–ordered assortments.* Prior to presenting our main result regarding static-MLA, we propose in Section 3.3 an elegant and easy-to-implement way to obtain a $1/2$ approximation, utilizing *preference weight–ordered assortments*. In a nutshell, such assortments prioritize products with higher preference weights. Specifically, when a product is included in a preference weight–ordered assortment, all products with higher preference weights are included as well. Interestingly, we prove that there exists a preference weight–ordered assortment whose expected maximum load is within a factor 1/2 of the optimum. Our policy then examines all such assortments, of which there are only linearly many, picking the best via our previously mentioned evaluation oracle for their expected maximum load. As a side note, avid readers may be familiar with the notion of "revenue-ordered" assortments, which has been explored and exploited in early literature. Most notably, in revenue maximization under the MNL model, optimal assortments are known to be revenue ordered (Talluri and Van Ryzin 2004). That said, beyond the natural resemblance through a certain parametric order, the analysis of preference weight–ordered assortments turns out to be entirely different and requires new analytical ideas.

• *PTAS.* Our main technical contribution with respect to the static formulation resides in developing a PTAS for this setting, whose specifics are provided in Section 3.4. Namely, for any fixed $\epsilon > 0$, our algorithm constructs in polynomial time an assortment whose expected maximum load is within factor $1 - \epsilon$ of the optimum. To derive this result, we prove the existence of a polynomially sized family of highly structured assortments via efficient enumeration ideas. We refer to these assortments as being block based, showing that at least one such assortment yields a $(1 - \epsilon)$ approximation. Finally, using our polynomial time evaluation oracle, we enumerate over all block-based assortments and pick the best one. We should note that, despite our best efforts in studying the computational complexity of static-MLA, we still do not know whether this problem is NP-hard or not. This difficulty mainly arises due to the nature of our objective function, as we are unaware of NP-hard problems with similar structure that would serve as candidates for potential reductions. Hence, attaining complexity lower bounds that will match our algorithmic guarantees remains an intriguing open question and is further discussed in Section 7.

• *Numerical analysis.* Complementing the aforementioned theoretical contributions, we conduct a series of numerical experiments to examine how optimal assortments behave with respect to the model primitives. Our analysis exhibits a notable tendency for the optimal static assortment to decrease in size as the number of arriving customers increases, as well as when the preference weights increase. Specifically, our experiments show that offering the whole universe of products may become optimal for instances with small preference weights. This observation is particularly significant in instances with a smaller number of customers. These results are reported in Section 6.

1.5.2. Dynamic Maximum Load Assortment Optimization

• *Adaptivity gap.* Our first line of investigation examines questions related to the adaptivity gap of the maximum load assortment optimization problem. In this context, the adaptivity gap is defined as the maximal ratio between the objective values of dynamic-MLA and static-MLA over all possible instances. This measure quantifies the value of introducing adaptivity, quantifying the improvement gained by employing a dynamic policy instead of a static one. In Section 4, we prove the existence of a static policy, utilizing preference-weight-ordered assortments, whose expected maximum load is within factor 1/4 of the adaptive optimum, implying that the adaptivity gap is surprisingly bounded by four. This result immediately translates to a polynomial time 1/4 approximation for dynamic-MLA. Moreover, when all products have the same preference weight, we improve the adaptivity gap to two. In the opposite direction, we present a family of instances demonstrating that the adaptivity gap of dynamic-MLA is at least 4/3. Additionally, in Online Appendix C, we present numerical experiments studying how the adaptivity gap behaves under different parametric regimes. This numerical section allows readers to gain a more concrete understanding of the inherent gap between the optimal static and dynamic objectives. Concurrently, these experiments motivate us to construct the family of instances yielding the aforementioned 4/3 lower bound on the adaptivity gap.

• *A $(1 - \epsilon)$-approximate adaptive policy in quasi-polynomial time.* Our cornerstone technical contribution in relation to dynamic-MLA resides is devising a quasi-polynomial time adaptive policy whose expected maximum load is within factor $1 - \epsilon$ of the optimum. This policy, whose finer details are discussed in Section 5, builds on two key ideas. Firstly, rather than attempting to solve an exponentially sized natural dynamic program, we demonstrate that its state space can be shrunk to a quasi-polynomial scale while only sacrificing an $O(\epsilon)$-factor in optimality. More specifically, we observe that once a sufficiently large load is attained, the expected marginal gain from offering any further assortments becomes negligible in comparison with the already-attained maximum load. Consequently, we can effectively terminate the arrival process (i.e., offer the empty assortment from this point on), which significantly reduces the state space size. Secondly, to compute an optimal action for the resulting recursive equations, an assortment-like optimization problem needs to be solved at each stage. We argue that this problem can be reformulated as an unconstrained revenue maximization question under the multinomial logit model, which can indeed be solved in polynomial time. It is worth mentioning that quasi-polynomial time approximation schemes have gained popularity in various domains including assortment optimization, network design, scheduling, computational game theory, and graph algorithms. Although presenting an exhaustive overview of such results would be impractical, we refer the reader to selected papers (Chekuri and Khanna 2002, Arora and Karakostas 2003, Lipton et al. 2003, Bansal et al. 2006, Remy and Steger 2009, Chan and Elbassioni 2011, Adamaszek and Wiese 2014, Das and Mathieu 2015, Mustafa et al. 2015, Désir et al. 2021, Aouad and Segev 2023), which highlight the ubiquitous nature of quasi-polynomial time approximation schemes.

1.6. Related Literature

In what follows, we discuss three lines of research that are directly relevant to our work. Firstly, we discuss the MNL, which stands as one of the most widespread choice models in both theoretical and practical domains. Secondly, we provide a concise overview of the assortment optimization literature, emphasizing how our model fits within this body of research. Lastly, we discuss several classic balls and bins problems, highlighting the relevant connections and similarities between these problems and our own setting.

1.6.1. MNL. The MNL is widely regarded as the predominant choice model employed by the revenue management community to capture customer behavior when selecting from a given assortment. This model was initially introduced by Luce (1959), with subsequent works by McFadden (1973) and Hausman and McFadden (1984) further refining its specification. Informally, the MNL models assigns a preference weight to each product. Then, each product is chosen with probability proportional to its preference weight, thereby capturing the substitution effect that occurs between various alternatives within any given assortment. The model's simplicity in calculating choice probabilities, its predictive power, and computational tractability have all contributed to its widespread adoption and extensive study in various domains. Some of these directions are evidenced by research works such as those of Mahajan and Van Ryzin (2001), Talluri and Van Ryzin (2004), Rusmevichientong et al. (2014), Sumida et al. (2021), Gao et al. (2021), El Housni and Topaloglu (2023), and Bai et al. (2025) to mention a few. For a comprehensive understanding and further references, we refer the reader to chapter 4 of the book by Gallego and Topaloglu (2019).

1.6.2. Assortment Optimization. Assortment optimization represents a long-standing research domain within revenue management, seeking to address fundamental questions regarding the selection of offer sets for customers under various choice models. Here, the typical goal is to optimize performance metrics such as revenue, market share, and engagement. Over the past decades, this field has witnessed substantial growth, resulting in an extensive literature encompassing different algorithmic developments under various choice models, such as MNL (Talluri and Van Ryzin 2004, Rusmevichientong et al. 2014, Aouad et al. 2021), Markov chain (Blanchet et al. 2016, Feldman and Topaloglu 2017), nested logit (Davis et al. 2014, Gallego and Topaloglu 2014), and nonparametric choice models (Farias et al. 2013, Aouad et al. 2018). For a comprehensive study and further references, we refer the reader to related surveys and books (Kök et al. 2009, Gallego and Topaloglu 2019, Phillips 2021). As previously mentioned, it is important to note that our work diverges from the classic assortment optimization literature in terms of the objective function we optimize. To the best of our knowledge, this paper is the first study to investigate the maximum load objective function from an assortment optimization perspective.

1.6.3. Balls and Bins. In its most general setting, the literature on balls and bins explores the outcomes of randomly placing m balls into n bins. This topic finds numerous applications, with load balancing and hashing being arguably the most commonly known ones (Mitzenmacher and Upfal 2017, Mirrokni et al. 2018). Relating such questions to our setting, each customer can be viewed as a ball, whereas each product can be represented by a bin. The probability of a particular ball falling into a specific bin corresponds to the likelihood of a particular customer selecting a particular product. In their seminal work, Raab and Steger (1998) provide a comprehensive analysis of the maximum number of balls in any bin, offering precise upper and lower bounds that hold asymptotically. Specifically, for n balls and n bins with equal probabilities, the expected maximum load is $(1 + o(1)) \cdot \frac{\log n}{\log \log n}$ with high probability. In a different direction, Azar et al. (1994) proved a significant drop in the maximum load to $\frac{\log \log n}{\log 2} + O(1)$ with high probability, when each ball is placed in the least loaded out of two randomly chosen bins. It is important to point out that the literature on balls and bins primarily focuses on load balancing rather than on load maximizing applications, where one actually wishes to overpack bins by actively selecting which bins to make use of, given the constant presence of an outside option. That being said, we find that some well-known results still offer preliminary insights. However, to our knowledge, none of these results are directly relevant to statically or adaptively making assortment decisions in order to optimize the maximum load. In other words, our paper is the first to study balls-and-bins–like problems within the context of choice modeling and assortment optimization.

2. Problem Formulation

2.1. MNL Choice Model

In what follows, we begin by explaining how the MNL choice model is formally defined. To this end, let $\mathcal{N} = \{1, 2, \ldots, n\}$ be the universe of products at our disposal, where each product $i \in \mathcal{N}$ is associated with a *preference weight* $v_i > 0$. In addition, the option of not selecting any of these products will be symbolically represented as product 0, referred to as the no-purchase or no-selection option, with a preference weight of $v_0 = 1$. Although the precise meaning of these parameters will be explained below, we mention in passing that the preference weight assigned to each product reflects its level of attractiveness, meaning that higher preference weights would indicate a greater level of popularity.

With these conventions, an assortment (or an offer set) is simply a subset of products $S \subseteq \mathcal{N}$. For convenience, we make use of $S_+ = S \cup \{0\}$ to denote the inclusion of the no-purchase option within this assortment. We define the weight $v(S)$ of an assortment S simply as the sum of the preference weights of its products, namely $v(S) = \sum_{i \in S} v_i$. Now, when any given assortment $S \subseteq \mathcal{N}$ is offered to an arriving customer, the MNL model prescribes a probability of $\phi_i(S) = \frac{v_i}{v(S_+)} = \frac{v_i}{1 + \sum_{j \in S} v_j}$ for picking product $i \in S$ as the one to be purchased. Alternatively, this customer may decide to avoid selecting any of these products (i.e., picking the no-purchase option), which happens with the complementary probability, $\phi_0(S) = \frac{1}{1 + \sum_{j \in S} v_j}$.

2.2. Stream of Customers

Next, we introduce the static formulation of the maximum load assortment problem, followed by the presentation of its dynamic counterpart. In both formulations, we will be facing a finite stream of T customers, arriving one after the other, and we therefore refer to these customers by their arrival indices, $1, \ldots, T$. We assume that the choice of each customer among any offered assortment is governed by the aforementioned MNL model, meaning in particular that their purchasing decisions are mutually independent.

2.3. Static-MLA

In the static setting, we will be operating under the restriction that all customers should be offered the exact same assortment of products throughout the arrival sequence. Specifically, consider an assortment $S \subseteq \mathcal{N}$. For any product $i \in S_+$ and for any customer $t \in [T]$, we define a Bernoulli random variable $X_{it}(S)$ to indicate whether customer t selects product i or not. Because customer t chooses this product with probability $\phi_i(S)$, we have $\mathbb{P}(X_{it}(S) = 1) = \phi_i(S)$. As such, $\sum_{i \in S+} X_{it}(S) = 1$, reflecting the fact that each customer chooses exactly one product from the assortment S or decides not to select any product at all. In addition, because customers' decisions are independent, the indicators $\{X_{it}(S)\}_{t \in [T]}$ of different customers are mutually independent.

Given an offered assortment S, we define the *load* of product $i \in \mathcal{N}$ as the total number of customers who select this product. This random quantity will be designated by $L_i(S)$, noting that it can be expressed as $L_i(S) = \sum_{t=1}^{T} X_{it}(S)$. We use $L_0(S) = \sum_{t=1}^{T} X_{0t}(S)$ to denote the no-purchase load on offering the assortment S, that is, the number of customers who did not select any product. Finally, $M(S)$ will stand for the maximum load over all products, that is, $M(S) = \max_{i \in S} L_i(S)$. Hence, the number of customers who choose the most selected product corresponds to the maximum load across all products. Our optimization problem consists in computing an assortment that maximizes the expected maximum load. We refer to this problem as static-MLA, compactly formulated as follows:

$$\max_{S \subseteq \mathcal{N}} \mathbb{E}(M(S)). \qquad \text{(Static-MLA)}$$

2.3.1. Closed-Form Expression for the Maximum Load.

Given this formulation, we first observe that efficiently computing the expected maximum load $\mathbb{E}(M(S))$ of a given assortment S is a nontrivial task. Prior to developing an efficient algorithm for this purpose, let us start by deriving a supposedly straightforward closed-form expression. Consider an assortment $S \subseteq \mathcal{N}$ and suppose without loss of generality that $S = \{1, \ldots, k\}$ for some $k \le n$. For each product $i \in S_+$, its corresponding random load $L_i(S)$ clearly follows a binomial distribution of mass parameter T and probability of success $\phi_i(S)$. However, these random variables are correlated, since $\sum_{i \in S_+} L_i(S) = T$. In fact, the load vector $\mathbf{L}(S) := (L_0(S), \ldots, L_k(S))$ is a random vector that follows a multinomial distribution. In particular, for every $\boldsymbol{\ell} := (\ell_0, \ldots, \ell_k) \in \mathbb{N}^{|S_+|}$ with $\sum_{i \in S_+} \ell_i = T$, we have $\mathbb{P}(\mathbf{L}(S) = \boldsymbol{\ell}) = \binom{T}{\ell_0, \ldots, \ell_k} \cdot \prod_{i \in S_+} (\phi_i(S))^{\ell_i}$, where $\binom{T}{\ell_0, \ldots, \ell_k} := \frac{T!}{\ell_0! \cdots \ell_k!}$ is the multinomial coefficient. As such, a direct expression for the expected maximum load is given by

$$\mathbb{E}(M(S)) = \sum_{\substack{\boldsymbol{\ell} \in \mathbb{N}^{k+1}: \\ \sum_{i \in S_+} \ell_i = T}} \mathbb{P}(\mathbf{L}(S) = \boldsymbol{\ell}) \cdot \max_{i \in S} \ell_i. \qquad (1)$$

However, in this representation, we sum over an exponential number of terms, $\binom{T+k}{k}$, which makes this computation intractable. In Section 3.1, we provide a polynomial time algorithm to compute the expected maximum load for any given assortment.

2.4. Dynamic-MLA

In the dynamic setting, customers arrive one after the other, allowing the decision maker to tailor the assortment offered to each customer based on the choices observed for previously arriving customers. In particular, at each time period, we have access to the current load vector, which provides the number of customers who have selected each product up to that point. Based on this information, we wish to determine a personalized assortment that will be offered to the next arriving customer. As such, the solution concept in this setting corresponds to an adaptive policy, captured by a function that takes as input the current system state (i.e., the number of customers remaining and the current load vector) and returns an assortment to offer to the next customer. The objective is to propose an adaptive policy that maximizes the expected maximum load over all products on termination of the arrival stream.

2.4.1. Dynamic Programming Representation. To formalize the dynamic setting, we take the view of a dynamic program that determines the actions taken by an optimal policy, that is, the personalized assortments that will be offered to arriving customers. For this purpose, we consider a planning horizon consisting of T periods, each with a single customer arrival. To describe the system state at the beginning of any time period, we introduce the state variable $\boldsymbol{\ell} = (\ell_i : i \in \mathcal{N})$, where ℓ_i represents the number of customers who have selected product i up to that point. For each time period $t = 1, \ldots, T$, we use $\mathcal{M}_t(\boldsymbol{\ell})$ to denote the optimal expected maximum load when there are t customers remaining in the planning horizon, and the system's state at the beginning of this period is characterized by the load vector $\boldsymbol{\ell}$. By employing $\mathbf{e}_i \in \mathbb{R}^n_+$ to represent the ith unit vector, we can compute the value functions $\{\mathcal{M}_t\}_{t\in[T]}$ via the following dynamic program:

$$\mathcal{M}_t(\boldsymbol{\ell}) = \max_{S\subseteq\mathcal{N}} \left(\mathcal{M}_{t-1}(\boldsymbol{\ell}) \cdot \phi_0(S) + \sum_{i\in S} \mathcal{M}_{t-1}(\boldsymbol{\ell} + \mathbf{e}_i) \cdot \phi_i(S)\right), \tag{Dynamic-MLA}$$

with the boundary condition $\mathcal{M}_0(\boldsymbol{\ell}) = \max_{i\in[n]} \ell_i$. To better understand the recursive equation above, note that when the current load vector is $\boldsymbol{\ell}$ and we offer the assortment S, the first possible outcome is that the currently arriving customer will choose the no-selection option, with probability $\phi_0(S)$, in which case the load vector remains unchanged. The second outcome corresponds to choosing one of the products $i \in S$, with probability $\phi_i(S)$; here, the load vector $\boldsymbol{\ell}$ is updated to $\boldsymbol{\ell} + \mathbf{e}_i$. Clearly, the optimal expected maximum load in the entire horizon is given by $\mathcal{M}_T(\mathbf{0})$.

It is imperative to mention that this formulation should be viewed as being an explicit characterization of optimal adaptive policies rather than as an efficiently implementable algorithm, due to being defined over a state space of exponential size, $\Omega(T^n)$. Moreover, a careful inspection of dynamic-MLA shows that each of the value functions $\mathcal{M}_t(\cdot)$ is recursively obtained by only considering $\mathcal{M}_{t-1}(\cdot)$-related terms. However, these equations by themselves ask us to solve an assortment-like optimization problem. Quite surprisingly, in Online Appendix D.5, we show that this inner problem can be reformulated as an unconstrained revenue maximization question under the MNL model, which can be solved in polynomial time.

3. Static Setting: Approximation Algorithms

In this section, we present our main algorithmic results for static-MLA, eventually showing that this setting can be efficiently approximated within any degree of accuracy. Toward this objective, in Section 3.1, we first provide a polynomial time evaluation oracle for computing the expected maximum load of a given assortment. In Section 3.2, we establish a number of structural lemmas that will be useful in analyzing our algorithmic framework. Using these claims, we show in Section 3.3 that an elegant policy based on preference-weight-ordered assortments yields a 1/2 approximation for static-MLA. In Section 3.4, we present our main contribution for the static formulation, showing that it admits a PTAS. Finally, in Section 3.5, we study the special case where the number of customers T is very large and characterize optimal assortments in this regime.

3.1. Polynomial Time Evaluation Oracle

As previously mentioned, one of the basic challenges in addressing static-MLA resides in simply evaluating the objective function of a given assortment. As discussed in Section 2.3, computing the expected maximum load via Representation (1) requires summing over exponentially many terms, which is clearly not a tractable approach. Our first contribution is to provide a polynomial time algorithm for computing the expected maximum load function.

Theorem 1. *The expected maximum load of any assortment can be computed in $O(n^2T^3)$ time.*

In a nutshell, our algorithm builds on the work of Frey (2009) and Lebrun (2013), who designed polynomial time procedures to evaluate *rectangular probabilities* for the multinomial distribution. Specifically, let $\mathbf{L}$ be a random vector that follows a multinomial distribution. A rectangular probability is the probability of a so-called *rectangular event*, of the form $\{\mathbf{a} \le \mathbf{L} \le \mathbf{b}\}$, where $\mathbf{a}$ and $\mathbf{b}$ are integer vectors.

3.1.1. Overview of Frey's Algorithm. Frey (2009) proposed a sophisticated approach to compute rectangular probabilities in polynomial time. Let us present a brief overview of his ideas. Recall that $\mathbf{L}$ is a random k-dimensional vector that follows a multinomial distribution, where T and $(p_1, \ldots, p_k)$ are the number of trials and the probability vector, respectively. To compute $\mathbb{P}(\mathbf{a} \le \mathbf{L} \le \mathbf{b})$, we start by summing over all possible realizations $\mathbf{a} \le \ell \le \mathbf{b}$ of the load vector $\mathbf{L}$:

$$\mathbb{P}(\mathbf{a} \le \mathbf{L} \le \mathbf{b}) = \sum_{\ell_1=a_1}^{b_1} \cdots \sum_{\ell_k=a_k}^{b_k} \mathbb{1}(\|\ell\|_1 = T) \cdot \frac{T!}{\ell_1! \cdots \ell_k!} \prod_{i=1}^{k} p_i^{\ell_i},$$

where $\|\ell\|_1 = \sum_{i=1}^{k} \ell_i$. Then, noticing that $\ell_i \ge a_i$ for all $i = 1, \ldots, k$ in the summations above, we can perform the following factorization:

$$\begin{aligned}
&\mathbb{P}(\mathbf{a} \le \mathbf{L} \le \mathbf{b}) \\
&= \frac{T! \cdot p_1^{a_1} \cdots p_k^{a_k}}{a_1! \cdots a_k!} \sum_{\ell_1=a_1}^{b_1} \cdots \sum_{\ell_k=a_k}^{b_k} \mathbb{1}(\|\ell\|_1 = T) \cdot \frac{1}{\frac{\ell_1!}{a_1!} \cdots \frac{\ell_k!}{a_k!}} \prod_{i=1}^{k} p_i^{\ell_i - a_i} \\
&= \frac{T! \cdot p_1^{a_1} \cdots p_k^{a_k}}{a_1! \cdots a_k!} \sum_{\ell_1=a_1}^{b_1} \cdots \sum_{\ell_k=a_k}^{b_k} \mathbb{1}(\|\ell\|_1 = T) \cdot \prod_{i=1}^{k} \prod_{j=1}^{\ell_i - a_i} \frac{p_i}{a_i + j}.
\end{aligned}$$

To proceed from this point on, notice that each summand in the last expression is the product of $T - \sum_{i=1}^{k} a_i$ terms. To compute each summand, a naive algorithm would simply start with $p_1/(a_1+1)$, then proceed to multiply it by $p_1/(a_1+2)$ or $p_2/(a_2+1)$, depending on whether $\ell_1 - a_1 > 1$ or not, and so on for each of the summands. However, Frey's algorithm proposes a way to gather all summands that are multiplied by the same factor $p_i/(a_i + j)$ at the same time step and multiply their sum by $p_i/(a_i + j)$, which allows us to only perform a single multiplication for those summands at that time step rather than a separate multiplication for each. Finally, using recursion, this algorithm computes the final values at the last step, and returns their sum. In what follows, we show how Frey's algorithm is employed to design our polynomial time evaluation oracle.

3.1.2. Preliminaries. Consider an arbitrarily structured assortment $S \subseteq \mathcal{N}$ and suppose without loss of generality that $S = \{1, \ldots, k\}$. We remind the reader that the random variable $L_i(S)$ stands for the load of product $i \in S$, with $\mathbf{L}(S) = (L_1(S), \ldots, L_k(S))$ being the overall load vector. Additionally, $M(S)$ is the random variable that refers to the maximum load across the products in S, that is, $M(S) = \max_{i \in S} L_i(S)$. As argued in Section 2.3, the load vector $\mathbf{L}(S)$ follows a multinomial distribution. In what follows, we explain how to compute $\mathbb{E}(M(S))$ using only a polynomial number of externals calls to evaluate rectangular probabilities. To this end, noting that

$$\mathbb{E}(M(S)) = \sum_{\ell=1}^{T} \ell \cdot \mathbb{P}(M(S) = \ell), \qquad (2)$$

it suffices to show how to efficiently compute each of the terms $\mathbb{P}(M(S) = \ell)$. In turn, we write each event $\{M(S) = \ell\}$ as a partition into $O(n)$ rectangular events with respect to the random vector $\mathbf{L}(S)$. Specifically, for $1 \le \ell \le T$ and $1 \le j \le k$, we define the event $F_{\ell j}(S)$ as

$$\begin{aligned}
F_{\ell j}(S) &= \left[\bigwedge_{i=1}^{j-1} \{L_i(S) < \ell\}\right] \wedge \{L_j(S) = \ell\} \\
&\wedge \left[\bigwedge_{i=j+1}^{k} \{L_i(S) \le \ell\}\right] = \{\mathbf{a}_{\ell j} \le \mathbf{L}(S) \le \mathbf{b}_{\ell j}\}, \qquad (3)
\end{aligned}$$

where $\mathbf{a}_{\ell j} = \ell \cdot \mathbf{e}_j$ and $\mathbf{b}_{\ell j} = (\ell - 1) \cdot \sum_{i=1}^{j-1} \mathbf{e}_i + \ell \cdot \sum_{i=j}^{k} \mathbf{e}_i$. Here, $F_{\ell j}(S)$ corresponds to the event where the maximum load is equal to ℓ, and product j is the minimal index product that attains this load. The above expression implies that $F_{\ell j}(S)$ is a rectangular event, meaning that its probability can be computed using Frey's algorithm.

3.1.3. Computing $\mathbb{E}(M(S))$. The next lemma shows how to utilize these rectangular events to compute $\mathbb{E}(M(S))$ for any assortment $S \subseteq \mathcal{N}$.

Lemma 1. *For any assortment $S = \{1, \ldots, k\} \subseteq \mathcal{N}$, we have $\mathbb{E}(M(S)) = \sum_{\ell=1}^{T} [\ell \cdot \sum_{j=1}^{k} \mathbb{P}(F_{\ell j}(S))]$.*

Proof. For convenience, we denote the random variables $M(S)$ and $L_j(S)$ simply by M and L_j; similarly, the events $F_{\ell j}(S)$ will be replaced by $F_{\ell j}$. Fixing some $\ell = 0, \ldots, T$, we will show that $(F_{\ell j})_{j=1,\ldots,k}$ is a partition of the event $\{M = \ell\}$; that is, the union of the events $(F_{\ell j})_{j=1,\ldots,k}$ is precisely $\{M = \ell\}$ and these events are mutually exclusive. Consequently, $\mathbb{P}(M = \ell) = \sum_{j=1}^{k} \mathbb{P}(F_{\ell j})$, and replacing this expression in Equation (2) yields the desired result.

First, we show the events $(F_{\ell j})_{j=1,\ldots,k}$ are mutually exclusive; that is, for all $j_1 \ne j_2$, we have $F_{\ell j_1} \cap F_{\ell j_2} = \emptyset$. To verify this claim, suppose without loss of generality that $j_1 < j_2$. In the event $F_{\ell j_2}$, we have by definition $L_{j_1} < \ell$ because $j_1 < j_2$. In particular, $L_{j_1} \ne \ell$, which implies that $\mathbf{L} \notin F_{\ell j_1}$. Hence, $F_{\ell j_1} \cap F_{\ell j_2} = \emptyset$. Second, let us show that $\bigvee_{j=1}^{k} F_{\ell j} = \{M = \ell\}$. First, by definition, the maximum load in any event $F_{\ell j}$ is exactly ℓ, and therefore $\bigvee_{j=1}^{k} F_{\ell j} \subseteq \{M = \ell\}$. In the opposite direction, suppose that $M = \ell$. Then, at least one product has a load of ℓ and all other products have a load of at most ℓ. Let $j_{\min}$ be the lowest-index product with a load of exactly ℓ, that is, $j_{\min} = \min\{i = 1, \ldots, k | L_i = \ell\}$. By definition of $j_{\min}$, we have $L_i < \ell$ for all $i = 1, \ldots, j_{\min} - 1$ and $L_{j_{\min}} = \ell$. Also, because $M = \ell$ by supposition, $L_i \le \ell$ for all $i = j_{\min} + 1, \ldots, k$, meaning that $\{M = \ell\} \subseteq F_{\ell j_{\min}} \subseteq \bigvee_{j=1}^{k} F_{\ell j}$. □

3.1.4. Concluding the Proof of Theorem 1. We consider the running time incurred by computing $\mathbb{E}(M(S))$ via Lemma 1. For each of the events $\{M(S) = \ell\}$, we have to compute $O(n)$ rectangular probabilities. Because Frey's algorithm admits an $O(nT^2)$-time implementation, we arrive at $O(n^2T^2)$ operations per event. In turn, Equation (2) involves $O(T)$ such events, amounting to an overall running time of $O(n^2T^3)$, precisely as stated in Theorem 1.

3.2. Structural Lemmas

In what follows, we shed light on a number of structural claims that will be useful in presenting our algorithmic framework. In Lemmas 2 and 3, we introduce two operations, referred to as *Merge* and *Transfer*, showing that their application to any assortment does not decrease the expected maximum load. In Lemmas 4 and 5, we show that minor alterations of the instance parameters (choice probabilities or preference weights) yield a correspondingly small deviation from the expected maximum load. Before stating these lemmas, let us introduce the following definition. We say that a product i is *lighter* (respectively, *heavier*) than a product j, if $v_i \le v_j$ (respectively, $v_i \ge v_j$), emphasizing that we have weak inequalities in both definitions.

3.2.1. Operation 1: Merge. Consider an assortment S and let i and j be two products in this assortment, with respective preference weights v_i and v_j. The operation of merging products i and j consists of replacing both products with a single new product, whose preference weight is $v_i + v_j$. In the next lemma, whose proof is presented in Online Appendix A.1, we show that the merge operation cannot decrease the expected maximum load.

Lemma 2. *Consider an assortment $S \subseteq \mathcal{N}$ and let $\tilde{S}$ be the assortment resulting from merging any two products of S. Then, $\mathbb{E}(M(\tilde{S})) \ge \mathbb{E}(M(S))$.*

Roughly speaking, the main idea behind proving this result argues that, when we merge products i and j, simple coupling arguments show that the maximum load of $\tilde{S}$ is stochastically larger than that of S. In fact, the load of the merged product is equal in distribution to the sum of the loads of products i and j. Moreover, because the total sum of MNL preferences weights remains unchanged after a merge, the choice probabilities of all remaining products in the assortment S are also unchanged, and consequently, their load is similar to its premerge counterpart. Hence, merging can only increase the maximum load across all products.

3.2.2. Operation 2: Transfer. Consider an assortment S and let i and j be two products in this assortment with respective preference weights $v_i \ge v_j$. For any $\delta \in [0, v_j]$, the operation of δ-weight transfer from product j to product i consists of (1) replacing product i with a new product of preference weight $v_i + \delta$ and (2) replacing product j with a new product of preference weight $v_j - \delta$. Notably, we always transfer weight from a lighter product to a heavier product. In the next lemma, whose proof is provided in Online Appendix A.2, we show that the transfer operation cannot decrease the expected maximum load.

Lemma 3. *Consider an assortment $S \subseteq \mathcal{N}$ and let $\tilde{S}_\delta$ be the assortment resulting from a δ-weight transfer. Then, $\mathbb{E}(M(\tilde{S}_\delta)) \ge \mathbb{E}(M(S))$.*

In the proof of this result, we analytically study the function $\delta \longmapsto \mathbb{E}(M(\tilde{S}_\delta))$, showing that it is nondecreasing. In fact, when δ increases, the choice probability of product i increases, the choice probability of product j decreases, and all other choice probabilities remain unchanged. Therefore, at least intuitively, by increasing δ, we virtually transfer part of the load of product j to product i. Because product i is heavier than product j, it is more likely that product i has a higher load, and therefore, transferring part of the load of product j to product i can only help increase the expected maximum load.

3.2.3. Sensitivity of the Expected Maximum Load Function. In the following, we study the effect of small changes in the instance parameters (choice probabilities or preference weights), on the expected maximum load of a given assortment. The next lemma shows that, given a multinomial vector, where zero is the no-selection option and $1, \dots, m$ are the product options, slight changes in the choice probabilities translate to small changes in the expected maximum load.

Lemma 4. *Let $\mathbf{Y} = (Y_0, Y_1, \dots, Y_m)$ and $\mathbf{W} = (W_0, W_1, \dots, W_m)$ be multinomial vectors, with parameters $(T, p_0^Y, \dots, p_m^Y)$ and $(T, p_0^W, \dots, p_m^W)$, respectively. Then, when $p_i^W \ge (1-\epsilon)p_i^Y$ for all $i \in \{1, \dots, m\}$, we have $\mathbb{E}(\max_{i=1,\dots,m} W_i) \ge (1-\epsilon) \cdot \mathbb{E}(\max_{i=1,\dots,m} Y_i)$.*

To prove this result, we construct a coupling between the random variables $\mathbf{Y}$ and $\mathbf{W}$, where every customer that selects some option i in W_i also selects the same option in Y_i with probability at least $1-\epsilon$. Consequently, when i is a deterministic option, it is relatively straightforward to claim that the load of this option only suffers an ϵ-fraction loss (in expectation). However, in our case, the option i itself is a random variable, corresponding to the product that attains the maximum load. Based on an elegant conditioning argument, we show that a claim of similar spirit can be extended to the latter setting. The full proof of this lemma appears in Online Appendix A.4.

Similarly, in Lemma 5, we show that with respect to any assortment, small changes in the preference weights of its products translate to small changes in the expected maximum load. The proof of this result can be found in Online Appendix A.5.

Lemma 5. *Let $S^+ = \{1^+, \dots, m^+\}$ and $S^- = \{1^-, \dots, m^-\}$ be a pair of assortments, and let v_i^+ (respectively, v_i^-) be the preference weight of product i^+ (respectively, i^-)' for all $i \in [m]$. When $(1-\epsilon)v_i^+ \le v_i^- \le v_i^+$ for all $i \in [m]$, we have $\mathbb{E}(M(S^-)) \ge (1-\epsilon) \cdot \mathbb{E}(M(S^+))$.*

3.3. The 1/2-Approximation via Preference-Weight-Ordered Assortments

3.3.1. Main Result. Roughly speaking, preference weight–ordered assortments prioritize products with higher preference weight. Formally, assuming without loss of generality that $v_1 \geq \cdots \geq v_n$, we say that an assortment S is preference weight ordered when it forms a prefix of this sequence, that is, $S = \{1, 2, \ldots, j\}$ for some $1 \leq j \leq n$. In what follows, we show that there exists a weight-ordered assortment whose expected maximum load is within factor 2 of optimal. Because there are only n such assortments, and because we can compute the expected maximum load for each of these assortments by employing our evaluation oracle (see Section 3.1), the latter claim yields a polynomial time 1/2 approximation for static-MLA, as stated in the following theorem.

Theorem 2. *There is a preference-weight-ordered assortment that forms a* 1/2 *approximation to static-MLA. Moreover, we can compute such an assortment in polynomial time.*

In order to establish this result, the overall idea is to consider an optimal assortment S^* for static-MLA that may not necessarily be preference weight ordered. We will then sequentially modify S^* to obtain a weight-ordered assortment. We prove that, because of these modifications, the loss incurred does not exceed 1/2 of the objective function. In other words, letting S be the resulting preference weight–ordered assortment, we would claim that $\mathbb{E}(M(S)) \geq \mathbb{E}(M(S^*))/2$.

3.3.2. Outline of Analysis. To prove Theorem 2, using the Merge and Transfer operations presented in Section 3.2, we first show that any sufficiently heavy assortment can be replaced by a preference weight–ordered assortment plus a so-called virtual product (i.e., not present in the universe $\mathcal{N}$), without decreasing the expected maximum load. Moreover, we argue that the preference weight of the latter product is upper bounded by every preference weight in the weight-ordered assortment.

Lemma 6. *Let $S \subseteq \mathcal{N}$ be an assortment with $v(S) \geq v_1$. Then, there exists a weight-ordered assortment $\tilde{S} \subseteq \mathcal{N}$ and a virtual product k whose preference weight is at most $\min_{i \in \tilde{S}} v_i$ such that $\mathbb{E}(M(\tilde{S} \cup \{k\})) \geq \mathbb{E}(M(S))$.*

To arrive at Lemma 6, we begin with an assortment S and execute a sequence of Merge and Transfer operations to generate the assortment $\tilde{S}$ along with the virtual product k, both with the desired structure. The proof of this claim can be found in Online Appendix A.6.

Recalling that product k is not part of the universe $\mathcal{N}$, the next lemma shows that this virtual product can be removed while losing a factor of at most $1/(|S|+1)$ in the objective function.

Lemma 7. *Let $S \subseteq \mathcal{N}$ be any nonempty assortment and let $k \notin S$ be a product with $v_k \leq \min_{i \in S} v_i$. Then, $\mathbb{E}(M(S)) \geq \frac{|S|}{|S|+1} \cdot \mathbb{E}(M(S \cup \{k\}))$.*

In particular, when $S \neq \emptyset$, the lemma above shows that, by removing the virtual product k from $S \cup \{k\}$, we lose a factor of at most 1/2 in the objective function. The proof of this result is based on our structural results regarding on the sensitivity of the expected maximum load function, along the lines of Lemma 5. Its complete details are given in Online Appendix A.7.

3.3.3. Concluding the Proof of Theorem 2. Let S^* be an optimal assortment for static-MLA. First, we observe that $v(S^*) \geq v_1$. Indeed, suppose by contradiction that $v(S^*) < v_1$. In this case, on the one hand, when offering the assortment S^*, the total number of customers who select an option in S^* (i.e., do not select the no-purchase option) is a binomial random variable with T trials and success probability $v(S^*)/(1+v(S^*))$. Because the maximum load when offering S^* is trivially upper bounded by the total number of purchases, it follows that $\mathbb{E}(M(S^*)) \leq Tv(S^*)/(1+v(S^*))$. On the other hand, when offering the single-product assortment $\{1\}$, the expected maximum load is given by $\mathbb{E}(M(\{1\})) = Tv_1/(1+v_1)$. Because the function $x \mapsto x/(1+x)$ is increasing over $[0, +\infty)$, and because $v(S^*) < v_1$, we have $\mathbb{E}(M(S^*)) < \mathbb{E}(M(\{1\}))$, contradicting the optimality of S^*.

Now, given that $v(S^*) \geq v_1$, the conditions of Lemma 6 are met, and therefore, there exists a weight-ordered assortment $\tilde{S}$ and a virtual product k whose preference weight is at most $\min_{i \in \tilde{S}} v_i$, such that $\mathbb{E}(\tilde{S} \cup \{k\}) \geq \mathbb{E}(M(S^*))$. By definition, weight-ordered assortments are nonempty, meaning that according to Lemma 7, we have $\mathbb{E}(M(\tilde{S})) \geq \frac{1}{2} \cdot \mathbb{E}(M(\tilde{S} \cup \{k\}))$. Putting both inequalities together,

$$\mathbb{E}(M(\tilde{S})) \geq \frac{1}{2} \cdot \mathbb{E}(M(\tilde{S} \cup \{k\})) \geq \frac{1}{2} \cdot \mathbb{E}(M(S^*)).$$

3.4. PTAS

Our main technical contribution for static-MLA consists in designing a PTAS. In other words, for any fixed $\epsilon \in (0, 1)$, we propose a polynomial time algorithm for identifying an assortment whose expected maximum load is within factor $1 - \epsilon$ of optimal. This result is formalized in the next theorem.

Theorem 3. *For any $\epsilon \in (0, 1)$, static-MLA can be approximated within a factor of $1 - \epsilon$ of optimal. The running time of our algorithm is $O(T^{O(1)} \cdot n^{O(\frac{1}{\epsilon}\log\frac{1}{\epsilon})})$.*

3.4.1. Block-Based Assortments. In what follows, we introduce a family of highly structured assortments, which will be referred to as being block based. As explained below, these assortments are defined in three steps, starting from the block of products with the highest preference weight and gradually moving to blocks with lower weights. Without loss of generality, we assume that $1/\epsilon$ takes an integer value, and that products are indexed in nonincreasing order of preference weights, that is, $v_1 \geq \cdots \geq v_n$. With these conventions, an assortment $S \subseteq \mathcal{N}$ is said to be block based either if its cardinality is at most $1/\epsilon$, or when it can be written as $S = S_1 \cup S_2 \cup S_3$, where the latter sets are structured as follows:

- *Block 1:* The first set, S_1, is an arbitrary collection $1/\epsilon$ products. These products will form the subset of the heaviest products in this assortment.
- *Block 2:* Let a be the highest-index product in S_1. The second subset of products in our assortment, S_2, is a contiguous block of products, starting from $a+1$. In other words, $S_2 = \{a+1, a+2, \ldots, b\}$, for some $b \leq n$.
- *Further blocks:* Let $c = b+1$, where b is the highest-index product in S_2. We create a multiplicative grid across $[\epsilon \cdot v_c, v_c]$ as follows. The class C_1 consists of all products whose weight falls within $[(1-\epsilon)\cdot v_c, v_c]$. Then, the class C_2 consists of all products with weights in $[(1-\epsilon)^2 \cdot v_c, (1-\epsilon)\cdot v_c)$. We continue until we hit the lower bound $\epsilon \cdot v_c$. Letting $C_1, \ldots, C_L$ be the resulting classes, one can easily verify that the number of classes is $L = O(\frac{1}{\epsilon}\log\frac{1}{\epsilon})$. Now, for each class C_ℓ, we select a number N_ℓ of products to be included in the assortment and then simply include the N_ℓ products with largest indices from this class. For example, if a certain class is $\{8, 9, 10, 11\}$, then we include $\emptyset$ when $N_\ell = 0$, $\{11\}$ when $N_\ell = 1$, up to $\{11, 10, 9, 8\}$ when $N_\ell = 4$. We will refer to the union of these sets over all classes as S_3.

We proceed by explaining how to explicitly construct the entire family of blocked-based assortments in $O(n^{O(\frac{1}{\epsilon}\log(\frac{1}{\epsilon}))})$ time. First, there are $O(n^{O(1/\epsilon)})$ options to construct an assortment whose cardinality is strictly smaller than $1/\epsilon$. Let us now construct the assortments whose cardinality is at least $1/\epsilon$. In order to create the first block, S_1, it is easy to see that there are $O(n^{O(1/\epsilon)})$ options. Because the second block S_2 is contiguous, there are at most $O(n)$ options here. For the remaining blocks, we create L classes, and for each of these classes, we simply choose the number of products N_ℓ to be included. Therefore, there are $O(n^L) = O(n^{O(\frac{1}{\epsilon}\log\frac{1}{\epsilon})})$ options to construct S_3.

3.4.2. Performance of Block-Based Assortments.

Our algorithmic approach consists of enumerating over all block-based assortments. Because the evaluation oracle provided in Section 3.1 can be implemented in $O(n^2T^3)$ time, our overall algorithm indeed has a running time of $O(T^{O(1)} \cdot n^{O(\frac{1}{\epsilon}\log\frac{1}{\epsilon})})$, as stated in Theorem 3. The next result proves that at least one of the assortments we are enumerating over yields a $(1-\epsilon)$ approximation of static-MLA.

Theorem 4. *Letting S^* be an optimal assortment for static-MLA, there exists a block-based assortment S for which $\mathbb{E}(M(S)) \geq (1-\epsilon)\cdot \mathbb{E}(M(S^*))$.*

3.4.3. Definitions and Notation.

In order to establish this theorem, we begin by introducing a number of useful definitions and their surrounding notation:

- For any assortment S, and for any $j = 1, \ldots, |S|$, we define $I_j(S)$ as the index of the jth heaviest product in S. That is, if $S = \{p_1, p_2, \ldots, p_{|S|}\}$ where $v_{p_1} \geq \cdots \geq v_{p_{|S|}}$, then $I_j(S) = p_j$.
- For any assortment S with $|S| > 1/\epsilon$, we define its ϵ*-hole* as the product with index $h_\epsilon(S)$ where $h_\epsilon(S) = \min\{j \notin S : j \geq I_{1/\epsilon}(S)\}$. Put simply, the ϵ-hole refers to the heaviest product in the universe $\mathcal{N}$ that is not part of the assortment S, but is lighter than the $1/\epsilon$ heaviest product of S.
- Finally, we say that an assortment S is ϵ*-restricted* either when $|S| \leq 1/\epsilon$, or when $|S| > 1/\epsilon$ and the weight of each product in S is larger than a fraction ϵ of the weight of its ϵ-hole, that is, $v_i \geq \epsilon \cdot v_{h_\epsilon(S)}$ for every $i \in S$. It is worth noting that, by definition, all block-based assortments are ϵ-restricted.

3.4.4. Analysis.

Given these definitions, the proof of Theorem 4 consists of two steps. In Lemma 8, we prove that for any sufficiently large assortment S, there exists an ϵ-restricted assortment $\hat{S}$ and a virtual product k, such that the expected maximum load of $\hat{S} \cup \{k\}$ is at least as large as that of S. The proof of this lemma makes use of the Merge and Transfer operations introduced in Section 3.2, transforming the assortment S into the union of an ϵ-restricted assortment and a virtual product. The detailed proof is included in Online Appendix A.8.

Lemma 8. *Let $S \subseteq \mathcal{N}$ be an assortment with $|S| > 1/\epsilon$. Then, there exists an ϵ-restricted assortment $\hat{S} \subseteq \mathcal{N}$ with $|\hat{S}| > 1/\epsilon$, and a virtual product k with $v_k \leq v_{h_\epsilon(\hat{S})}$, such that $\mathbb{E}(M(\hat{S} \cup \{k\})) \geq \mathbb{E}(M(S))$.*

In the following lemma, whose proof is presented in Online Appendix A.9, we show that the assortment $\hat{S} \cup \{k\}$ obtained in Lemma 8 can be transformed into a block-based assortment, losing at most a factor ϵ in its objective value.

Lemma 9. *Let $\hat{S} \subseteq \mathcal{N}$ be an ϵ-restricted assortment with $|\hat{S}| > 1/\epsilon$ and let k be a virtual product with $v_k \leq v_{h_\epsilon(\hat{S})}$. Then, there exists a block-based assortment $\tilde{S} \subseteq \mathcal{N}$ such that $\mathbb{E}(M(\tilde{S})) \geq (1-\epsilon)\cdot \mathbb{E}(M(\hat{S} \cup \{k\}))$.*

To conclude the proof of Theorem 4, let S^* be an optimal assortment for static-MLA. When $|S^*| \leq 1/\epsilon$, we know that S^* is a block-based assortment by definition, and it remains to consider the opposite case, where $|S^*| > 1/\epsilon$. By Lemma 8, there exists an ϵ-restricted assortment $\hat{S}$ with $|\hat{S}| > 1/\epsilon$ and a virtual product k with $v_k \leq v_{h_\epsilon(\hat{S})}$ such that $\mathbb{E}(M(\hat{S} \cup \{k\})) \geq \mathbb{E}(M(S^*))$. In turn, because $\hat{S}$ and k satisfy the conditions of Lemma 9, there exists a block-based assortment $\tilde{S}$ such that $\mathbb{E}(M(\tilde{S})) \geq (1-\epsilon) \cdot \mathbb{E}(M(\hat{S} \cup \{k\}))$. Combining these two inequalities yields $\mathbb{E}(M(\tilde{S})) \geq (1-\epsilon) \cdot \mathbb{E}(M(S^*))$, as desired.

3.5. Many-Customers Regime: Optimal Assortment

In this section, we show that when the number of customers T is sufficiently large, offering the single product with the highest preference weight is optimal for static-MLA. As mentioned in Section 1.4, optimal assortments balance a tradeoff between offering a large or a small number of products. In particular, offering more products decreases the choice probability of the no-purchase option, thereby capturing more customers. However, this decision comes at the cost of cannibalizing the demand between products due to the underlying substitution effect in the MNL model. That is, when we offer more products, the demand will not be concentrated in a single product as desired in the maximum load assortment problem. For large values of T, the effect of cannibalization is accentuated, since the load of each product is more and more concentrated around its mean, by virtue of Chernoff-type bounds. Consequently, lighter products' loads are highly unlikely to surpass those of heavier products, and offering them only contributes to cannibalizing the demand. This intuition suggests that large values of T favor smaller-sized assortments. In the following lemma, whose proof is presented in Online Appendix A.10, we show that there exists a threshold $\overline{T}$, depending on the problem parameters, above which offering only the product with the highest preference weight is optimal. Recall that products are assumed to be indexed in weakly decreasing order of their preference weights, meaning that product 1 is the heaviest in $\mathcal{N}$.

Lemma 10. *There exists a threshold $\overline{T}$ such that $\mathbb{E}(M(\{1\})) = \max_{S \subseteq \mathcal{N}} \mathbb{E}(M(S))$ when $T \geq \overline{T}$.*

4. Dynamic Setting: Constant Factor Adaptivity Gaps

In this section, we examine how well an optimal static assortment could perform in comparison with an optimal adaptive policy. Specifically, we study the adaptivity gap of maximum load optimization, namely the worst-possible ratio between the expected maximum load of an optimal adaptive policy and that of an optimal static policy over all problem instances. Specifically, any instance I is characterized by its number of customers T, number of products n, and their preference weights. Letting $\mathcal{I}$ be the set of all possible instances, the adaptivity gap is formally defined as

$$\max_{I \in \mathcal{I}} \frac{\mathsf{OPT}_I^{\mathrm{DP}}}{\mathsf{OPT}_I^{\mathrm{Static}}},$$

where $\mathsf{OPT}_I^{\mathrm{Static}}$ and $\mathsf{OPT}_I^{\mathrm{DP}}$, respectively, denote the expected maximum load of an optimal static policy and an optimal dynamic policy for the instance I. Quite surprisingly, we establish an adaptivity gap of at most four, showing that statically offering a weight-ordered assortment guarantees a $1/4$ approximation to dynamic-MLA. Moreover, we show that this gap reduces to at most two when all products have identical preference weights.

In Section 4.1, we provide some useful notation and describe our main adaptivity gap results in greater detail. In Section 4.2, we present several auxiliary claims that will be helpful in the subsequent analysis. Then, we prove an adaptivity gap of at most four for general instances in Section 4.3, deferring the improved finding for the identical weight setting to Online Appendix B.5. Finally, we construct a dynamic-MLA instance, demonstrating that the adaptivity gap of this problem is at least $4/3$.

4.1. Notation and Main Results

4.1.1. Notation. Let us start by introducing some helpful notation and definitions. First, in the remainder of this section, we fix a single instance, consisting of n products represented by the universe $\mathcal{N}$, their preference weights, and the number of customers T. In what follows, we generalize the notion of preference weight–ordered assortments to any universe of products $U \subseteq \mathcal{N}$, still prioritizing those with higher preference weights. Formally, suppose that $U = \{i_1, \ldots, i_k\} \subseteq \mathcal{N}$ and that $v_{i_1} \geq \cdots \geq v_{i_k}$. We say that the assortment $S \subseteq U$ is preference weight ordered in U when $S = \{i_1, \ldots, i_m\}$ for some $1 \leq m \leq k$. With this definition, let $\mathsf{OPT}^{\mathrm{WO}}(U)$ be the optimal expected maximum load achievable by a static preference weight–ordered assortment in U. In other words,

$$\mathsf{OPT}^{\mathrm{WO}}(U) = \max_{m=1,\ldots,k} \mathbb{E}(M(\{i_1, \ldots, i_m\})).$$

In addition, we define $\mathsf{OPT}^{\mathrm{DP}}(U)$ as the expected maximum load of an optimal dynamic policy, using only products in U.

4.1.2. Main Results. Quite surprisingly, we show that by statically offering a weight-ordered assortment, one can attain an expected maximum load of at least $1/4$ of the optimal expected maximum load of dynamic-MLA. The proof of this result appears in Section 4.3.

Theorem 5. *There exists a static weight-ordered assortment that provides a $1/4$ approximation to dynamic-MLA, that is, $\mathsf{OPT}^{\mathrm{WO}}(\mathcal{N}) \geq \frac{1}{4} \cdot \mathsf{OPT}^{\mathrm{DP}}(\mathcal{N})$.*

It is worth noting that $\mathsf{OPT}^{\mathrm{DP}}(\mathcal{N})$ represents the expected maximum load of an optimal dynamic policy for dynamic-MLA, where all products in $\mathcal{N}$ are considered. Additionally, $\mathsf{OPT}^{\mathrm{WO}}(\mathcal{N})$ denotes the expected maximum load of an optimal weight-ordered static assortment in $\mathcal{N}$, which is clearly upper bounded by the expected maximum load of an optimal assortment for static-MLA. Therefore, the adaptivity gap of this setting is at most four. Moreover, as explained in Section 3.1, we can compute $\mathsf{OPT}^{\mathrm{WO}}(\mathcal{N})$ in polynomial time, meaning that the above theorem yields a $1/4$ approximation for dynamic-MLA.

When all products are associated with identical preference weights, we derive an improved adaptivity gap of two, as stated in the next theorem, whose proof is provided in Online Appendix B.5.

Theorem 6. *Suppose that all products have identical preference weights. Then, there exists a static weight-ordered assortment that provides a* $1/2$ *approximation to* dynamic-MLA*, that is,* $\mathsf{OPT}^{\mathrm{WO}}(\mathcal{N}) \geq \frac{1}{2} \cdot \mathsf{OPT}^{\mathrm{DP}}(\mathcal{N})$.

4.2. Auxiliary Claims

4.2.1. Upper Bounding $\mathsf{OPT}^{\mathrm{DP}}(U)$.

For a fixed universe of products $U \subseteq \mathcal{N}$, recall that $\mathsf{OPT}^{\mathrm{DP}}(U)$ represents the expected maximum load attained by an optimal dynamic policy with respect to the universe U. In Lemma 11, whose proof is given in Online Appendix B.1, we provide an upper bound on $\mathsf{OPT}^{\mathrm{DP}}(U)$ that will serve as an initial step toward bounding the expected maximum load of an optimal dynamic policy. Specifically, we consider a random multinomial vector $(L_1, \ldots, L_k)$, and establish a condition on its vector of probabilities and on the preference weights of products in U, ensuring that the expected maximal component of this multinomial vector exceeds $\mathsf{OPT}^{\mathrm{DP}}(U)$.

Lemma 11. *Let* $(L_1, \ldots, L_k)$ *be a random multinomial vector with* T *trials and probability vector* $(p_1, \ldots, p_k)$. *Let* $U \subseteq \mathcal{N}$ *be a set of products with* $\min_{i=1,\ldots,k} p_i \geq \max_{i \in U} \frac{v_i}{1+v_i}$. *Then,* $\mathbb{E}(\max(L_1, \ldots, L_k)) \geq \mathsf{OPT}^{\mathrm{DP}}(U)$.

To interpret the condition stated above, consider an optimal dynamic policy for the maximum load assortment problem with respect to the universe U. Whenever this policy offers an assortment S to some customer $t \in [T]$, the MNL choice probability of each product $i \in S$ is $\frac{v_i}{1+v(S)} \leq \frac{v_i}{1+v_i}$. Therefore, $\min_{i=1,\ldots,k} p_i \geq \max_{i \in U} \frac{v_i}{1+v_i}$ can be viewed as a condition where, regardless of the offered assortment, the MNL choice probabilities of all products in U are upper bounded by $\min_{i=1,\ldots,k} p_i$. Under this condition, we prove that the expected maximal component of $(L_1, \ldots, L_k)$ is an upper bound on the expected maximum load of an optimal dynamic policy.

4.2.2. Consequences of Offering Larger Subsets

Consider an adaptive policy A for the maximum load assortment problem with respect to the universe U. We denote by $\mathcal{E}^A(U)$ the expected maximum load achieved by this policy. Additionally, for each $t = 1, \ldots, T$, we make use of S_t^A to designate the subset of U offered by A to customer t. This subset is clearly random because it depends on the random selections made by previously arriving customers.

Now, consider two adaptive policies, A and B, such that for any $t \in [T]$, the assortment offered by policy A to customer t is almost surely a subset of the one offered by policy B to this customer. However, we assume that the difference in total preference weight between these two assortments is almost surely upper bounded by some $\epsilon \geq 0$. The next lemma, whose proof is included in Onlien Appendix B.3, gives a lower bound on the ratio between the expected maximum loads of the two policies as a function of ϵ.

Lemma 12. *Let* A *and* B *be two adaptive policies with respect to the universe* U. *For every* $t \in [T]$, *suppose that* $S_t^A \subseteq S_t^B$ *and* $v(S_t^B \setminus S_t^A) \leq \epsilon$ *almost surely. Then,* $\mathcal{E}^B(U) \geq \frac{1}{1+\epsilon} \cdot \mathcal{E}^A(U)$.

4.2.3. Subadditivity.

Finally, we prove that $\mathsf{OPT}^{\mathrm{DP}}(\cdot)$ is a subadditive function, as formally stated below.

Lemma 13. *For any* $U_1, U_2 \subseteq \mathcal{N}$, *we have* $\mathsf{OPT}^{\mathrm{DP}}(U_1 \cup U_2) \leq \mathsf{OPT}^{\mathrm{DP}}(U_1) + \mathsf{OPT}^{\mathrm{DP}}(U_2)$.

The proof of this result relies on a coupling argument involving three dynamic policies: (1) a policy that offers only products from the universe U_1; (2) a policy offering only products from U_2; and (3) an optimal policy that offers products from the combined universe, $U_1 \cup U_2$. Within this coupling, we demonstrate that for at least one of the policies (1) and (2), its maximum load is almost surely at least as large as that of policy (3) with respect to the constructed coupling. The complete proof is provided in Online Appendix B.4.

4.3. Proof of Theorem 5

4.3.1. Easy Regime: $v(\mathcal{N}) < 1$.

In this case, recalling that $v_1 \geq \cdots \geq v_n$, we simply make use of the static policy A where all products in $\mathcal{N}$ are offered to every customer $t \in [T]$. We will show that the expected maximum load of this policy is at least $\mathsf{OPT}^{\mathrm{DP}}(\mathcal{N})/2$. To this end, focusing on an optimal dynamic policy A^*, let $S_t^{A^*}$ be the random assortment it offers to customer t. We have $S_t^{A^*} \subseteq \mathcal{N}$ and $v(\mathcal{N} \setminus S_t^{A^*}) \leq 1$, because $v(\mathcal{N}) < 1$ by the case hypothesis. Therefore, Lemma 12 implies that the expected maximum load $\mathcal{E}^A(\mathcal{N})$ of our static policy is at least $\mathcal{E}^{A^*}(\mathcal{N})/2$. In other words, $\mathbb{E}(M(\mathcal{N})) \geq \mathsf{OPT}^{\mathrm{DP}}(\mathcal{N})/2$. Clearly, $\mathcal{N}$ is also preference weight ordered, meaning that $\mathsf{OPT}^{\mathrm{WO}}(\mathcal{N}) \geq \mathsf{OPT}^{\mathrm{DP}}(\mathcal{N})/2$.

4.3.2. Overview of the Difficult Regime: $v(\mathcal{N}) \geq 1$.

Let k be the minimal integer for which $\sum_{i=1}^{k} v_i \geq 1$ and consider the assortment $U = \{1, \ldots, k\}$. In the following, we argue that by statically offering this assortment to all customers, the expected maximum load is at least $\mathsf{OPT}^{\mathrm{DP}}(\mathcal{N})/4$. In other words, $\mathbb{E}(M(U)) \geq \mathsf{OPT}^{\mathrm{DP}}(\mathcal{N})/4$. Because U is weight ordered, the latter bound would imply that $\mathsf{OPT}^{\mathrm{WO}}(\mathcal{N}) \geq \mathsf{OPT}^{\mathrm{DP}}(\mathcal{N})/4$.

For this purpose, by Lemma 13, we know that $\mathsf{OPT}^{\mathrm{DP}}(\cdot)$ is a subadditive function, meaning in particular that

$$\mathsf{OPT}^{\mathrm{DP}}(\mathcal{N}) \leq \mathsf{OPT}^{\mathrm{DP}}(U) + \mathsf{OPT}^{\mathrm{DP}}(\mathcal{N} \setminus U). \tag{4}$$

Now, let $(\hat{L}_1, \ldots, \hat{L}_k)$ be a multinomial vector with T trials and probability vector $(p_1, \ldots, p_k)$, where $p_i = \frac{v_i}{v(U)}$ for every $i = 1, \ldots, k$. In the next two lemmas, whose proofs appear following the current one, we show that both $\mathsf{OPT}^{\mathrm{DP}}(U)$ and $\mathsf{OPT}^{\mathrm{DP}}(\mathcal{N} \setminus U)$ are upper bounded by $\mathbb{E}(\hat{M})$, where $\hat{M} = \max_{i=1,\ldots,k} \hat{L}_i$.

Lemma 14. $\mathsf{OPT}^{\mathrm{DP}}(U) \leq \mathbb{E}(\hat{M})$.

Lemma 15. $\mathsf{OPT}^{\mathrm{DP}}(\mathcal{N} \setminus U) \leq \mathbb{E}(\hat{M})$.

On the other hand, we argue that the expected maximum load achieved by the static assortment U is at least $\mathbb{E}(\hat{M})/2$.

Lemma 16. $\mathbb{E}(M(U)) \geq \mathbb{E}(\hat{M})/2$.

We are now ready to complete the proof of Theorem 5. To this end, noting that the assortment U is preference-weight-ordered, we know by Lemma 16 that $\mathsf{OPT}^{\mathrm{WO}}(\mathcal{N}) \geq \mathbb{E}(\hat{M})/2$. Consequently,

$$\begin{aligned}\mathsf{OPT}^{\mathrm{WO}}(\mathcal{N}) &\geq \frac{1}{2} \cdot \mathbb{E}(\hat{M}) \\ &\geq \frac{1}{4} \cdot (\mathsf{OPT}^{\mathrm{DP}}(U) + \mathsf{OPT}^{\mathrm{DP}}(\mathcal{N} \setminus U)) \\ &\geq \frac{1}{4} \cdot \mathsf{OPT}^{\mathrm{DP}}(\mathcal{N}),\end{aligned}$$

where the second inequality follows from Lemmas 14 and 15, and the third inequality is precisely (4).

Proof of Lemma 14. Let $(\tilde{L}_1, \ldots, \tilde{L}_k)$ be the random loads of the products when employing an optimal adaptive policy for the universe U and let $\tilde{M} = \max_{i \in U} \tilde{L}_i$. By definition, $\mathsf{OPT}^{\mathrm{DP}}(U) = \mathbb{E}(\tilde{M})$, and our objective is to prove that $\mathbb{E}(\tilde{M}) \leq \mathbb{E}(\hat{M})$.

For this purpose, we begin by observing that, for every $i \in [k]$,

$$\frac{v_i}{v(U)} > \frac{v_i}{1 + v_k} \geq \frac{v_i}{1 + v_i}, \tag{5}$$

where the first and second inequalities hold, respectively, because $\sum_{j=1}^{k-1} v_j < 1$ by definition of k and because $v_1 \geq \cdots \geq v_k$. Looking into Equation (5), its left-hand side is exactly the probability of component i in the process of generating the random multinomial vector $(\hat{L}_1, \ldots, \hat{L}_k)$. The right-hand side is the choice probability of product i when the single-product assortment $\{i\}$ is offered, which is an upper bound on the choice probability of product i at any step of generating $(\tilde{L}_1, \ldots, \tilde{L}_k)$. In other words, letting $S_t^{A^*} \subseteq U$ be the random assortment offered by the optimal dynamic policy A^* to customer $t \in [T]$, for every product $i \in [k]$, we have

$$\phi_i(S_t^{A^*}) \leq \frac{v_i}{1 + v_i} \leq \frac{v_i}{v(U)}. \tag{6}$$

This observation implies the existence of a simple way to couple $(\tilde{L}_1, \ldots, \tilde{L}_k)$ and $(\hat{L}_1, \ldots, \hat{L}_k)$ such that $\tilde{L}_i \leq \hat{L}_i$, for every product $i \in [k]$. As a result, $\mathbb{E}(\tilde{M}) \leq \mathbb{E}(\hat{M})$.

Proof of Lemma 15. The key idea is to notice that $\frac{v_i}{v(U)} \geq \frac{v_i}{1+v_i} \geq \frac{v_j}{1+v_j}$, for every pair of products $1 \leq i \leq k$ and $k+1 \leq j \leq n$. Here, the first inequality is precisely Equation (5), and the second inequality holds because $v_1 \geq \cdots \geq v_n$. Therefore, $\min_{i=1,\ldots,k} \frac{v_i}{v(U)} \geq \max_{j=k+1,\ldots,n} \frac{v_j}{1+v_j}$. Recall that $v_i/v(U)$ is the probability of picking option i in the multinomial vector $(\hat{L}_1, \ldots, \hat{L}_k)$. Therefore, by applying Lemma 11, we have $\mathbb{E}(\max(\hat{L}_1, \ldots, \hat{L}_k)) \geq \mathsf{OPT}^{\mathrm{DP}}(\mathcal{N} \setminus U)$. Finally, $\mathbb{E}(\hat{M}) \geq \mathsf{OPT}^{\mathrm{DP}}(\mathcal{N} \setminus U)$, by definition of $\hat{M}$.

Proof of Lemma 16. Consider the static policy where we offer the weight-ordered assortment U to every customer and let $(L_0, L_1, \ldots, L_k)$ be the load vector associated with this policy. The latter vector follows a multinomial distribution, where the choice probability of each product $i \in U$ is $\frac{v_i}{1+v(U)}$ and the no-purchase option has probability $\frac{1}{1+v(U)}$. On the other hand, consider the random vector $(\hat{L}_1, \ldots, \hat{L}_k)$, recalling that it has been defined as being multinomial with T trials and probabilities $(p_1, \ldots, p_k)$, where $p_i = \frac{v_i}{v(U)}$ for every $i = 1, \ldots, k$. We complement this vector with $\hat{L}_0$ that has a probability of zero.

We proceed by applying Lemma 4 to the multinomial vectors $(L_0, L_1, \ldots, L_k)$ and $(\hat{L}_0, \hat{L}_1, \ldots, \hat{L}_k)$. Specifically, because $v(U) \geq 1$, we have $\frac{v_i}{1+v(U)} \geq \frac{1}{2} \cdot \frac{v_i}{v(U)}$, implying that the conditions of this lemma are met with $\epsilon = 1/2$. It follows that $\mathbb{E}(M(U)) \geq \mathbb{E}(\hat{M})/2$.

4.4. Lower Bound on the Adaptivity Gap

Whereas Theorem 5 attains an upper bound of four on the adaptivity gap, we proceed to consider the opposite direction and provide a lower bound on this measure. In particular, we construct an instance of the maximum load assortment problem, demonstrating that the adaptivity gap of this setting is at least 4/3. The upcoming construction is motivated in Online Appendix C, where we present a numerical analysis of the adaptivity gap with respect to various problem parameters in order to identify a regime where the largest adaptivity gaps are attained. In particular, we observe that high adaptivity gaps are attained when $T = 2$ with uniform preference weights, which leads to our focus on such instances in the construction below.

Lemma 17. *The adaptivity gap of* dynamic-MLA *is at least* $4/3$.

4.4.1. Instance. In what follows, we consider an instance defined over a universe of n products, each with a preference weight of one. In addition, the number of customers is $T = 2$. Let us compare the optimal adaptive policy against the performance of an optimal static assortment.

4.4.2. Optimal Dynamic Policy. Because products have identical preference weights, it is easy to verify that the optimal dynamic policy starts by offering the whole universe of products to the first customer. With probability $n/(1+n)$, the first customer selects some product, say i. In this event, the optimal policy will offer the assortment $\{i\}$ to the second customer, as adding any other product can only cannibalize product i. In the complementary event, the first customer selects the no-purchase option, and in this case, the optimal policy offers the whole universe of products to the second customer. Therefore, by conditioning on the choice of the first customer, the expected maximum load of the optimal dynamic policy is given by

$$\begin{aligned} \mathsf{OPT}^{\mathrm{DP}}(\mathcal{N}) &= \frac{n}{1+n}\cdot\left(1+\frac{1}{2}\right)+\frac{1}{1+n}\cdot\frac{n}{1+n} \\ &= \frac{3}{2}\cdot\left(1-\frac{1}{n+1}\right)+\frac{n}{(1+n)^2}. \end{aligned} \tag{7}$$

4.4.3. Optimal Static Assortment. For every $k = 1,\dots,n$, we compute the expected maximum load achieved by statically offering an assortment S_k consisting of k products. For this assortment, the maximum load is two if and only if the same product is selected by both customers, which happens with probability $\frac{k}{(1+k)^2}$. Similarly, the maximum load is zero if and only if the no-purchase option is selected by both customers, which happens with probability $\frac{1}{(1+k)^2}$. It follows that the maximum load is one with probability $\frac{k^2+k}{(1+k)^2}$, and a simple calculation shows that

$$\mathbb{E}(M(S_k)) = \frac{k^2+3k}{(1+k)^2}. \tag{8}$$

To bound the latter expectation, elementary calculus arguments show that the function $x \longmapsto \frac{x^2+3x}{(1+x)^2}$ attains its maximum value over $[0,\infty)$ at $x = 3$, and therefore $\max_{k\in[n]}\mathbb{E}(M(S_k)) \leq 9/8$.

4.4.4. Lower Bound on the Adaptivity Gap. By combining Equations (7) and (8), we obtain an adaptivity gap of at least

$$\begin{aligned} &\lim_{n\to\infty}\frac{\mathsf{OPT}^{\mathrm{DP}}(\mathcal{N})}{\max_{k\in[n]}\mathbb{E}(M(S_k))} \\ &\geq \frac{8}{9}\cdot\lim_{n\to\infty}\left(\frac{3}{2}\cdot\left(1-\frac{1}{n+1}\right)+\frac{n}{(1+n)^2}\right)=\frac{4}{3}. \end{aligned}$$

As a side note, because of considering the case of identical preference weights, by Theorem 6, we know that the adaptivity gap in this case is upper bounded by two. It is important to note that the choice of the instance presented above is not arbitrary. Rather, it results from numerically optimizing the number of customer T and their (uniform) preference weights to obtain the highest lower bound on the adaptivity gap.

4.5. Numerical Insights on the Adaptivity Gap

Beyond the theoretical results presented in Sections 4.1 and 4.4, several key numerical insights can be drawn from the numerical analysis we conduct in Online Appendix C. These insights offer a clearer understanding of the scenarios in which dynamic policies outperform static ones, highlighting the potential advantages of using more complex adaptive strategies compared with static approaches.

The first insight, observable in Table 1 in Online Appendix C, suggests that an increase in the customer base diminishes the advantage of employing a dynamic policy over a static one. This trend is reflected in the reduction of the adaptivity gap as the value of T grows. To better grasp the underlying reasons, it is essential to identify which problem instances effectively leverage adaptiveness and which do not. Specifically, in cases with a large number of customers, the flexibility to utilize adaptive policies is limited. These instances often require early commitment to specific products or subsets of products, making the different sample paths of the dynamic process resemble a static policy.

Secondly, when examining instances with a smaller customer base, Table 2 in Online Appendix C reveals that a larger adaptivity gap is observed in cases with a greater number of products. The underlying intuition here is that expanding the product set $\mathcal{N}$ enhances the flexibility of a dynamic policy, allowing it to initially offer larger assortments and thereby increasing the likelihood of capturing customer demand before making a final commitment. In contrast, a static policy is constrained by its commitment to a fixed assortment from the outset. Expanding the product universe may have little to no impact on the actual assortment offered under a static policy.

5. Dynamic Setting: Quasi-Polynomial $(1-\epsilon)$-Approximate Policy

In this section, we shift our focus toward designing a truly near-optimal policy for dynamic-MLA. Specifically, for any $\epsilon > 0$, we propose a $(1-\epsilon)$-approximate adaptive policy, admitting a quasi-polynomial time implementation. Our approach involves exploring a carefully selected class of policies with distinct properties, allowing one to dramatically reduce the search space of seemingly intractable dynamic programming ideas. Formally, we say that an algorithm terminates in quasi-polynomial time when, for every instance I, its running time is $O(|I|^{\mathrm{polylog}|I|})$, where $|I|$ stands for the input size in its binary representation. We start by presenting our main result in Section 5.1, stating the existence of a near-optimal dynamic policy that can be implemented in quasi-polynomial time. Subsequently, Section 5.2 will provide a number of auxiliary lemmas and observations. In Section 5.3, we present the specifics of our algorithmic approach by formally constructing a near-optimal dynamic policy.

5.1. Main Result

5.1.1. Main Result. As previously mentioned, our primary result consists of a quasi-PTAS (QPTAS) for dynamic-MLA. The next theorem describes this finding in greater detail, noting that $O_\epsilon(\cdot)$ simply hides polynomial dependencies in $1/\epsilon$.

Theorem 7. *For any $\epsilon > 0$, we can compute an adaptive policy for dynamic-MLA whose expected maximum load is within factor $1-\epsilon$ of optimal. This policy can be implemented in $O(n^{O_\epsilon(\log^3 n)})$.*

It is worth mentioning that the term "implementation" in this specific context encompasses two important aspects. Firstly, it includes any preprocessing steps undertaken prior to the beginning of the customer arrival process. Secondly, it contains the additional procedures required to compute a personalized assortment that will be offered to each arriving customer. As stated above, the running time of our algorithm is $O(n^{O_\epsilon(\log^3 n)})$, meaning that it indeed qualifies as a QPTAS. This running time is not as efficient as a PTAS, in which the exponent would have been dependent only on $1/\epsilon$.

5.1.2. Technical Overview. The fundamental challenge in addressing the dynamic setting arises from the exponential size of its dynamic programming state space (see Section 2.4). Thus, our initial focus lies in modifying the original instance, with the intent of arriving at a dramatically scaled-down state space. This alteration involves transforming the original universe of products $\mathcal{N}$ into a modified universe, where product weights are slightly altered. Additionally, we confine our exploration to a specific class of policies that truncates the arrival sequence once a predetermined threshold on the maximum load is reached, incurring at most an ϵ-fraction loss in the objective function. Although the idea of altering the space of products $\mathcal{N}$ helps mitigate the search space issue, it introduces a new source of complexity due to the dissimilarity between the products in the new universe and our initial universe. Consequently, our second step consists of recovering a policy with respect to the original universe, while essentially preserving the expected maximum load.

5.2. Useful Claims

In this section, we introduce several auxiliary claims that will be helpful in designing our near-optimal policy and in its analysis. For convenience, we assume without loss of generality that $T \geq 2$ and $n \geq 2$. Indeed, when $T = 1$, it is optimal to offer the whole universe of products. Similarly, the setting of $n = 1$ corresponds to having a single product, in which case it is optimal to offer this product to every customer.

5.2.1. Two Parametric Regimes. Let us start by explaining how any given instance can be classified into two possible regimes, referred to as high weight and low-weight. For this purpose, let $\alpha = v_{\max}/(1+v_{\max})$, which is precisely the choice probability of the heaviest product by a single customer, when it is the only one offered. Consequently, $T\alpha$ represents the expected load of this product when it is being statically offered to all customers. Then, the high-weight regime captures problem instances where $T\alpha \geq 12\ln(nT)/\epsilon^3$ or $v_{\max} \geq 1/\epsilon$, in which case we will show that a very simple static policy achieves a $(1-\epsilon)$ approximation. As explained later, the core difficulty lies within the low-weight regime, where $T\alpha < 12\ln(nT)/\epsilon^3$ and $v_{\max} < 1/\epsilon$, which will be the main focus of this section.

5.2.2. High-Weight Regime: $T\alpha \geq 12\ln(nT)/\epsilon^3$ or $v_{\max} \geq 1/\epsilon$. In this case, we prove that statically offering the heaviest product to all customers provides a $(1-\epsilon)$ approximation to dynamic-MLA, as formally stated below. The proof of this result appears in Online Appendix D.1.

Lemma 18. *When $T\alpha \geq \frac{12\ln(nT)}{\epsilon^3}$ or $v_{\max} \geq 1/\epsilon$, the static policy that offers the heaviest product guarantees a $(1-\epsilon)$ approximation for dynamic-MLA.*

Roughly speaking, the high-weight regime allows us to efficiently employ concentration bounds. These bounds demonstrate that, for any given policy, the event where its (random) maximum load exceeds $T\alpha$ by a nonnegligible factor is highly improbable. Consequently, with the right choice of parameters, we will show that the optimal expected maximum load is upper bounded by $(1+\epsilon)\cdot T\alpha$. As such, Lemma 18 will follow by recalling that $T\alpha$ represents the expected maximum load of statically offering the heaviest product to all customers.

5.2.3. Low-Weight Regime: $T\alpha < 12\ln(nT)/\epsilon^3$ and $v_{\max} < 1/\epsilon$. In this case, we establish a polylogarithmic upper bound on the optimal expected maximum load, which will be utilized in Section 5.3 to prove the near-optimality of a specific class of policies. Intuitively, under the low weight regime, products are not associated with high enough weights to prompt frequent selections of the same product. By formalizing this intuition, we show that within the low weight regime, the expected maximum load is polylogarithmically bounded. The proof of this result is included in Online Appendix D.2.

Lemma 19. *When* $T\alpha < \frac{12\ln(nT)}{\epsilon^3}$ *and* $v_{\max} < 1/\epsilon$, *we have* $\mathsf{OPT}^{\mathrm{DP}}(\mathcal{N}) \le \frac{300\ln^2(nT)}{\epsilon^6}$.

5.3. Constructing Our Policy

According to Lemma 18, statically offering the heaviest product to all customers achieves a $(1-\epsilon)$ approximation of the optimum under the high-weight regime. Therefore, in the remainder of this section, we focus on the low-weight regime. For convenience, let $\mathcal{B} = \frac{300\ln^2(nT)}{\epsilon^6}$ be the upper bound we obtained in Lemma 19 on the expected maximum load $\mathsf{OPT}^{\mathrm{DP}}(\mathcal{N})$ of an optimal dynamic policy. The primary idea behind our adaptive policy is to only explore policies that

- Stop offering products as soon as the maximum load reaches a value of $\mathcal{B}/\epsilon$, assuming without loss of generality that the latter term is an integer. Namely, the empty assortment is offered to all remaining customers once we hit this threshold.
- Avoid offering products of "tiny" preference weight, upper-bounded by $\epsilon^2 v_{\max}/n$.

We refer to such adaptive policies as *truncated* policies. The motivation behind this restriction is that it allows us to considerably shrink the search space, and in particular, to compute a near-optimal policy in quasi-polynomial time. An important question that remains to be answered is obviously centered around the performance guarantee of such policies. Our analysis will argue that, based on truncated policies, we indeed construct a near-optimal dynamic policy.

Step 1: Dropping light products. We start by considering a new universe of products U, where we drop all products whose preference weight is at most $\epsilon^2 \cdot v_{\max}/n$, that is, $U = \{i \in \mathcal{N} \,|\, v_i > \epsilon^2 \cdot v_{\max}/n\}$. Clearly, any U policy is also an $\mathcal{N}$ policy, with the restriction of not offering any products whose preference weight is at most $\epsilon^2 \cdot v_{\max}/n$. We assume without loss of generality that $U = \{1, \dots, k\}$ for some $k \le n$, and let $v_{\min}$ be the smallest weight among the products in U. By construction, $v_{\max}/v_{\min} \le n/\epsilon^2$.

Step 2: Creating weight classes. We create a new universe of products $\tilde{U}$ by modifying U as follows. First, we partition the interval $[v_{\min}, v_{\max}]$ geometrically by powers of $1+\epsilon$, into a collection of buckets $I_1, I_2, \dots, I_J$, where $J = \lfloor \frac{\log(v_{\max}/v_{\min})}{\log(1+\epsilon)} \rfloor = O_\epsilon(\log n)$. Formally,

$$I_j = [v_{\min} \cdot (1+\epsilon)^j, v_{\min} \cdot (1+\epsilon)^{j+1}\Big),$$

for $j = 0, \dots, J$. Now, we associate a product $\tilde{i}$ to each product $i \in U$, whose weight is the left endpoint of the bucket containing v_i. In other words, the universe of products $\tilde{U} = \{\tilde{1}, \dots, \tilde{k}\}$ is created such that, for every product $i = 1, \dots, k$, we determine the interval I_j where v_i resides and then set $v_{\tilde{i}} = (1+\epsilon)^j \cdot v_{\min}$. Consequently, the products in $\tilde{U}$ take only $O(J)$ possible values.

Step 3: Solving a reduced dynamic program. We proceed by explaining how to compute an optimal truncated policy $\tilde{A}$ with respect to the universe $\tilde{U}$. To this end, let us define *constrained* load vectors $\boldsymbol{\ell} = (\ell_1, \dots, \ell_k)$ as those whose maximal component is at most our threshold, that is, $\max_{i=1,\dots,k} \ell_i \le \mathcal{B}/\epsilon$. We denote by $\mathcal{L}$ the collection of all such vectors. Although at a first glance, the number of constrained vectors is exponential in n, we present in Online Appendix D.5 an efficient representation of these vectors that effectively reduces their number to a quasi-polynomial magnitude. Now, in order to compute an optimal truncated policy $\tilde{A}$ with respect to $\tilde{U}$, we solve the so-called reduced dynamic program for every load vector $\boldsymbol{\ell} \in \mathcal{L}$, given through the following recursive equations:

$$M_t(\boldsymbol{\ell}) = \max_{S \subseteq \tilde{U}} \left(M_{t-1}(\boldsymbol{\ell}) \cdot \phi_0(S) + \sum_{i \in S} M_{t-1}(\boldsymbol{\ell} + \mathbf{e}_i) \cdot \phi_i(S) \right). \tag{9}$$

However, we modify the boundary conditions of this program, such that $M_t(\boldsymbol{\ell}) = \mathcal{B}/\epsilon$ for every vector $\boldsymbol{\ell}$ with $\max_{i=1,\dots,k} \ell_i = \mathcal{B}/\epsilon$. Note that the recursive equations here are identical to those characterizing dynamic-MLA in Section 2.4, as these two programs only differ in their boundary condition.

Step 4: Recovering the approximate policy. Our final step consists of converting the $\tilde{U}$ policy $\tilde{A}$ from Step 3 into an approximate policy A with respect to U. To this end, in Online Appendix D.3, we introduce the so-called ϵ-tightness property of U and $\tilde{U}$, which stipulates that for every assortment $S \subseteq U$ and every product $i \in S$, we have $\phi_i(S) \ge (1-\epsilon) \cdot \phi_{\tilde{i}}(\tilde{S})$. It is easy to see that this property is satisfied in our case, as we have

$$\phi_i(S) \ge \frac{v_{\tilde{i}}}{1 + \frac{1}{1-\epsilon} \cdot \sum_{j \in S} v_{\tilde{j}}} \ge (1-\epsilon) \cdot \frac{v_{\tilde{i}}}{1 + \sum_{\tilde{j} \in \tilde{S}} v_{\tilde{j}}} = (1-\epsilon) \cdot \phi_{\tilde{i}}(\tilde{S}).$$

Deferring the technical details to Online Appendix D.3, we show that Lemma 23 enables us to recover a policy A whose expected maximum load is

$$\mathcal{E}^A \ge (1-\epsilon) \cdot \mathcal{E}^{\tilde{A}}. \tag{10}$$

In order to conclude the proof of Theorem 7, it remains to show that the policy described above can indeed be implemented in $O(n^{O_\epsilon(\log^3 n)})$ time, and that the computed policy attains the desired performance guarantee, namely achieving a $(1-\epsilon)$ approximation for dynamic-MLA. We defer these technical proofs and thereby the conclusion of Theorem 7 to Online Appendix D.5.

6. Numerical Studies

In this section, we conduct numerical experiments in order to examine how optimal assortments for static-MLA behave with respect to relevant model primitives, namely the number of customers T and the preference weights $v_1, , \ldots, v_n$. These experiments are mostly intended to study the sensitivity of optimal assortments with respect to model primitives.

6.1. Effect of the Parameter *T*

Here, we study how the number of customers T affects the number of products in an optimal assortment of static-MLA. We consider the following experimental setup: We fix the number of products at $n = 10$ and generate preference weights from the positive part of a normal distribution with mean μ and standard deviation $\mu/2$, varying μ in the range $\{0.05, 0.1, 0.3, 0.5\}$. We also vary the number of customers T in the range $\{2, 3, \ldots, 12\}$. For $T = 1$, offering the entire universe of products is clearly optimal, and we therefore exclude this case. For each pair (T, μ) in the specified ranges, we generate 1,000 static-MLA instances and determine the size of an optimal assortment by exhaustively enumerating all feasible assortments. When the optimal assortment is not unique, we represent each assortment S by an n-dimensional binary vector whose ith entry is one if and only if $i \in S$ and report the first optimal assortment in lexicographical order. This choice does not affect our results, as we observe that the optimal assortment in these experiments is unique due to the randomness in generating preference weights.

6.1.1. Results. Our results are summarized in Figure 1, where we generate a plot for each value of μ considered. Specifically, for every $T = 2, \ldots, 10$ (depicted on the x axis), we compute the size of the optimal assortment (depicted on the y axis) for the 1,000 generated instances. Subsequently, we present a box plot highlighting the quartiles of the obtained optimal assortment sizes. In particular, for every value of T, the endpoints of the vertical line delimit the range of values taken by the optimal assortment sizes, excluding outliers, which are represented by the points outside the delimited region. The extremities of each box represent the first and third quartile, and the horizontal line inside this box represents the median. The dotted line inside the box represents the mean.

We observe that, for all values of μ, the optimal assortment size tends to decrease as the number of customers T grows. In particular, when T is large enough, the optimal assortment ends up converging toward a single-product assortment, containing the heaviest product. Conversely, smaller values of T result in larger assortments. In particular, for $\mu = 0.05$ and $T \leq 5$, offering the whole universe of products is consistently optimal over all generated instances; then, the optimal assortment size starts decreasing for larger values of T. The rate of decrease depends on the preference weights, as one can observe in the four plots of Figure 1. To interpret this behavior, recall that the downside to offering a large number of products is the potential dispersal of demand across all offered options. However, when $\mu = 0.05$ and $T \leq 5$, the probability of capturing two or more customers is sufficiently small. Specifically, if there are either zero or one captured customers, there is no demand to disperse. Consequently, the downside of offering more products does not play a meaningful role in this case, resulting in larger optimal assortment sizes. Moreover, by comparing the plots themselves, we notice that the settings with higher values for μ yield optimal assortments with smaller sizes. We further explore this trend in Section 6.2.

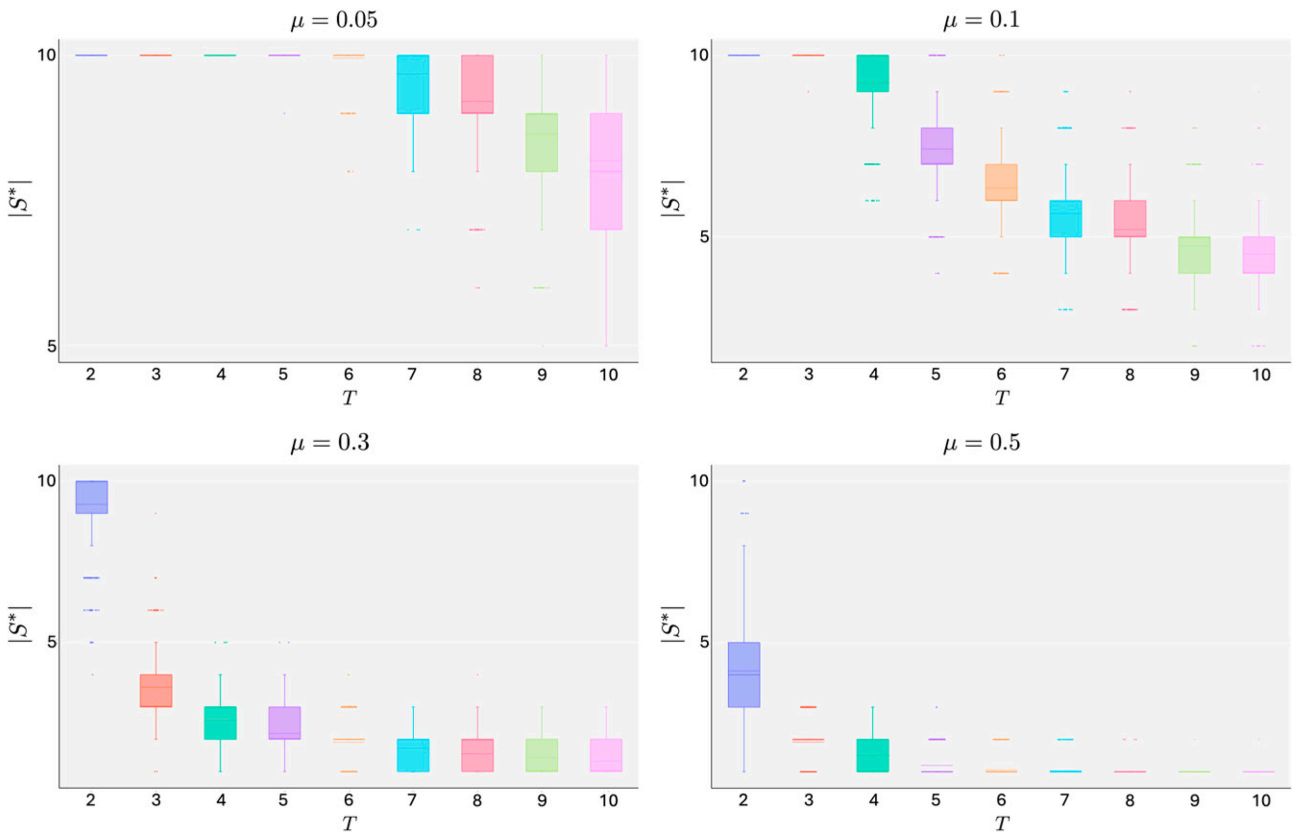
As expected, the phenomenon where the optimal assortment size shrinks as T increases aligns with Lemma 10, which states that offering only the heaviest product becomes optimal with sufficiently many customers. Let us provide an interpretation of this observation. When we offer an assortment S, the load vector of the products follows a multinomial distribution (see Section 3). In particular, these loads are negatively correlated binomial random variables. By virtue of the concentration of binomial random variables around their mean, as T grows, the loads of the less attractive products in S (i.e., with lower preference weights) become less likely to surpass those of the more attractive products. In other words, the effect of cannibalization of the attractive products by the additional products is accentuated. Consequently, as T grows, removing a greater number of unattractive products becomes optimal. This interpretation is the basic intuition behind the proof of Lemma 10.

6.2. Effect of the Preference Weights

In what follows, we study how the optimal assortment size behaves with respect to the preference weights. In the current experimental setup, the number of products is once again fixed at $n = 10$, and the preference weights are drawn from the positive part of a normal distribution with mean μ and standard deviation $\mu/2$. The number of customers T is varied in the range $\{2, 5, 8, 10\}$, and the parameter μ is varied on an additive grid in $[0.1, 1]$ with a step size of 0.1. For each pair (T, μ), we sample 1,000 static-MLA instances and compute the optimal assortment size through exhaustive enumeration.

6.2.1. Results. Our results are summarized in Figure 2, where the x axis of each plot represents the values of μ, and for each such value, we create a box plot describing the quartiles of the obtained optimal assortment sizes, similar to Figure 1. We notice that the optimal assortment size generally decreases with respect to the parameter μ. In other words, when the preference weights are

Figure 1. (Color online) Optimal Assortment Size Behavior with Respect to T



larger on average, fewer products are included in an optimal static assortment. To better understand this behavior, let us consider the two extreme cases, where the preference weights are either close to zero or very large. In the former case, the probability that a customer purchases a given product is very small, and in particular, the probability that any product is purchased twice or more is very small. Consequently, when preference weights are small, the cannibalization effect is marginal, and adding products to our assortment guarantees a larger captured portion of customers. Consequently, optimal assortments tend to increase in size for small values of preference weights, even reaching the whole universe of products for a number of instances with $\mu = 0.1$ and $T \in \{2, 5, 8\}$. When preference weights are large, a considerable portion of customers is captured even with small-sized assortments. In particular, when there exists a product i with $v_i \gg 1$, offering only this product would guarantee a load of nearly T with high probability, due to basic concentration arguments. Therefore, adding products would only cannibalize this product. In other words, the cannibalization effect outweighs the marginal increase in the captured portion of customers. In the general case, adding products to any given assortment induces both a cannibalization effect and a marginal increase in the captured portion of customers. In our experiments, for greater values of the preference weights, the benefit of capturing a larger portion of customers is overshadowed by the loss incurred due to cannibalization.

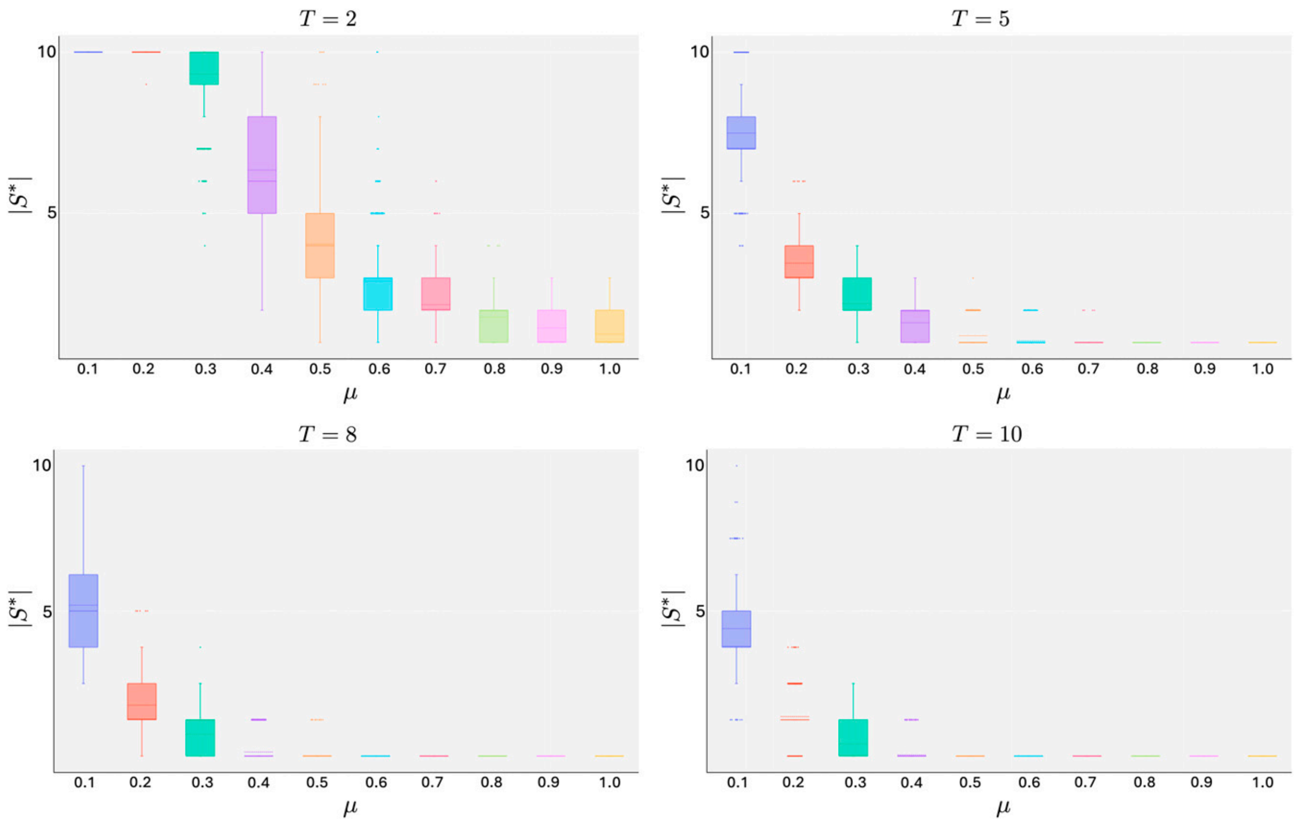
7. Concluding Remarks

This comprehensive study elucidates the potential of assortment optimization in manipulating customer choices toward maximum product selection, offering rigorous methods to address contemporary applications such as attended home delivery and preference-based group scheduling. We believe that our work lays solid foundations for maximum load assortment optimization, potentially being the onset of further exploration. In what follows, we discuss several intriguing open questions, along with particularly appealing extensions of our modeling approach.

7.1. Hardness of the Static Formulation?

Despite our best efforts, the computational complexity of the static-MLA problem remains an open question. Specifically, we still do not know whether this setting is NP-hard or whether optimal static assortments can be computed in polynomial time. This question is particularly challenging due to the unique problem structure, appearing to require either innovative optimization

Figure 2. (Color online) Optimal Assortment Size Behavior with Respect to μ



techniques or hardness proofs that are very different from what one typically meets in assortment optimization.

7.2. Dynamic Formulation: Improved Bounds and Tightness of Adaptivity Gap?

In the dynamic setting, devising a polynomial time $(1-\epsilon)$-approximate policy poses a great technical challenge due to the inherent high-dimensional nature of this problem. Through new algorithmic techniques, we have been successful at attaining quasi-polynomial running times; however, further progress seem to necessitate yet-uncharted ideas. On a different front, even though we have established a lower bound of 4/3 on the adaptivity gap of this problem, and an upper bound of 4, there is still a meaningful room for improved constructions in this context, potentially bridging this gap and identifying the exact adaptivity gap.

7.3. Practical Applications

Our work's practical implications bring forth captivating questions. In future research, it would be interesting to conduct data-driven case studies, examining the applicability of maximum load assortment optimization in real-world settings, thereby bridging the gap between theory and practice. Furthermore, exploring new domains and industries beyond attended home delivery and preference-based group scheduling could uncover novel challenges and untapped practical impact.

7.4. Extensions

Along the above-mentioned lines, extending our problem formulation to additional families of choice models, such as the Markov Chain model (Blanchet et al. 2016, Feldman and Topaloglu 2017) or the nonparametric ranking-based model (Farias et al. 2013, Aouad et al. 2018), is an interesting direction for future research. Although our evaluation oracle (see Section 3.1) could, in theory, be extended to other choice models, most of our analysis relies on specific properties of the MNL model. For instance, the crucial merge and transfer operations depend on the invariance of choice probabilities for uninvolved products. Defining these operations in other models, such as nested logit or Markov chain, is not straightforward. Moreover, in the dynamic setting, even solving a single step of the dynamic program can be NP-hard for models such as mixture of multinomial logits. Extending our results to other choice models would therefore require new approaches, outside the scope of this work. Yet another fundamental question is that of exploring a wide array of constraints on the offered assortments, such as cardinality, capacity, and matroid constraints. Finally, it would be interesting to investigate an extended formulation, where our goal is to optimize the expected summation of k-highest loads rather than solely focusing on the maximum load. At present time, this objective function appears to be significantly more challenging to deal with.

Acknowledgments

The authors thank the area editor, associate editor, and two reviewers for comments and suggestions that significantly improved the paper.

References

Adamaszek A, Wiese A (2014) A quasi-PTAS for the two-dimensional geometric knapsack problem. *Proc. 26th Ann. ACM-SIAM Sympos. Discrete Algorithms* (ACM, New York), 1491–1505.

Amorim P, DeHoratius N, Eng-Larsson F, Martins S (2024) Customer preferences for delivery service attributes in attended home delivery. *Management Sci.* 70(11):7559–7578.

Aouad A, Segev D (2023) The stability of MNL-based demand under dynamic customer substitution and its algorithmic implications. *Oper. Res.* 71(4):1216–1249.

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

Aouad A, Farias V, Levi R, Segev D (2018) The approximability of assortment optimization under ranking preferences. *Oper. Res.* 66(6):1661–1669.

Arora S, Karakostas G (2003) Approximation schemes for minimum latency problems. *SIAM J. Comput.* 32(5):1317–1337.

Azar Y, Broder AZ, Karlin AR, Upfal E (1994) Balanced allocations. *Proc. 26th Ann. ACM Sympos. Theory Comput.*, 593–602.

Bai Y, El Housni O, Rusmevichientong P, Topaloglu H (2025) Coordinated inventory stocking and assortment personalization. *Oper. Res.*, ePub ahead of print February 10, https://doi.org/10.1287/opre.2022.0651.

Bansal N, Chakrabarti A, Epstein A, Schieber B (2006) A quasi-PTAS for unsplittable flow on line graphs. *Proc. 38th Ann. ACM Sympos. Theory Comput.* (ACM, New York), 721–729.

Berry PM, Gervasio M, Peintner B, Yorke-Smith N (2007) A preference model for over-constrained meeting requests. *Proc. AAAI Workshop Preference Handling Artificial Intelligence* (AAAI, Washington, DC), 7–14.

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

Brzozowski M, Carattini K, Klemmer SR, Mihelich P, Hu J, Ng AY (2006) groupTime: Preference-based group scheduling. *Proc. SIGCHI Conf. Human Factors Comput. Systems* (ACM, New York), 1047–1056.

Campbell AM, Savelsbergh M (2006) Incentive schemes for attended home delivery services. *Transportation Sci.* 40(3):327–341.

Casazza M, Ceselli A, Létocart L (2016) Optimizing time slot allocation in single operator home delivery problems. *Proc. Ann. Internat. Conf. German Oper. Res. Soc.* (Springer, New York), 91–97.

Chan THH, Elbassioni K (2011) A QPTAS for TSP with fat weakly disjoint neighborhoods in doubling metrics. *Discrete Comput. Geometry* 46:704–723.

Chekuri C, Khanna S (2002) Approximation schemes for preemptive weighted flow time. *Proc. 34th Ann. ACM Sympos. Theory Comput.* (ACM, New York), 297–305.

Das A, Mathieu C (2015) A quasipolynomial time approximation scheme for Euclidean capacitated vehicle routing. *Algorithmica* 73(1):115–142.

Davis JM, Gallego G, Topaloglu H (2014) Assortment optimization under variants of the nested logit model. *Oper. Res.* 62(2):250–273.

Désir A, Goyal V, Jagabathula S, Segev D (2021) Mallows-smoothed distribution over rankings approach for modeling choice. *Oper. Res.* 69(4):1206–1227.

El Housni O, Topaloglu H (2023) Joint assortment optimization and customization under a mixture of multinomial logit models: On the value of personalized assortments. *Oper. Res.* 71(4):1197–1215.

Farias VF, Jagabathula S, Shah D (2013) A nonparametric approach to modeling choice with limited data. *Management Sci.* 59(2): 305–322.

Feldman JB, Topaloglu H (2017) Revenue management under the Markov chain choice model. *Oper. Res.* 65(5):1322–1342.

Frey J (2009) An algorithm for computing rectangular multinomial probabilities. *J. Statist. Comput. Simulations* 79(12):1483–1489.

Gallego G, Topaloglu H (2014) Constrained assortment optimization for the nested logit model. *Management Sci.* 60(10):2583–2601.

Gallego G, Topaloglu H (2019) *Revenue Management and Pricing Analytics* (Springer, New York).

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

Gao P, Ma Y, Chen N, Gallego G, Li A, Rusmevichientong P, Topaloglu H (2021) Assortment optimization and pricing under the multinomial logit model with impatient customers: Sequential recommendation and selection. *Oper. Res.* 69(5): 1509–1532.

Hausman J, McFadden D (1984) Specification tests for the multinomial logit model. *Econometrica* 52(5):1219–1240.

Kök AG, Fisher ML, Vaidyanathan R (2009) Assortment planning: Review of literature and industry practice. *Retail Supply Chain Management: Quantitative Models and Empirical Studies* (Springer, New York), 99–153.

Lebrun R (2013) Efficient time/space algorithm to compute rectangular probabilities of multinomial, multivariate hypergeometric and multivariate pólya distributions. *Statist. Comput.* 23(5):615–623.

Lipton RJ, Markakis E, Mehta A (2003) Playing large games using simple strategies. *Proc. 4th ACM Conf. Electronic Commerce* (ACM, New York), 36–41.

Luce RD (1959) *Individual Choice Behavior* (John Wiley, Hoboken, NJ).

Mackert J (2019) Choice-based dynamic time slot management in attended home delivery. *Comput. Industrial Engrg.* 129:333–345.

Mahajan S, Van Ryzin G (2001) Stocking retail assortments under dynamic consumer substitution. *Oper. Res.* 49(3):334–351.

Manerba D, Mansini R, Zanotti R (2018) Attended home delivery: Reducing last-mile environmental impact by changing customer habits. *IFAC Papers Online* 51(5):55–60.

McFadden D (1973) *Conditional Logit Analysis of Qualitative Choice Behavior* (Institute of Urban and Regional Development, University of California, Berkeley).

Mirrokni V, Thorup M, Zadimoghaddam M (2018) Consistent hashing with bounded loads. *Proc. 29th Ann. ACM-SIAM Sympos. Discrete Algorithms* (ACM, New York), 587–604.

Mitzenmacher M, Upfal E (2017) *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis* (Cambridge University Press, Cambridge, UK).

Mustafa NH, Raman R, Ray S (2015) Quasi-polynomial time approximation scheme for weighted geometric set cover on pseudodisks and halfspaces. *SIAM J. Comput.* 44(6):1650–1669.

Phillips RL (2021) *Pricing and Revenue Optimization* (Stanford University Press, Redwood City, CA).

Raab M, Steger A (1998) "Balls into bins": A simple and tight analysis. *Proc. Internat. Workshop Randomization Approximation Techniques Comput. Sci.* (Springer, New York), 159–170.

Remy J, Steger A (2009) A quasi-polynomial time approximation scheme for minimum weight triangulation. *J. ACM* 56(3):1–47.

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

Sumida M, Gallego G, Rusmevichientong P, Topaloglu H, Davis J (2021) Revenue-utility tradeoff in assortment optimization under the multinomial logit model with totally unimodular constraints. *Management Sci.* 67(5):2845–2869.

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

Truden C, Maier K, Jellen A, Hungerländer P (2022) Computational approaches for grocery home delivery services. *Algorithms* 15(4):125.

van der Hagen L, Agatz N, Spliet R, Visser TR, Kok L (2024) Machine learning–based feasibility checks for dynamic time slot management. *Transportation Sci.* 58(1):94–109.

Waβmuth K, Köhler C, Agatz N, Fleischmann M (2023) Demand management for attended home delivery: A literature review. *Eur. J. Oper. Res.* 311(3):801–815.

Yang X, Strauss AK (2017) An approximate dynamic programming approach to attended home delivery management. *Eur. J. Oper. Res.* 263(3):935–945.

Omar El Housni is an assistant professor in the School of Operations Research and Information Engineering at Cornell University and Cornell Tech. His research focuses on decision making under uncertainty, revenue management, assortment optimization, and matching.

Marouane Ibn Brahim is a PhD student in the School of Operations Research and Information Engineering at Cornell University and Cornell Tech. His research focuses on revenue management and assortment optimization.

Danny Segev is a faculty member at the School of Mathematical Sciences, Tel Aviv University. His research focuses on stochastic, dynamic, and combinatorial optimization, as well as on computational revenue management.