Joint Product Framing (Display, Ranking, Pricing) and Order Fulfillment under the MNL Model for E-Commerce Retailers

Yanzhe (Murray) Lei
Stephen J.R. Smith School of Business, Queen's University, Kingston, Ontario, Canada
yl64@queensu.ca

Stefanus Jasin, Joline Uichanco
Stephen M. Ross School of Business, University of Michigan, Ann Arbor, MI, USA
sjasin, jolineu@umich.edu

Andrew Vakhutinsky
Oracle Labs, Burlington, MA
andrew.vakhutinsky@oracle.com

Problem definition: We study a joint product framing and order fulfillment problem with both inventory and cardinality constraints faced by an e-commerce retailer. There is a finite selling horizon and no replenishment opportunity. In each period, the retailer needs to decide how to "frame" (i.e., display, rank, price) each product on her website as well as how to fulfill a new demand. **Academic/Practical Relevance:** E-commerce retail is known to suffer from thin profit margins. Using the data from a major U.S. retailer, we show that jointly planning product framing and order fulfillment can have a significant impact on online retailers' profitability. This is a technically challenging problem as it involves both inventory and cardinality constraints. In this paper, we make progress toward resolving this challenge. **Methodology:** We use techniques such as randomized algorithms and graph-based algorithms to provide a tractable solution heuristic which we analyze through asymptotic analysis. **Results:** Our proposed randomized heuristic policy is based on the solution of a deterministic approximation to the stochastic control problem. The key challenge is in constructing a randomization scheme that is easy to implement and that guarantees the resulting policy is asymptotically optimal. We propose a novel two-step randomization scheme based on the idea of matrix decomposition and a re-scaling argument. **Managerial Implications:** Our numerical tests show that the proposed policy is very close to optimal, can be applied to large-scale problems in practice, and highlights the value of jointly optimizing product framing and order fulfillment decisions. When inventory across the network is imbalanced, the widespread practice of planning product framing without considering its impact on fulfillment can result in high shipping costs, regardless of the fulfillment policy used. Our proposed policy significantly reduces shipping costs by using product framing to manage demand so that it occurs close to the location of the inventory.

1. Introduction

E-commerce sales is an important driver of growth in the U.S. retail industry. According to the U.S. Census Bureau (U.S. Census Bureau 2020), the total retail sales in the U.S. amounted to more than \$5.4 trillion from Q4 2018 to Q3 2019. During this same period, e-commerce contributed to about 11% of the total retail sales. Although this is a small portion, e-commerce represented 58%

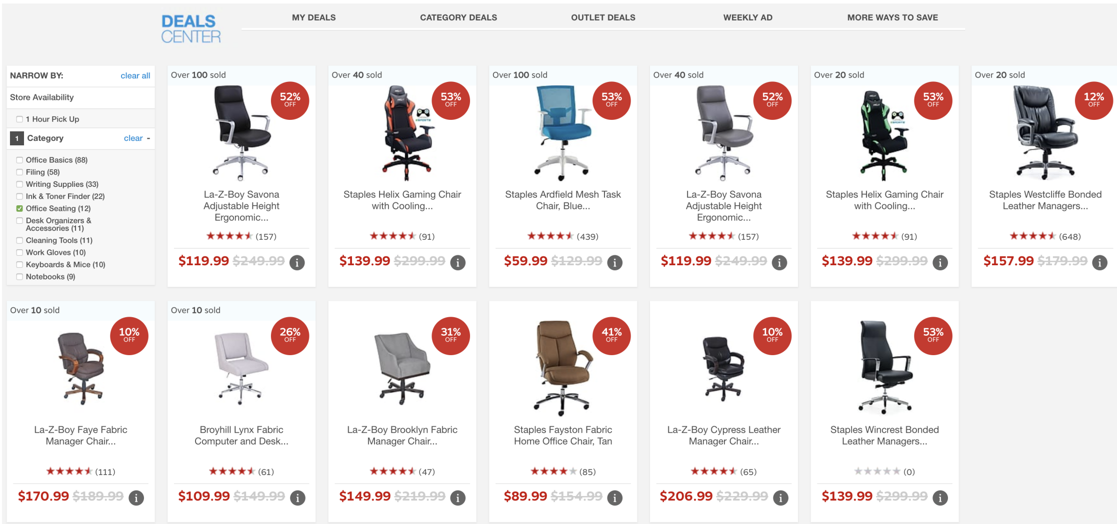


Figure 1 Clearance deals webpage at an online retailer showing products in the Office Seating category.

of all growth in retail sales in this period. The year-over-year growth in U.S. e-commerce retail sales has been between 11% to 17% in every quarter from 2012 to 2019, and due to the COVID-19 pandemic, the growth was more than 40% in 2020. Hence, e-commerce is an increasingly important sales channel in the retail industry. However, online retailing also has many operational challenges that do not afflict a brick-and-mortar retailer. A 2017 National Retail Foundation survey of North American retailers revealed that the costs of running an e-commerce business are rising. These costs include marketing costs, costs of returns, and fulfillment costs.

Given the high cost of operating an online channel, revenue management can be an important tool to enhance profitability. This is because the online setting provides many potential opportunities for revenue management through the real-time personalization of the shopping experience for each customer. Specifically, an online retailer can control how products are being *framed* in the website. The *framing effect*, demonstrated by Tversky and Kahneman (1981) in psychology literature, is the phenomenon that individual preferences are affected by the way in which the options are presented. In an e-commerce setting, *product framing* refers to the ways products can be displayed, ranked, and priced on the site to affect a customer's purchase probability. Figure 1 is an example of how products are framed in an online store. The concept of product framing was brought to the Revenue Management (RM) literature by Abeliuk et al. (2016), Aouad and Segev (2020) and Gallego et al. (2020) (see more detailed discussion in Section 2).

The RM literature on product framing studies the effectiveness of framing in improving total revenues. However, the impact of product framing strategies on profitability and fulfillment costs is a largely unexplored topic. In fact, ignoring the effect of framing on fulfillment costs can hurt an online retailer's overall profit. This is especially true when the inventory is scarce and unevenly distributed across a network of fulfillment centers (FCs), so the location of the demand (which can be controlled by the framing strategy) has an important bearing on the fulfillment cost. Acimovic and Graves (2017) argue that inventory imbalance is a common occurrence in online supply chains due to various operational realities. We observe the same occurrence when we examined the inventory data of our industry partner, a major U.S. retailer, at the start of its clearance sales season. In a specific product category, more than 40% of the products started the clearance season stocked out in at least 80% of the FCs. Hence, the fulfillment cost can be high if the purchasing customer is located far from an in-stock FC. Yet, balancing the inventory through transshipment is not done by this retailer due to the long distances between FCs and, more prohibitively, the cost and complexity involved in picking and preparing FC inventory for transshipment.

In this paper, we propose a strategy to improve total profits by customizing product framing based on a combination of customers' characteristics (local demand) and inventory levels (supply). Indeed, as we show with the data of our partner retailer, a significant profit improvement can be achieved by a product framing policy that takes into account the impact on fulfillment costs. To derive this insight, we study a multi-period *Joint Framing and Fulfillment* (JFF) problem with multiple products, multiple customer regions, and multiple FCs with finite non-replenishable inventory. In JFF, an online retailer decides in real-time how to frame the products to an arriving customer, and how to fulfill any realized sale. We assume that customers make their purchase decisions according to the classic multinomial logit (MNL) choice model whose parameters are affected by product framing. The objective of the online retailer is to optimize the total expected profit (defined as revenue minus fulfillment cost) while respecting the inventory constraints.

The product framing decision in our model is sufficiently general (it goes beyond the types of framing decisions discussed in Abeliuk et al. 2016, Aouad and Segev 2020, Gallego et al. 2020) and includes the following applications: deciding an assortment of products to display, assigning the products to regions (top, middle, bottom) of a website's content area, sequencing (ranking) the products, choosing which products to promote, selecting prices to offer, or any combination thereof. To capture these different applications, we consider a model in which the retailer is deciding how to assign each product to one of multiple *framing options* or *framing groups*. This assignment must satisfy certain cardinality constraints (e.g., the number of products in each group cannot be larger than a certain number). We provide some examples in Section 3 to show how the abstraction of framing options/groups can be used to model some of the applications mentioned above.

To the best of our knowledge, our work is the first in the literature to model a setting with joint framing and fulfillment controls. JFF is a practically challenging problem as the framing decisions for different customer locations are interconnected due to the presence of the fulfillment decisions that force those locations to share the finite inventory across multiple FCs. Indeed, as will be clear in this paper, even the pure product framing problem with inventory constraints (i.e., with a single customer region and a single FC) is technically challenging to solve. Our paper makes an important contribution by developing a framing policy that is provably asymptotically optimal in the JFF problem. This algorithmic advance is achieved by exploiting the structure of the MNL choice model. Hence, our work establishes that this popular choice model is suitable for developing a cost-aware framing policy that can be implemented within a decision-support tool.

Results and contributions. The results and insights of our work are as follows:

1. Given the intractability of solving JFF, we approximate JFF with its deterministic relaxation. The resulting deterministic problem can be reformulated as a tractable linear program (LP) with a polynomial number of decision variables and constraints. We show that the optimal value of this LP is an upper bound of the optimal value of JFF. This is analogous to the standard upper-bound results in the RM literature (e.g., Gallego and Van Ryzin 1994, 1997). However, due to the different structure in the optimization problem under JFF, the proof of this result requires a different argument than that used in the RM literature.
2. Motivated by the upper-bound result, we construct a policy based on the solution to the deterministic relaxation. Due to the inventory constraints, the LP solution could be fractional and may not correspond to a feasible framing configuration. However, we show that the LP solution can always be decomposed as a weighted sum of integral matrices, where each matrix corresponds to a framing configuration that satisfies the cardinality constraints. We then propose a randomized framing policy that treats the weights as the probabilities of offering the framing configurations in the decomposition. Furthermore, if the weights are re-scaled, the resulting randomized policy is provably asymptotically optimal.
3. The asymptotic optimality of our randomized policy does not depend on the specific matrix decomposition algorithm used. However, it may be practically desirable if the randomization is over a small set of framing configurations, since sampling from a high-cardinality set is computationally challenging. We propose a new polynomial-time algorithm that decomposes the fractional framing solution with a small number of integral framing matrices. In numerical experiments, we show that the decomposition size of our algorithm can be an order of magnitude smaller than the state-of-the-art decomposition algorithm in the literature.
4. Using the data from a major U.S. retailer, we gain insights into the value of joint framing and fulfillment decisions by comparing the performance of our randomized policy to those of policies that optimize framing without considering its impact on fulfillment. Our policy reduces total shipping costs by 23% by using product framing to manage demand across locations so that demands occur close to the FCs with available inventories. We test the sensitivity of our policy to different arrival rates and find that, in all test instances, the optimality gap of our

randomized policy is consistently below 2.5%. We also test some variants of our policy where the framing decision at each location is made infrequently, i.e., only changed once a week. Even with infrequent changes, the optimality gap of our policy is no larger than 4%. Since the probabilities are not updated with the inventory state, the framing policy can be essentially calibrated at the start of the horizon. Thus, our proposed policy is practical, near-optimal, and can be applied to large-scale problems commonly in practice.

Our work opens new avenues for research in joint product framing and fulfillment strategies. We mention several promising future directions in Section 7. For example, while the MNL model is very popular both in theory and in practice, it is known to suffer from limitations. Thus, a potentially impactful direction is to develop randomized policies for more general choice models, such as nested logit. Another direction is to extend our setting to overlapping framing groups, which would allow applications wherein a product can be assigned to multiple framing groups.

Organization of the paper. The remainder of the paper is organized as follows. We discuss related literature in Section 2. To better highlight our approach, we first discuss a base model where the online retailer only needs to decide on product framing in Sections 3 and 4. In Section 5, we discuss the JFF problem where the online retailer needs to decide both product framing and fulfillment. Finally, in Section 6, we report results from numerical experiments and in Section 7 we conclude the paper. All proofs can be found in the Electronic Companion.

2. Literature Review

Broadly speaking, our work is related to the vast literature on assortment, pricing, and fulfillment optimization. Due to space limitation, we will only focus on the most closely related literature.

Assortment Optimization. Assortment optimization refers to the problem of selecting a set of products to be displayed to customers with the goal of maximizing total expected revenues. The literature on assortment optimization has grown steadily over the past decades (see Kök et al. (2015) for a recent review). Researchers have been devising efficient optimization methods under different assumptions on how a customer's preference is formed (i.e., different choice models), including multinomial logit (MNL) model (Talluri and van Ryzin 2004, Rusmevichientong et al. 2010), nested logit model (Gallego and Topaloglu 2014, Feldman and Topaloglu 2015, Li et al. 2015), Markov chain choice model (Désir et al. 2020, Feldman and Topaloglu 2017), and mixture of multinomial logit model (Rusmevichientong et al. 2014, Désir et al. 2016).

From the choice modeling perspective, our work is closely related to a line of recent works that capture the impact of product display on a customer's decision (e.g., Davis et al. 2015, Abeliuk et al. 2016, Aouad and Segev 2020 Gallego et al. 2020, Çetin et al. 2019 and Sumida et al. 2020). Similar to ours, Sumida et al. (2020) and Abeliuk et al. (2016) also assume that a customer's preference is formed under the MNL model. Sumida et al. (2020) show that, when facing totally unimodular constraints, the revenue optimization problem can be equivalently reformulated as an LP. Similar to Sumida et al. (2020), we also construct an LP whose solution is used to construct our heuristic policy. However, due to the presence of inventory constraints, our constraints are not totally unimodular and our LP solution is fractional. Çetin et al. (2019) examine a brick-and-mortar retailer's optimal promotional display decision using a nested-MNL model. The choice models considered by Aouad and Segev (2020) and Gallego et al. (2020) are richer than ours since they allow customers to have different consideration sets of products. The common theme of this line of works is to find a static framing configuration that has a near-optimal revenue guarantee in the setting *without* inventory constraints. Our work complements these works by considering the problem with a dynamic framing configuration in the setting *with* inventory constraints *and* cardinality constraints. Another line of works considers the construction of near-optimal dynamic assortment (and pricing) policy under inventory constraints (e.g., Golrezaei et al. 2014, Bernstein et al. 2015, Ma and Simchi-Levi 2020). From the technical perspective, Golrezaei et al. (2014) and Ma and Simchi-Levi (2020) derive competitive ratio guarantees, whereas our work focuses on the asymptotic setting with large demand and initial inventory. However, these works neglect the impact of framing configuration, such as the effect of display position. They also ignore fulfillment costs, which is a key feature in our model.

Pricing Optimization. Research on pricing studies how a firm should change price dynamically over time to mitigate the mismatch between the supply and demand in order to maximize the revenues. Some classic papers on dynamic pricing that study the effectiveness of simple heuristic policies include Gallego and Van Ryzin (1994, 1997). Jasin (2014) and Chen et al. (2015) study the same setting analyzed in Gallego and Van Ryzin (1994, 1997) and provide improved heuristic policies that can be implemented in real time without re-optimization. Although dynamic pricing was popularized by its application in the airline industry, the recent advent of e-commerce has drawn researchers' attention to devising effective pricing policies under various business settings (see Chen 2014 for a recent review). For example, Caro and Gallien (2012), Ferreira et al. (2015) and Harsha et al. (2019) all propose demand prediction techniques and price optimization algorithms whose effectivenesses are validated through field experiments. There are also works that consider joint assortment and pricing decisions. under many different choice models (e.g., Maddah and Bish 2007, Wang 2012, Kök and Xu 2011, Gallego and Topaloglu 2014, Gallego et al. 2020). Compared to our model, none of the aforementioned works consider the impact of the framing configuration on customer choice (except for the aforementioned Gallego et al. 2020) and the challenge that arises when fulfillment decisions also need to be made jointly.

Fulfillment Optimization. Given the high cost of operating an online retail channel, large online retailers strive to achieve efficiency in various aspects of their supply chains. The fulfillment component of our model exclusively focuses on the assignment of outbound shipping (i.e., from which warehouse to ship the requested product), an operation which induces a significant amount of logistical costs for every online retailer. There is a growing body of literature that studies how to optimize the fulfillment assignment decisions dynamically; see Acimovic and Farias (2019) for a recent review. Xu et al. (2009) propose real-time strategies on the order assignment decisions with an objective of minimizing the total number of shipments (i.e., packages shipped). Under different assumptions on the arrival process, Acimovic and Graves (2014) and Andrews et al. (2019) both study a similar problem with the objective of minimizing total outbound shipping costs. Jasin and Sinha (2015) consider a multiproduct fulfillment problem where each shipment can contain multiple items. They proposed an LP-based policy and provide analytical bounds on their performances. Acimovic and Graves (2017) consider the fulfillment cost minimization from the perspective of inventory allocation. Assuming a myopic assignment policy, they propose a joint replenishment strategy that improves the status quo policy numerically. Lei et al. (2018) consider joint pricing and fulfillment optimization. However, they do not consider the impact of a more general framing configuration discussed in our work.

3. Base Model

We first consider a base model where the only decision that needs to be considered is product framing under inventory constraints. The base model will be used in addressing the joint framing and fulfillment (JFF) problem in Section 5. We will use $[N]$ to denote the set $\{1,\dots,N\}$ for any $N \in \mathbb{N}_+$, and $\mathbf{1}$ to denote a column vector of ones with a proper dimension. Unless otherwise noted, all vectors are to be treated as column vectors. We use the notation $\mathbf{a} \equiv \mathbf{b}$ for vectors $\mathbf{a} = (a_i)_{i=1}^n, \mathbf{b} = (b_i)_{i=1}^n$ if $a_i = b_i$ for all i.

3.1. Problem setting

We consider a monopolistic online retailer who sells a catalog of I products throughout a finite (discrete) selling season with T periods. We assume that at most one customer arrives in each period and each customer purchases at most one product. This is a standard assumption in the literature (see Jasin 2014). The inventory level of product $i \in [I]$ is C_i at the start of the selling season. Since we focus on tactical decisions instead of the strategic inventory decision, similar to existing papers in the literature (e.g., Golrezaei et al. 2014, Bernstein et al. 2015, Ma and Simchi-Levi 2020), we assume *no* replenishment during the selling season. This is consistent with the practice of our industry partner of no replenishments or transshipments during clearance sales.

There are R framing options (or groups) to which a product can be assigned. Each product must be assigned to exactly one group, and each group $r \in [R]$ has a cardinality constraint $Q_r \in \mathbb{N}_+$, meaning that at most Q_r products can be assigned to group r. To allow the possibility that some products are not displayed to the customers (e.g., in the case of stock-outs), without loss of generality, we assume that group R corresponds to a "no display" group. Specifically, if a product is assigned to group R, then it will not be displayed to the customer and hence will have zero demand. Below is an example of how to apply the concept of "framing groups" to an application:

- *Constrained assortment problem.* Suppose the retailer needs to choose an assortment of products to display on the webpage. It can be modeled as follows. We define two groups: Group 1 represents the set of products to be displayed and group 2 represents the set of products not to be displayed. If there is a restriction that we can only display at most 10 ($< I$) products at a time, then we simply set $Q_1 = 10$ and $Q_2 = I$.

In general, it may sometimes make sense to split the groups into two sets $\mathcal{R}_1$ and $\mathcal{R}_2$, where (i) at most Q_r products can be assigned to group $r \in \mathcal{R}_1$ and (ii) exactly Q_r products must be assigned to group $r \in \mathcal{R}_2$. Although for simplicity of our analysis we assume $\mathcal{R}_2 = \emptyset$, extension to a non-empty $\mathcal{R}_2$ is straightforward, and it allows us to use "framing groups" for the following applications:

- *Pure ranking problem.* Suppose the retailer needs to decide how to sort a set of I product to optimize revenue. We can define $R = I$ groups with $Q_r = 1$ for all r, where each group can be interpreted as a specific position in a display page. Moreover, since exactly one product must be displayed at each position, we use equality sign in all the cardinality constraints.
- *Mixed framing problem.* The objective is to decide which set of products is to be displayed at each page (or a section of a page), where a page (or a section of a page) could refer to either a regular display page, a special promotion page, the upper part of a top-deal page, the bottom part of a top-deal page, a small promotional banner, etc. In this case, each group represents the set of products to be displayed at each page (or section of a page) and Q_r denotes the exact (or maximum) number of products that can be displayed on page r.

Our industry partner is facing a mixed framing problem since it needs to determine which products should be displayed under a promotional banner on the clearance deals page. The retailer's practice is to remove any out-of-stock product from the display, which corresponds to assigning it to the "no display" group. Hence, the retailer has 3 groups: displayed (promoted), displayed (not promoted), and not displayed. Nevertheless, in what follows, we present a general model for any $R \le I$.

Let $X_{ir}^t \in \{0,1\}$ denote the framing decision for product i in period t. At the beginning of period t, a new customer (he) arrives with probability $\lambda \in [0,1]$. Let $\boldsymbol{X}^t = (X_{ir}^t)_{i\in[I], r\in[R]}$ denote the vector of the online retailer's framing decision in period t. The customer's purchase decision is determined by the online retailer's framing decision. Specifically, let $D_i^t(\boldsymbol{X}^t) \in \{0,1\}$ be a Bernoulli random variable denoting the joint event of a customer's arrival and his purchase decision of product i given how all products are framed, $\boldsymbol{X}^t$. We denote by $\theta_i(\boldsymbol{X}^t) = \mathbb{P}(D_i^t(\boldsymbol{X}^t) = 1)$ the purchase probability of product i given $\boldsymbol{X}^t$. We assume that a customer's purchase decision can be modeled by the Multinomial Logit (MNL) choice model, i.e.,

$$\theta_i(\boldsymbol{X}^t) = \lambda \cdot \frac{\sum_{r\in[R]} v_{ir} \cdot X_{ir}^t}{v_0 + \sum_{\ell\in[I], r\in[R]} v_{\ell r} \cdot X_{\ell r}^t}, \tag{1}$$

where v_0 captures the utility of no-purchase and v_{ir} captures the utility of purchasing product i when it is displayed in group r. In the standard MNL framework, we can also write: $v_0 = \exp\{u_0\}$ and $v_{ir} = \exp\{u_{ir}\}$ with $u_{ir} = a_i - b_i \cdot p_{ir} + \gamma_{ir}$, where u_{ir} is the expected utility of purchasing product i when it is displayed based on framing option r, p_{ir} is the price for product i under framing option r, b_i is the effect of price on the utility from product i, and γ_{ir} is the non-price effect of framing option r which, in general, could be product-dependent. Given the framing decision $\boldsymbol{X}^t$, the expected revenue from product i is

$$\mathcal{R}_i(\boldsymbol{X}^t) := \lambda \cdot \frac{\sum_{r\in[R]} v_{ir} \cdot X_{ir}^t \cdot p_{ir}}{v_0 + \sum_{\ell\in[I], r\in[R]} v_{\ell r} \cdot X_{\ell r}^t}. \tag{2}$$

Without loss of generality, from this point onward, we will assume that $v_0 = 1$.

It is worth noting that the MNL choice model is one of the most commonly used choice models in the economics, marketing, and operations management literatures. In addition to MNL, our proposed approach in this paper can also be used in conjunction with the so-called *General Attraction Model* (GAM), which addresses some of the deficiencies of the MNL choice model (Gallego et al. 2014). We discuss the limitation of an MNL choice model in Section 7.

3.2. Stochastic control formulation

The online retailer's objective is to maximize her total expected revenue. Let Π denote the set of all feasible non-anticipating policies. The optimal control formulation of the retailer's problem is

$$\textbf{(OPT):} \quad \mathcal{J}^* = \max_{\pi\in\Pi} \sum_{t\in[T], i\in[I]} \mathbb{E}\left[\mathcal{R}_i(\boldsymbol{X}^{t,\pi})\right]$$

$$\text{s.t.} \quad \sum_{i\in[I]} X_{ir}^{t,\pi} \le Q_r \qquad \forall\, r, t, \tag{3a}$$

$$\sum_{r\in[R]} X_{ir}^{t,\pi} = 1 \qquad \forall\, i, t, \tag{3b}$$

$$\sum_{t\in[T]} D_i^t(\boldsymbol{X}^{t,\pi}) \le C_i \qquad \forall\, i, \tag{3c}$$

$$X_{ir}^{t,\pi} \in \{0,1\} \qquad \forall\, i, r, t, \tag{3d}$$

where all the constraints must hold almost surely. The meaning of each constraint is obvious. Note that, in general, the summation in (3a) can be taken over a pre-specified set $\mathcal{A}_r \subseteq [I]$ and, similarly, the summation in (3b) can be taken over a pre-specified set $\mathcal{B}_i \subseteq [R]$. This can be useful in the case where some framing options are not feasible for certain products (e.g., when we do not want apply any discount to product 1, or product 2 can only be displayed in certain sections of the website). However, since it will not affect our analysis, for expositional simplicity, we will simply assume that the summations are taken over $[I]$ and $[R]$, respectively.

Let $\boldsymbol{C} = (C_i)_{i\in[I]}$ and $\boldsymbol{D}^t(\boldsymbol{X}^t) = (D_i^t(\boldsymbol{X}^t))_{i\in[I]}$. We can write **OPT** more compactly as follows:

$$
\begin{aligned}
\mathcal{J}^* = \max_{\pi\in\Pi} \quad & \sum_{t\in[T],i\in[I]} \mathbb{E}\left[\mathcal{R}_i(\boldsymbol{X}^{t,\pi})\right] \\
\text{s.t.} \quad & A_1\boldsymbol{X}^{t,\pi} \leq \boldsymbol{Q} \qquad \forall t, && \text{(4a)}\\
& A_2\boldsymbol{X}^{t,\pi} = \mathbf{1} \qquad \forall t, && \text{(4b)}\\
& \sum_{t\in[T]} \boldsymbol{D}^{t,\pi}(\boldsymbol{X}^{t,\pi}) \leq \boldsymbol{C}, && \text{(4c)}\\
& \boldsymbol{X}^{t,\pi} \in \{0,1\}^{IR} \qquad \forall t, && \text{(4d)}
\end{aligned}
$$

where A_1 (A_2) is a coefficient matrix corresponding to constraint (3a) (constraint (3b)) and $\boldsymbol{Q}$ ($\mathbf{1}$) is a coefficient vector corresponding to the right side of constraint (3a) (constraint (3b)).

Let π^* denote the optimal framing policy for **OPT**. In theory, π^* can be computed exactly by solving a Dynamic Programming (DP). However, it is computationally intractable when the number of products is large due to the curse of dimensionality. Motivated by this, instead of trying to compute π^*, we will focus on constructing a provably near-optimal heuristic policy using an optimal solution of a deterministic approximation to **OPT**. We discuss this approximation next.

3.3. A deterministic approximation

Let $\boldsymbol{\theta}(\boldsymbol{x}) = (\theta_i(\boldsymbol{x}))_{i\in[I]}$ for any vector $\boldsymbol{x} \in [0,1]^{IR}$, where $\theta_i(\cdot)$ is the function defined in (1). Consider the following deterministic optimization problem:

$$
\begin{aligned}
\textbf{(DET):} \quad \mathcal{J}^D = \max_{\{\boldsymbol{x}^t\}} \quad & \sum_{t\in[T],i\in[I]} \mathcal{R}_i(\boldsymbol{x}^t) \\
\text{s.t.} \quad & A_1\boldsymbol{x}^t \leq \boldsymbol{Q} \qquad \forall t, && \text{(5a)}\\
& A_2\boldsymbol{x}^t = \mathbf{1} \qquad \forall t, && \text{(5b)}\\
& \sum_{t\in[T]} \boldsymbol{\theta}(\boldsymbol{x}^t) \leq \boldsymbol{C}, && \text{(5c)}\\
& \boldsymbol{x}^t \in [0,1]^{IR} \qquad \forall t, && \text{(5d)}
\end{aligned}
$$

where $\boldsymbol{x}^t = (x_{ir}^t)_{i\in[I],r\in[R]}$. One can arrive at the above formulation by replacing the random variables in **OPT** with their deterministic values. Let $\boldsymbol{x}^{t,D}$ denote an optimal solution of **DET**.

Note that **DET** has linear constraints in (5a),(5b),(5d), while the objective and the left-hand side of (5c) are the sum of ratios of linear functions. **DET** can be written as a linear program using the standard Charnes-Cooper variable transformation (see Charnes and Cooper 1962). Since the resulting LP has only a polynomial size, we can solve for $\boldsymbol{x}^{t,D}$ efficiently. We state a lemma.

Lemma 1. *$\mathcal{J}^* \leq \mathcal{J}^D$.*

Lemma 1 provides us with an easily computable benchmark (i.e., $\mathcal{J}^D$) for performance analysis. In addition, it can be shown that $\mathcal{J}^D$ is a sufficiently tight upper bound of $\mathcal{J}^*$, in the sense that $\mathcal{J}^*/\mathcal{J}^D \approx 1$ in the setting with large demand and large inventory (see Section 4). This suggests that a heuristic policy π constructed using the deterministic optimal solution $\boldsymbol{x}^{t,D}$ might be near-optimal provided that its total expected revenue well-approximates $\mathcal{J}^D$. The important question, which we address in Section 4, is *how* to construct π. As a preamble to Section 4, we briefly discuss why constructing π is *not* trivial. The challenge arises since $\boldsymbol{x}^{t,D}$ may not be integral and needs to be mapped into a probabilistic (or randomized) policy π that satisfy the following conditions:

$$\sum_{i\in[I]} X_{ir}^{t,\pi} \;\leq\; Q_r \qquad \forall\, r, t \tag{6a}$$

$$\sum_{r\in[R]} X_{ir}^{t,\pi} \;=\; 1 \qquad \forall\, i, t \tag{6b}$$

$$\mathbf{E}\left[\theta_i(\boldsymbol{X}^{t,\pi})\right] \;=\; \theta_i(\boldsymbol{x}^{t,D}) \qquad \forall\, i, t \tag{6c}$$

$$\mathbf{E}\left[\mathcal{R}_i(\boldsymbol{X}^{t,\pi})\right] \;=\; \mathcal{R}_i(\boldsymbol{x}^{t,D}) \qquad \forall\, i, t \tag{6d}$$

The first two conditions follow from the definition of a feasible policy in Π and must be satisfied almost surely. The last two conditions guarantee that, on average, the inventory constraints are satisfied and the expected total revenue is equal to the total revenue in **DET**. We will show in Section 4 that conditions (6a)–(6d) are sufficient to guarantee the near-optimality of π (in some sense). Note that $\theta_i(\cdot)$ and $\mathcal{R}_i(\cdot)$ defined in (1)–(2) are *non-linear* functions, hence, a framing policy that satisfies the marginal distribution conditions does not necessarily satisfy conditions (6c) and (6d). Indeed, it is *a priori* not clear that a framing policy that satisfies conditions (6a)–(6d) even exists. Surprisingly, we show in Section 4 that it does exist and its construction requires a combination of matrix decomposition techniques and a certain *re-scaling* argument.

4. Proposed Randomized Framing Policy

We next present our heuristic policy for **OPT**, which we refer to as the *Randomized Framing* (RF) policy. We start by introducing some notations that will be useful for our discussions. For time period t, we denote the framing assignment by the vector $\phi^t = (\phi^t(1), \ldots, \phi^t(I))$, where $\phi^t(i) = r$ means that, in period t, we apply framing option r to product i. For notational simplicity, we will often neglect the superscripts on ϕ whenever we are not referring to a customer at a specific time period. Given the cardinality vector $\boldsymbol{Q} = (Q_r)$, we define the set of all feasible framing assignments as $\Omega(\boldsymbol{Q}) := \{\phi \in [R]^I : \sum_{i=1}^{I} \mathbf{1}\{\phi(i) = r\} \leq Q_r \ \forall\, r\}$.

4.1. Description and performance of the policy

Framing policy RF chooses how to frame the products to a customer in period t by sampling a framing assignment ϕ from a probability distribution $\rho^t(\phi) : \Omega_\phi(\boldsymbol{Q}) \to [0,1]$ over the set of all possible framing assignments. The distribution $\rho^t(\cdot)$ is constructed from the solution $\boldsymbol{x}^{t,D}$ in such a way that the expected total revenues and demands under RF are approximately close to those in the deterministic relaxation. Specifically, define: $v_i(\phi) := \sum_{r\in[R]} v_{ir} \cdot \mathbf{1}\{\phi(i) = r\}$, $r_i(\phi) := \sum_{r\in[R]} v_{ir} p_{ir} \cdot \mathbf{1}\{\phi(i) = r\}$, $\theta_i(\phi) := \lambda \cdot \frac{v_i(\phi)}{1+\sum_{k\in[I]} v_k(\phi)}$, and $\mathcal{R}_i(\phi) := \lambda \cdot \frac{r_i(\phi)}{1+\sum_{k\in[I]} v_k(\phi)}$. The constructed distribution $\rho^t(\cdot)$ satisfies the following conditions:

$$\sum_{\phi^t \in \Omega(\boldsymbol{Q})} \rho^t(\phi^t) \;=\; 1, \tag{7a}$$

$$\sum_{\phi^t \in \Omega(\boldsymbol{Q})} \rho^t(\phi^t) \cdot \theta_i^t(\phi^t) \;=\; \theta_i^t(\boldsymbol{x}^{t,D}) \qquad \forall\, i, \tag{7b}$$

$$\sum_{\phi \in \Omega(\boldsymbol{Q})} \rho^t(\phi^t) \cdot \mathcal{R}_i^t(\phi^t) \;=\; \mathcal{R}_i^t(\boldsymbol{x}^{t,D}) \qquad \forall\, i. \tag{7c}$$

Condition (7a) is important to guarantee that $\rho^t(\cdot)$ is a valid probability distribution, whereas conditions (7b) and (7c) are crucial for making sure that conditions (6c) and (6d) in Section 3.3 are satisfied. By definition of $\Omega(\boldsymbol{Q})$, condition (6b) is immediately satisfied by $\rho^t(\cdot)$.

A detailed discussion on the construction $\rho^t(\cdot)$ is provided in subsection 4.2. For now, we discuss RF policy and its performance. Let C_i^t denote the inventory level of product i at the beginning of period t. The complete description of RF is given below.

Randomized Framing (RF) Policy

1. Initialization:
   - (i) Solve **DET** to get $\boldsymbol{x}^{t,D}$ and construct $\rho^t(\cdot)$;
   - (ii) Set $C_i^1 = C_i$ for all i.
2. For $t = 1$ to T, do:
   - (i) Sample framing assignment ϕ^t using $\rho^t(\cdot)$;
   - (ii) Construct $\tilde{\phi}^t$ as follows: For each i, set $\tilde{\phi}^t(i) = R \cdot \mathbf{1}\{C_i^t = 0\} + \phi^t(i) \cdot \mathbf{1}\{C_i^t > 0\}$;
   - (iii) Apply modified framing assignment $\tilde{\phi}^t$;
   - (iv) Observe the realized demand and update remaining inventory.

Note that the construction of the modified assignment $\tilde{\phi}^t$ is important for making sure that we do not show an out-of-stock product to a customer, consistent with the industry partner's practice.

REMARK 1. A key technical contribution of our work is in constructing a randomization scheme over framing decisions based on $\boldsymbol{x}^{t,D}$ that satisfies both cardinality constraints (almost surely) and non-linear constraints (i.e., (6c)–(6d)). Topaloglu (2013) and Jasin and Sinha (2015) also construct randomization schemes, although in different settings. Topaloglu (2013) studied the joint stocking and assortment problem under MNL, and showed that the fraction of customers purchasing a product can be used to directly compute a joint distribution over the set of assortment decisions. Jasin and Sinha (2015) studied the multi-item fulfillment problem, and showed how to use the marginal distribution over fulfillment assignment decisions of individual items to construct correlated coupling among multiple items. Our work is similar in spirit to both Topaloglu (2013) and Jasin and Sinha (2015) in that we also attempt to construct a joint distribution using limited information that captures a certain feature of the problem. However, in contrast to these papers, in our work, we need to construct a joint probability distributions that satisfy both cardinality constraints (almost surely) and non-linear constraints (in expectation). The presence of these constraints create a new challenge that requires a development of a new randomization approach. ■

In analyzing the performance of RF, we will focus on a so-called *asymptotic* analysis with large demand and large inventory. In the e-commerce setting, retailers typically carry at least hundreds to thousands of units of inventory for a significant fraction of her products. Thus, the consideration of an asymptotic setting for e-commerce application is easily justified. Specifically, we consider a problem instance where both the length of the selling season and the number of units of initial inventory are linearly scaled by a constant $\theta > 0$ while keeping all the other parameters unchanged (this is the standard asymptotic setting in the literature, e.g., Jasin 2014). That is, the length of the selling season is given by $T(\theta) = \theta \cdot T$ and the initial vector of inventories is given by $\boldsymbol{C}(\theta) = \theta \cdot \boldsymbol{C}$, and we will focus on the case where θ is large. Let $\mathcal{J}^{\pi}(\theta)$ denote the total expected revenues under $\pi \in \Pi$ given θ. Similarly, let $\mathcal{J}^{*}(\theta)$ and $\mathcal{J}^{D}(\theta)$ denote the total expected revenues under the optimal policy π^* and the optimal value of the deterministic relaxation **DET**, respectively, given θ. The following theorem tells us the performance of RF.

THEOREM 1. *There exists a constant $M > 0$ independent of θ such that, for all large θ, we have $0 \leq \mathcal{J}^{D}(\theta) - \mathcal{J}^{RF}(\theta) \leq M \cdot \sqrt{\theta}$.*

The constant M in Theorem 1 may depend on the values of all other parameters except θ. Since $\mathcal{J}^* \geq \mathcal{J}^{RF}$, Theorem 1 also implies $\mathcal{J}^{D}(\theta) - \mathcal{J}^{*}(\theta) \leq M \cdot \sqrt{\theta}$ (i.e., $\mathcal{J}^{D}(\theta)$ is a good approximation of $\mathcal{J}^{*}(\theta)$ for large θ). Most importantly, it tells us that the revenue loss percentage of RF with respect to the optimal policy converges to zero at the rate of $\theta^{-1/2}$. We state this as a corollary.

COROLLARY 1. *If $\mathcal{J}^{*}(1) > 0$, then we have $1 - \mathcal{J}^{RF}(\theta)/\mathcal{J}^{*}(\theta) = \mathcal{O}(\theta^{-1/2}) \to 0$ as $\theta \to \infty$.*

Although in practice θ is much smaller than ∞, the strength of Theorem 1 is in guaranteeing that RF is approximating optimal as θ grows large. How large θ should be before the performance of RF is sufficiently close to optimal is determined by other problem parameters. Thus, as with other research in the literature that focus on developing near-optimal policies, this type of question is best answered empirically using numerical experiments. We do this in Section 6.

4.2. Constructing the probability distribution

As noted in Section 3, the key challenge in developing the RF policy is constructing a distribution $\rho^t(\cdot)$ from $\boldsymbol{x}^{t,D}$ that simultaneously satisfies (6a)–(6d) or, equivalently, (7a)–(7c). We will do this through a novel two-step procedure. Given that the procedure is identical for all time periods, we neglect the superscript t for simplicity in Section 4.2 and 4.3.

The first step is to construct an auxiliary distribution $\tilde{\rho}(\cdot)$ that satisfies (6a)–(6b) or (7a). The mathematical tool that we use to accomplish this is matrix decomposition. We start with a definition that relates a framing decision to a matrix.

Definition 1 (Framing matrix). We call an $N \times R$ matrix $B = (B_{nr})_{n\in[N],r\in[R]}$ a *framing matrix* with respect to some integer-valued R-dimension vector $\tilde{\boldsymbol{Q}}$ if the following hold: (i) $\sum_{r\in[R]} B_{nr} = 1$ for all $n \in [N]$ and (ii) $\sum_{n\in[N]} B_{nr} = \tilde{Q}_r$ for all $r \in [R]$.

We say that a matrix is *integral* if its components are either 0 or 1, and *fractional* otherwise. Note that, by definition, a framing matrix could either be integral or fractional.

Matrix $X = (x^D_{ir})_{i\in[I],r\in[R]}$ constructed from the **DET** solution is not always a framing matrix. Although requirement (i) on the row sums is always met since $\boldsymbol{x}^D$ satisfies constraint (5b), requirement (ii) may be violated since the column sums may be fractional if the inequality constraints (5a) are not binding. However, we can append rows to X so that it becomes a framing matrix with integer column sums. Appending rows to X is equivalent to introducing *fictitious* products. As we will later show in Section 4.3, any fractional framing matrix can be decomposed as a weighted sum of integral framing matrices. We use this matrix decomposition to construct an auxiliary distribution $\tilde{\rho}(\cdot)$. This construction is illustrated in the following example.

Example 1. Suppose that we have 3 products (i.e., $I = 3$) and at most 2 products can be displayed (i.e., $Q_1 = 2$ and $Q_2 = 3$). Suppose that after solving **DET**, we have:

$$X = (x^D_{ir})_{i\in[I],r\in[R]} = \begin{pmatrix} 1 & 0 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{pmatrix}.$$

The row sums of X are equal to 1, but the column sums are fractional (i.e., 1.7 and 1.3). Appending the row (0.3 0.7), corresponding to a fictitious product, results in a framing matrix with all column sums now equal to 2. The framing matrix could be decomposed as follows:

$$\begin{pmatrix} 1 & 0 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \\ 0.3 & 0.7 \end{pmatrix} = 0.5 \times \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} + 0.3 \times \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} + 0.2 \times \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

The corresponding sampling distribution $\tilde{\rho}$ can be described as follow: Display only products 1 and 2 with probability 0.5, display only product 1 with probability 0.3, and display only products 1 and 3 with probability 0.2. ■

The matrix decomposition is done using a subroutine, which we will introduce later in Section 4.3. This subroutine requires a framing matrix as its input, which is why introducing a fictitious product is necessary. Each integral framing matrix in the decomposition is a feasible framing assignment for the I products. This is later formalized in Lemma 2. The assignment of fictitious products are ultimately disregarded by the algorithm since assigning a fictitious product to a group r means that we leave one position in group r *empty*.

The second step in the two-step procedure applies a re-scaling procedure to $\tilde{\rho}(\cdot)$ to compute $\rho(\cdot)$. As we will formalize later in Lemma 3, this step is necessary for the distribution to satisfy conditions (6c)–(6d) or, equivalently, (7b)–(7c). Below, we provide the outline of our two-step procedure.

ConstrDist

C1. ***Framing Decomposition***

(i) Define $q_r = \sum_{i\in[I]} x^D_{ir}$ and $\tilde{Q}_r := \lceil q_r \rceil$ for each $r \in [R]$, and $\tilde{I} = \sum_{r\in[R]} \lceil q_r \rceil - I$.

(ii) Construct an $(I+\tilde{I}) \times R$ framing matrix $B = (B_{ir})$ with

$$B_{ir} := x^D_{ir} \quad \text{for all } 1 \le i \le I,\, r \in [R],$$
$$B_{ir} := \frac{\lceil q_r \rceil - q_r}{\tilde{I}} \quad \text{for all } I+1 \le i \le I+\tilde{I},\, r \in [R].$$

(iii) Decompose B into a sum of integral framing matrices with respect to $\tilde{\boldsymbol{Q}}$, i.e., $B = \alpha_1 B^{(1)} + \cdots + \alpha_L B^{(L)}$, where $\alpha_\ell \in (0,1]$ for all $\ell \in [L]$ and $\sum_{\ell=1}^{L} \alpha_\ell = 1$.

(iv) For each $\ell \in [L]$, define $\phi^{(\ell)}(\cdot)$ to be a framing assignment such that $\phi^{(\ell)}(i) = r$ iff $B^{(\ell)}_{ir} = 1$

C2. ***Constructing an Auxiliary Distribution*** $\tilde{\rho}$

(i) Define $\tilde{\Omega}(\boldsymbol{Q}) := \{\phi^{(\ell)} : \ell \in [L]\}$.

(ii) For each $\ell \in [L]$, define: $\tilde{\rho}(\phi^{(\ell)}) := \alpha_\ell$; For each $\phi \notin \tilde{\Omega}(\boldsymbol{Q})$, set $\tilde{\rho}(\phi) := 0$.

C3. ***Constructing*** ρ ***via Re-scaling***

For each $\phi \in \Omega(\boldsymbol{Q})$, define:

$$\rho(\phi) := \tilde{\rho}(\phi) \cdot \frac{1 + \sum_{i \in [I]} v_i(\phi)}{1 + \sum_{\phi' \in \Omega(\boldsymbol{Q})} \tilde{\rho}(\phi') \cdot \left[\sum_{i \in [I]} v_i(\phi')\right]} \,;$$

Define $\rho(\phi) = 0$ for each $\phi \notin \Omega(\boldsymbol{Q})$.

Steps C1(i)-(ii) of the algorithm aim at constructing a fractional framing matrix B by introducing *fictitious* products $I+1, \ldots, I+\tilde{I}$. By introducing fictitious rows to X, we can augment the column sums such that they are all integral. The resulting matrix B is a valid framing matrix (with $N = I + \tilde{I}$) with respect to an integral vector $\tilde{\boldsymbol{Q}}$. Step C1(iii) then decomposes B as a weighted sum of integral framing matrices using the subroutine in Section 4.3.

In general, **ConstrDist** is guaranteed to produce a feasible sampling distribution $\tilde{\rho}$ over a set of feasible framing decisions. We state a lemma.

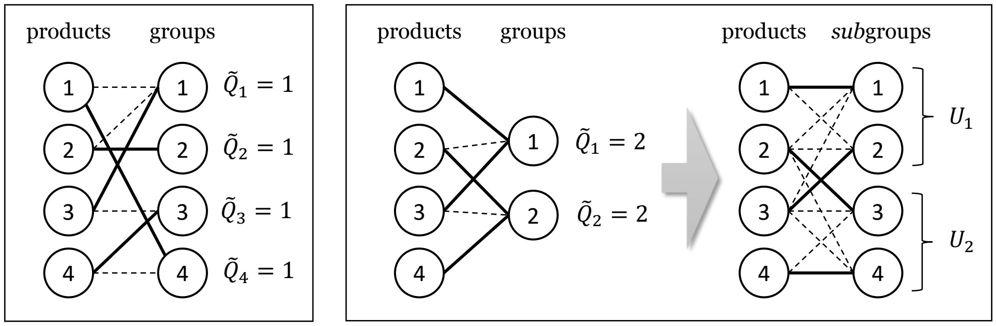
LEMMA 2. *Matrix B constructed in C1(ii) is a framing matrix with respect to $\tilde{\mathbf{Q}}$. Moreover, for every integral framing matrix $B^{(\ell)}$ with respect to $\tilde{\mathbf{Q}}$, a unique and feasible framing assignment ϕ can be found by setting $\mathbf{1}\{\phi(i) = r\} = B^{(\ell)}_{ir}$.*

It is not difficult to show that the sampling distribution $\tilde{\rho}(\cdot)$ produced in Step C2 satisfies the following conditions: $\sum_{\phi \in \Omega(\boldsymbol{Q})} \tilde{\rho}(\phi) = 1$ and $\sum_{\phi \in \Omega(\boldsymbol{Q})} \tilde{\rho}(\phi) \cdot \mathbf{1}\{\phi(i) = r\} = x^D_{ir}$, $\forall\, i, r$. In other words, not only is $\tilde{\rho}(\cdot)$ a valid joint probability distribution, it also satisfies the following marginal distribution condition: the probability of displaying product i in group r is exactly x^D_{ir}. Unfortunately, neither condition (7b) nor condition (7c) is necessarily satisfied by $\tilde{\rho}(\cdot)$ and this stems from the non-linearity of $\theta_i(\cdot)$ and $\mathcal{R}_i(\cdot)$. Interestingly, it is possible to modify $\tilde{\rho}(\cdot)$ via a re-scaling procedure given in C3 to construct the joint probability distribution $\rho(\cdot)$ that satisfies conditions (7b)-(7c). (In a nutshell, the re-scaled distribution ρ ensures that the relative proportions of the MNL model, due to the substitution effects, are respected. Note that, under re-scaling, the marginal distribution condition can be violated.) Since re-scaling only changes the value of the probability mass, the framing decisions sampled by $\rho(\cdot)$ are feasible. Moreover, the re-scaling argument leverages the fact that the choice probabilities for different products under MNL model has a common denominator, and the numerator only depends on the perceived utility of the corresponding product. We state this formally as a lemma below.

LEMMA 3. *The constructed distribution $\rho(\cdot)$ in* **ConstrDist** *satisfies conditions (7a)-(7c).*

4.3. The framing decomposition algorithm

We next discuss how to decompose a fractional framing matrix B as a weighted sum of integral framing matrices. Recall that RF uses the probability distribution $\rho(\cdot)$ to sample from the set $\Omega(\boldsymbol{Q})$



(a) $I + \tilde{I} = R$ (b) $I + \tilde{I} > R$

Figure 2 Graph representation of fractional framing matrix $\boldsymbol{B}$. The set of solid edges in a products-groups graph corresponds to an integral framing matrix with respect to $\tilde{\boldsymbol{Q}}$. (a) When $\boldsymbol{I} + \tilde{\boldsymbol{I}} = \boldsymbol{R}$, an integral framing matrix is associated with a perfect matching. (b) When $\boldsymbol{I} + \tilde{\boldsymbol{I}} > \boldsymbol{R}$, an integral framing matrix is a perfect matching in an equivalent products-subgroups graph.

of all feasible framing decisions. The set $\Omega(\boldsymbol{Q})$ has cardinality R^I. Therefore, an important consideration is that $\rho(\cdot)$ must have a small support, since sampling from a large-cardinality set is known to be computationally challenging. We accomplish this by proposing a framing decomposition algorithm (**FrameDecomp**) that decomposes B into a weighted sum of L integer framing matrices, where L is a small number. Our algorithm is motivated by some existing matrix decomposition algorithms in the literature.

The algorithm operates on a *weighted bipartite graph* associated with the framing matrix B (see Figure 2a). One node set is composed of all $I + \tilde{I}$ products, and the second node set is comprised of all R groups. There is an edge (i, r) if $B_{ir} > 0$, and B_{ir} is the edge weight. Finding an integral framing matrix in the decomposition is equivalent to choosing a set of edges where (i) each product node is adjacent to exactly one group node, and (ii) each group node r is adjacent to exactly $\tilde{Q}_r$ product nodes. For example, the solid edges of Figure 2a and the product-group graph in Figure 2b correspond to integral framing matrices. **FrameDecomp** utilizes graph-theoretic algorithms to efficiently find the set of edges satisfying these two requirements. The decomposition algorithm proceeds iteratively. Each iteration finds a new set of edges (i.e., an integral framing matrix A), and a multiplier α where the update $B \leftarrow B - \alpha A$ results in at least one entry of the matrix zeroed out (i.e., at least one edge in the graph is dropped). It does this until no edges are left in the graph. The decomposition of B is formed from this sequence of multipliers and integral framing matrices.

How do we know that a framing decomposition of B exists? And if it does, can we guarantee that our algorithm finds a decomposition? We address these important questions for the special case where the number of products $I + \tilde{I}$ is equal to the number of groups R. In this case, B is a doubly stochastic matrix (i.e., a square matrix with nonnegative entries where each row and each column sums to 1), and an integral framing matrix is a permutation matrix (i.e., exactly one nonzero entry per column and per row) that corresponds to a *perfect matching* in the graph (e.g. Figure 2a). Birkhoff's Theorem (Birkhoff 1946, Von Neumann 1953) states that a doubly stochastic matrix can always be decomposed into a weighted sum of permutation matrices, known as a Birkhoff-von Neumann (BvN) decomposition. Our algorithm will always find a decomposition since there will always exist a permutation matrix in each iteration. This is because the row and column sums of matrix B are decreased by the same amount α in every iteration.

In general, the number of groups R can be strictly smaller than the number of products $I+\tilde{I}$, in which case B is not doubly stochastic. Therefore, the first step in **FrameDecomp** is to convert the framing matrix B to a doubly stochastic matrix $B^\dagger$. It does this by creating $\tilde{Q}_r$ *subgroups* of group r, then distributing the value of B_{ir} uniformly across the entries $(B^\dagger_{iu})_{u\in U_r}$, where U_r are the subgroups of r (see Figure 2b). This transformation is motivated by the fact that an assignment of product i to group r (whose cardinality is $\tilde{Q}_r$) can be viewed as an assignment of product i to one of the $\tilde{Q}_r$ subgroups of group r (the cardinality of each subgroup is 1). Since $\sum_{r\in[R]}\tilde{Q}_r = I+\tilde{I}$ and B is a framing matrix, it is not difficult to see that $B^\dagger$ is a doubly stochastic matrix.

Converting B to a doubly stochastic matrix is useful since it allows us to translate any framing matrix to a permutation matrix, then utilize existing methods in matrix decomposition literature that efficiently find perfect matchings in a graph (see e.g., Gandhi et al. 2006, Dufossé and Uçar 2016, 2018 and Kulkarni et al. 2017). Ideally, we would like a decomposition with the smallest number L of permutation matrices. Unfortunately, the task of finding this decomposition is NP-complete (Dufossé and Uçar 2016). Dufossé and Uçar (2016) propose an algorithm resulting in an empirically small L. They do this by ensuring that, in each iteration, the set of edges chosen correspond to a *bottleneck matching*. (A bottleneck matching is the perfect matching whose smallest edge weight is the largest, hence the multiplier α is the largest possible.) A bottleneck matching can be found in polynomial time using a method by Duff and Koster 2001. By choosing the bottleneck matching in each iteration, the termination criteria (i.e., no edges in the graph) will be met after only a small number of iterations.

FrameDecomp implements the iterative algorithm of Dufossé and Uçar (2016) on $B^\dagger$ to find a BvN decomposition with a small L. In fact, any BvN decomposition algorithm can be used. Once a permutation matrix A is found in each iteration, the integral framing matrix can be recovered by setting the (i,r) entry to be equal to $\sum_{u\in U_r} A_{iu}$. Note that since each group can have multiple subgroups, multiple perfect matchings can map to the same framing matrix. Hence, if a framing matrix has been found in a previous iteration, **FrameDecomp** will simply update the weight for that framing matrix.

The framing decomposition algorithm is formally outlined below. (In the definition below, $A\odot B$ denotes the entry-wise product of two matrices A and B.)

FrameDecomp

Input: B, a proper framing matrix with respect to $\tilde{\boldsymbol{Q}}$

Output: A decomposition $B = \sum_{\ell \in [L]} \alpha^\ell \cdot B^{(\ell)}$

D1. i. Define the set of subgroups $\mathcal{U} = [I + \tilde{I}] = \cup_{r=1}^R U_r$, where each $r \in [R]$

$$U_r := \left\{ 1 + \sum_{v=1}^{r-1} \tilde{Q}_v, \ldots, \tilde{Q}_r + \sum_{v=1}^{r-1} \tilde{Q}_v \right\}.$$

ii. Construct an associated mapping $g : \mathcal{U} \mapsto [R]$, where $g(u) = r$ iff $u \in U_r$.

iii. Construct the $(I + \tilde{I}) \times (I + \tilde{I})$ doubly stochastic matrix $B^\dagger = (B^\dagger_{iu})_{i \in [I], u \in \mathcal{U}}$ by setting $B^\dagger_{iu} = B_{i,g(u)}/\tilde{Q}_{g(u)}$.

Define a weighted bipartite graph $G_{B^\dagger} = (\mathcal{I} \cup \mathcal{U}, E)$ associated with $B^\dagger$.

D2. Initialize $L \leftarrow 0$ (the number of unique integral framing matrices found).

While $B^\dagger$ is not a zero matrix, do:

i. Find a bottleneck matching M in $G_{B^\dagger}$.

ii. Construct the permutation matrix A_M associated with matching M;

iii. Set $\alpha \leftarrow \min\{A_M \odot B^\dagger\}$, i.e. the minimum edge weight of M;

iv. Construct a integral framing matrix $\tilde{B}$ by setting $\tilde{B}_{ir} = \mathbf{1}\{(i, g^{-1}(r)) \in M\}$;

v. If $\tilde{B} = B^{(\ell^*)}$ for some $\ell^* \in [L]$, then update weight $\alpha^\ell \leftarrow \alpha^\ell + \alpha$;

Otherwise, update $L \leftarrow L + 1$, and set $\alpha^L = \alpha$, $B^{(L)} = \tilde{B}$;

vi. Update $B^\dagger \leftarrow B^\dagger - \alpha A_M$.

Improving the decomposition algorithm. Translating the products-groups graph G to an equivalent products-*sub*groups graph $G^\dagger$ typically requires a significant increase in the number of edges (see Figure 2b). This is because each edge between a product and a group r in G must be copied Q_r times in $G^\dagger$. In algorithm **FrameDecomp**, each iteration of the *while* loop finds a bottleneck matching in $G^\dagger$, and at least one edge of this matching is removed from the graph $G^\dagger$. However, this typically does not correspond to removing an edge in the graph G. This is because an edge between a product and a group r is removed in G if and only if *all* Q_r edges adjacent to this product are removed in $G^\dagger$. Hence, if the sequence of bottleneck matchings correspond to very dissimilar product-to-group assignments, it may take many iterations before all edges in the G graph are removed (i.e., the termination criteria).

It is possible to modify **FrameDecomp** to find an even smaller L. The key is to ensure that, in each iteration, instead of choosing a bottleneck matching, we choose a perfect matching that has the least number of reassigned products compared to the matching in the last iteration. Given two perfect matchings, we say that a product is *reassigned* if its adjacent subgroup in the two matchings correspond to two distinct groups. If there are multiple perfect matchings that match this criteria,

Instance	Num. products	Displayed cardinality	Num. framing L		
	I^1	$(Q_1, Q_2)^2$	All[3]	Original[4]	Improved[5]
1	19	(15, 4)	3,876	165	18
2	16	(12, 4)	1,820	132	17
3	12	(8, 4)	495	72	13
4	11	(7, 4)	330	62	14
5	12	(9, 3)	220	33	18
6	10	(6, 4)	210	31	9
7	9	(4, 5)	126	30	10
8	9	(5, 4)	126	21	9
9	13	(11, 2)	78	45	11
10	8	(5, 3)	56	15	7

Table 1 **FrameDecomp** and improved **FrameDecomp** on ten fractional framing matrices.

[1] Number of rows of B^j after rows are removed (the number of columns is $R=2$).
[2] Cardinality constraint on the two framing groups.
[3] Cardinality of $\Omega(\boldsymbol{Q})$.
[4] Cardinality of the support (i.e., L) under **FrameDecomp**.
[5] Cardinality of the support (i.e., L) under the improved **FrameDecomp**.

we choose the one where the smallest edge weight is the largest. In Section EC 6, we introduce a polynomial-time algorithm that can find such a matching using augmenting paths. By choosing perfect matchings with minimum reassignments between iterations, the modified algorithm is forced to choose the same set of edges in graph G in consecutive iterations until at least one edge is removed in G. Therefore, the number of matrices in the framing decomposition is smaller than that found by the unmodified algorithm.

We want to point out that both **FrameDecomp** and the improved **FrameDecomp** are computationally efficient. More importantly, the resulting number of permutation matrices under either algorithm is relatively small (at most $4I^2$). In computational experiments (see Table 1 discussed later), this upper bound is loose. We state this formally below.

LEMMA 4. **FrameDecomp** *and the improved* **FrameDecomp** *have polynomial running time. Moreover, the number of unique integral framing matrices L in either algorithm is at most $4I^2$.*

We compare the two decomposition algorithms in numerical experiments. Note that the only difference between the two algorithms is Step D2(i), where the original algorithm chooses a bottleneck matching, while the improved algorithm chooses a perfect matching with minimum reassignments. The choice of model parameters are described in Section 6. Specifically, we have $I=56$ and $R=2$ for $J=98$ instances of the framing problem. For each instance $j \in [J]$, the constructed fractional $(I+\tilde{I}) \times R$ matrix $B^j = (B^j_{ir})$ in step D1 is often sparse. Therefore, to speed up both algorithms, we drop products with integral rows (since those framings are fixed) and adjust I accordingly, B^j, and $\tilde{\boldsymbol{Q}}$ appropriately. As a result, we apply both algorithms on a less sparse matrix B^j. Table 1 summarizes the results for the 10 instances with the most number of feasible framing assignments in $\Omega(\boldsymbol{Q})$. We observe that the improved **FrameDecomp** results in a significantly sparser framing decomposition compared to the original heuristic **FrameDecomp**. In fact, the decomposition size can be an order of magnitude smaller in some cases.

5. Joint Framing and Fulfillment Problem (JFF)

Thus far, we have studied a base model of product framing under inventory constraints where the objective is to maximize the expected revenue. In practice, inventory is distributed in a network of fulfillment centers (FCs). When demand for a product is realized, the cost for fulfilling the demand is increasing in the distance of the customer to the fulfilling FC. Hence, to maximize the total expected profit, the product framing decisions must also take into consideration how orders are fulfilled. We do this by incorporating fulfillment decisions into our base model, turning our attention to the Joint Framing and Fulfillment (JFF) problem. We start this section with problem formulations and then we discuss our heuristic policy.

We assume that customers are distributed across J regions and the online retailer has K fulfillment centers (FCs). We use the binary variables X_{ijr}^t to denote the framing decision (i.e., $X_{ijr}^t = 1$ means that, in period t, the online retailer assigns product i to framing option r for a customer from region j) and Y_{ijk}^t to denote the fulfillment decision (i.e., $Y_{ijk}^t = 1$ means that, in period t, the online retailer uses FC k to fulfill an order for product i from a customer from region j). At the beginning of period t, a new customer arrives from region j with probability $\lambda_j \in [0,1]$. Let $\boldsymbol{X}_j^t = (X_{ijr}^t)_{i\in[I],r\in[R]}$. The purchase probability of product i by a customer in region j under $\boldsymbol{X}_j^t$ is given by $\theta_{ij}(\boldsymbol{X}_j^t) = \lambda_j \cdot \frac{\sum_{r\in[R]} v_{ijr}\cdot X_{ijr}^t}{v_{0j}+\sum_{\ell\in[I],r\in[R]} v_{\ell jr}\cdot X_{\ell jr}^t}$ (we will simply assume $v_{0j}=1$ for all j), and the expected revenue generated by product i from a customer in region j under $\boldsymbol{X}_j^t$ is given by $\mathcal{R}_{ij}(\boldsymbol{X}_j^t) = \lambda_j \cdot \frac{\sum_{r\in[R]} v_{ijr}\cdot X_{ijr}^t\cdot p_{ir}}{1+\sum_{\ell\in[I],r\in[R]} v_{\ell jr}\cdot X_{\ell jr}^t}$. Let c_{ijk} denote the cost of shipping (fulfilling) product i from FC k to region j and let $\boldsymbol{C}_k = (C_{ik})_{i\in[I]}$, where $C_{ik} \geq 0$, denote the vector of initial inventory at FC k at the beginning of period 1. We can express the optimal control formulation of our joint framing and fulfillment problem as

$$
\begin{aligned}
\textbf{(OPT'):}\quad \mathcal{J}^* = \max_{\pi\in\Pi} \quad & \sum_{t\in[T],i\in[I],j\in[J]} \mathbb{E}\left[\mathcal{R}_{ij}(\boldsymbol{X}_j^{t,\pi}) - \sum_{k\in[K]} c_{ijk}\cdot Y_{ijk}^{t,\pi}\right] \\
\text{s.t.} \quad & \sum_{i\in[I]} X_{ijr}^{t,\pi} \leq Q_r \qquad \forall\, j, r, t \qquad \text{(8a)} \\
& \sum_{r\in[R]} X_{ijr}^{t,\pi} = 1 \qquad \forall\, i, j, t, \qquad \text{(8b)} \\
& \sum_{k\in\{0\}\cup[K]} Y_{ijk}^{t,\pi} = D_{ij}^t(\boldsymbol{X}_j^{t,\pi}) \qquad \forall\, i, j, t, \qquad \text{(8c)} \\
& \sum_{t\in[T],j\in[J]} Y_{ijk}^{t,\pi} \leq C_{ik} \qquad \forall\, i, k, \qquad \text{(8d)} \\
& X_{ijr}^{t,\pi}, Y_{ijk}^{t,\pi} \in \{0,1\} \qquad \forall\, i, j, k, r, t, \qquad \text{(8e)}
\end{aligned}
$$

where all the constraints must hold almost surely. Note that constraints (8c) and (8d) guarantee that each order will be fulfilled from one of the FCs and that we do not ship more than the available inventory from any FC, respectively. Let $\boldsymbol{c}_j = (c_{ijk})_{i\in[I],k\in[K]}$, $\boldsymbol{C} = (C_{ik})_{i\in[I],k\in[K]}$, and $\boldsymbol{D}_j^t(\boldsymbol{X}^t) = (D_{ij}^t(\boldsymbol{X}^t))_{i\in[I]}$. Also, let $\boldsymbol{Y}_j^t = (Y_{ijk}^t)_{i\in[I],k\in[K]}$. Similar to the formulation in Section 3, we can write **OPT'** compactly as follows:

$$
\begin{aligned}
\mathcal{J}^* = \max_{\pi\in\Pi} & \sum_{t\in[T],i\in[I],j\in[J]} \mathbb{E}\left[\mathcal{R}_{ij}(\boldsymbol{X}_j^{t,\pi})\right] - \sum_{t\in[T],j\in[J]} \mathbb{E}\left[\boldsymbol{c}_j\cdot \boldsymbol{Y}_j^{t,\pi}\right] \\
\text{s.t.} \quad & A_1\boldsymbol{X}_j^{t,\pi} \le \boldsymbol{Q} && \forall j,t, && \text{(9a)}\\
& A_2\boldsymbol{X}_j^{t,\pi} = \boldsymbol{1} && \forall j,t, && \text{(9b)}\\
& A_3\boldsymbol{Y}_j^{t,\pi} = \boldsymbol{D}_j^t(\boldsymbol{X}^{t,\pi}) && \forall j,t, && \text{(9c)}\\
& \sum_{t\in[T],j\in[J]} \boldsymbol{Y}_j^{t,\pi} \le \boldsymbol{C} && && \text{(9d)}\\
& \boldsymbol{X}_j^{t,\pi}\in\{0,1\}^{IR},\ \boldsymbol{Y}_j^{t,\pi}\in\{0,1\}^{IK} && \forall j,t, && \text{(9e)}
\end{aligned}
$$

for some coefficient matrices A_1, A_2, and A_3. Let $\boldsymbol{x}_j = (x_{ijr})_{i\in[I],r\in[R]}$, $\boldsymbol{y}_j = (y_{ijk})_{i\in[I],k\in[K]}$, and $\boldsymbol{\theta}_j(\boldsymbol{x}_j) = (\theta_{ij}(\boldsymbol{x}_j))_{i\in[I]}$. The deterministic counterpart of **OPT'** is

$$
\begin{aligned}
\textbf{(DET')}: \quad \mathcal{J}^D = \max_{\{\boldsymbol{x}_j^t,\boldsymbol{y}_j^t\}} & \sum_{t\in[T],i\in[I],j\in[J]} \mathcal{R}_{ij}(\boldsymbol{x}_j^t) - \sum_{t\in[T],j\in[J]} \boldsymbol{c}_j\cdot\boldsymbol{y}_j^t \\
\text{s.t.} \quad & A_1\boldsymbol{x}_j^t \le \boldsymbol{Q} && \forall j,t && \text{(10a)}\\
& A_2\boldsymbol{x}_j^t = \boldsymbol{1} && \forall j,t && \text{(10b)}\\
& A_3\boldsymbol{y}_j^t = \boldsymbol{\theta}_j(\boldsymbol{x}_j^t) && \forall j,t && \text{(10c)}\\
& \sum_{t\in[T],j\in[J]} \boldsymbol{y}_j^t \le \boldsymbol{C} && && \text{(10d)}\\
& \boldsymbol{x}_j^t\in[0,1]^{IR},\ \boldsymbol{y}_j^t\in[0,1]^{IK} && \forall j,t && \text{(10e)}
\end{aligned}
$$

Let $(\boldsymbol{x}^{t,D}, \boldsymbol{y}^{t,D})$ denote the optimal solution to $\mathcal{J}^D$. The following result is similar to Lemma 1.

LEMMA 5. $\mathcal{J}^* \le \mathcal{J}^D$

We now discuss our heuristic policy, which we call *Randomized Framing and Fulfillment* (RFF) policy. The key idea behind RFF is to decouple the framing decision from the fulfillment decision. Specifically, for each region j, we first construct a probability distribution $\rho_j(\cdot)$ over the set of all feasible framing assignments using **ConstrDist** in Section 4 with the deterministic solution $\boldsymbol{x}_j^{t,D}$ as its input. Next, the fulfillment decision Y_{ijk}^t is decided purely based on the values of $\boldsymbol{y}_j^{t,D}$. Let $\boldsymbol{\sigma}_j^t := (\sigma_j^t(1),\ldots,\sigma_j^t(I))$ denote the fulfillment assignment for region j at period t (i.e., $\sigma_j^t(i) = k$ means that, at period t, we fulfill an order of product i from region j from FC k), and let C_{ik}^t denote the inventory level of product i in FC k at the beginning of period t. Also, let $j(t)$, $i(t)$, and

$k(t)$ denote the region from which the new customer arrives at period t, the product purchased by the customer, and the initial FC decision for fulfilling the order, respectively. (If there is neither a new arrival nor a new purchase at period t, we simply set $j(t) = \emptyset$ and $i(t) = \emptyset$, respectively.)

The complete description of RFF is given below.

Randomized Framing and Fulfillment (RFF)

1. Initialization:
   (i) Solve **DET** to get $\{\boldsymbol{x}^{t,D}, \boldsymbol{y}^{t,D}\}$ and construct $\rho_j^t(\cdot)$ for all j with **ConstrDist**;
   (ii) Set $C_{ik}^1 = C_{ik}$ for all i and k.
2. For $t = 1$ to T, do:
   (i) If $j(t) = \emptyset$, move to the next period; otherwise, go to the next step;
   (ii) Sample framing assignment $\phi_{j(t)}^t$ using $\rho_{j(t)}^t(\cdot)$;
   (iii) For each i, set $\tilde{\phi}_{j(t)}^t(i) = R \cdot \mathbf{1}\{\max_k C_{ik}^t = 0\} + \phi_{j(t)}^t(i) \cdot \mathbf{1}\{\max_k C_{ik}^t > 0\}$;
   (iv) Apply framing assignment $\tilde{\phi}_{j(t)}^t$;
   (v) If $i(t) = \emptyset$, move to the next period; otherwise go to the next step;
   (vi) Sample fulfillment decision $\sigma_{j(t)}^t(i(t)) = k := k(t)$ with probability $\tilde{y}_{i(t)jk}$ where
$$\tilde{y}_{ijk} := \frac{y_{ijk}^{t,D}}{\sum_{k' \in [K]} y_{ijk'}^{t,D}} \qquad \text{for all } i, j, k;$$
   (vii) If $C_{i(t)k(t)}^t > 0$, fulfill from FC $k(t)$ and update inventory; else, fulfill from the next available cheapest FC and update inventory.

Similar to our result in Theorem 1, it can be shown that RFF is asymptotically optimal.

THEOREM 2. *There exists $M > 0$ independent of θ such that, for all large θ, we have $0 \leq \mathcal{J}^D(\theta) - \mathcal{J}^{RFF}(\theta) \leq M \cdot \sqrt{\theta}$. Moreover, if $\mathcal{J}^*(1) > 0$, then $1 - \mathcal{J}^{RFF}(\theta)/\mathcal{J}^*(\theta) = \mathcal{O}(\theta^{-1/2}) \to 0$ as $\theta \to \infty$.*

6. Numerical Experiments

In this section, we test several framing and fulfillment heuristics (including our proposed RFF policy) in experiments that simulate the sequential arrival of e-commerce customers in a geographical network. We compare the performance of different heuristics based on their resulting expected total profits, expected total sales, expected revenue-per-sale, and expected cost-per-sale.

We designed the experimental setup based on the data provided by a major U.S. retailer. Specifically, the retailer provided inventory data of 56 products ($I = 56$) in its 27 FCs ($K = 27$) at the start of an end-of-season clearance period. All products belong to the same category and can be offered in the clearance page in the seller's online store for only 12 weeks. Within the clearance page, some products can be prominently displayed in a special section (i.e., promoted). Thus, there are three framing options ($R = 3$): displayed but not promoted ($r = 1$), displayed and promoted

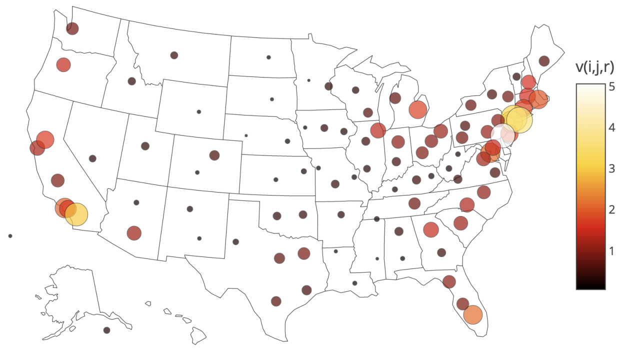


Figure 3 Demand map where bubbles are placed on the 98 customer location centers; the size of each bubble corresponds to the arrival rate, and the color corresponds to the MNL attraction coefficient v_{ijr} for product $i=53$ and group $r=1$ (displayed but not promoted).

($r=2$), and not displayed ($r=3$). In our experiments, we assume that at most 30 products can be promoted at any time.

We are provided with the estimates of MNL demand models for 98 customer regions ($J=98$) within the continental US. (The demand estimation procedure is beyond the scope of this paper and is not part of the research project; interested readers could refer to Vakhutinsky et al. (2019) or other empirical literature for standard approaches in estimating the parameters of an MNL model.) Figure 3 shows that there is geographic heterogeneity in the arrival rates (bubble size) and the MNL attraction coefficients of a specific product (bubble color). In the map, we fix a product ($i=53$) and its framing option ($r=1$) to illustrate the heterogeneity in its attractiveness. Additional details of the attraction coefficients can be found in Section O.1 in the Online Supplements (see Lei et al. 2021). Although the 56 products are in the same category, there is a wide range in their clearance prices. Figure 4 shows that prices range from as low as \$11.99 to as high as \$129.99 (product indices are assigned in decreasing order of their clearance prices).

It can also be seen from Figure 4 that some products start the clearance period with a high level of aggregate inventory, but notably most of the products have little remaining stock. Further investigation of the data reveals that more than 30% of the products have less than 100 units of inventory, and more than 60% have less than 500 units (see Figure O.1a in Lei et al. 2021). Even the product with the most inventory (product 7) is only in-stock at roughly half of the 27 fulfillment centers (see Figure O.1b in Lei et al. 2021). In fact, more than 40% of all products are in-stock in 5 FCs or less. Hence, there is an uneven distribution of inventory across the FCs in the network, which is a typical characteristic of clearance sales products. We estimate the shipping costs from an FC to a customer location according to the 2019 USPS Priority Mail business price rates for shipping a 3 lb. product. Therefore, the cost of fulfillment varies from \$8.30 to \$25.95.The average fulfillment cost among the origin-destination pairs is \$16.68.

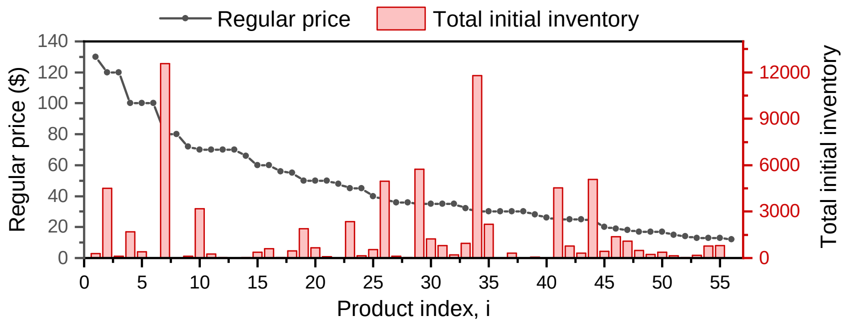


Figure 4 Regular price and initial inventory during clearance period in one product category.

It is clear from this data that maximizing the total expected revenues through framing decisions alone without considering the fulfillment decision can hurt the retailer's overall profits. The reason is that, if the retailer only maximizes its revenue, it will promote the high priced products. But, since these products have little inventory (Figure 4), this strategy would result in a high total shipping costs when only FCs that are distant to the customer location are in-stock. This problem is exacerbated by the wide range of fulfillment cost values (\$8.30 to \$25.95). We next set out to test this hypothesis by simulating customer arrivals and comparing RFF to heuristics that optimize the framing decision independently from the fulfillment. These framing heuristics are:

1. **Myopic-L**: When a customer arrives, it offers the product framing that maximizes the expected profit from that customer given that the sale will be fulfilled from the closest FC. Note that with this policy, a product will not be displayed to the customer if it is out-of-stock at the closest FC (i.e., locally), even if this product is available elsewhere. In our simulation, as soon as an FC stocks out of all products, it is dropped from the simulation. The "L" in the policy name refers to local inventory.
2. **Myopic-G**: When a customer arrives, it offers the product framing that maximizes the expected revenue from that customer, but only among the products with remaining inventory anywhere in the network. A product will not be displayed if it is out-of-stock in all FCs (i.e., globally). The "G" in the policy name refers to global inventory.
3. **Myopic-IB**: It implements an "inventory balancing" policy, which is an adaptation of the dynamic assortment policy of Golrezaei et al. (2014) to a dynamic framing problem. When a customer arrives, this policy offers the product framing that maximizes the expected *discounted* revenue from that customer. The discount factor applied to the revenue of product i is $\Psi(I_i/C_i)$, where I_i and C_i are the current and initial aggregate inventory of product i, respectively, and $\Psi : [0,1] \mapsto [0,1]$ is an increasing penalty function. In our simulations, we test the exponential penalty function $\Psi(x) = (e/(e-1))(1-e^{-x})$.

4. **RF**: It implements the Randomized Framing policy (Section 4), a framing heuristic that aims to maximize the total expected revenue under the constraint that total sales must not exceed the *aggregate inventory level* of each product (i.e. $C_i = \sum_{k\in[K]} C_{ik}$). Note that RF is forward-looking since it relies on a forecast of future demand. We test two versions of this policy: (i) the version where the framing probabilities are calibrated only once at the start of the clearance period, and (ii) the version where the probabilities are re-calibrated weekly (a total of 12 times) by re-optimizing **DET** with the most up-to-date inventory levels.

We couple the framing policies above with a fulfillment heuristic. The myopic framing policies (Myopic-L, Myopic-G, Myopic-IB) are paired with the myopic fulfillment policy that fulfills a customer sale from the closest in-stock FC. For the RF policy, we test both a myopic fulfillment and a LP-based policy. The LP-based policy solves a deterministic linear program that minimizes the total shipping costs (taking the expected demands from the framing policy as inputs), and randomly chooses an FC based on the probabilities corresponding to the LP solution. Note that the RFF policy (for the joint framing and fulfillment problem in Section 5) differs from RF with LP fulfillment in that the latter optimizes the framing while ignoring its impact on fulfillment.

With the LP-based policy, the fulfillment LP model is infeasible if the aggregate expected demand for a product exceeds its aggregate inventory. Hence, we introduce a dummy FC with infinite inventory of all products, and we set the shipping cost from this FC to be larger than any price or any other shipping cost. This dummy FC serves as a backup facility when a product is depleted at all real FCs and technically guarantees that there is always a feasible solution to the LP fulfillment model. In our simulations, fulfilling from the dummy FC is equivalent to no sale.

In order to test the different heuristics, we randomly generate 30 sample paths, where each sample path is a different sequence of customer arrivals. We also provide the 95% confidence intervals when presenting our results. To randomly generate the location of a customer arrival in a given period, we use the arrival probabilities that were provided by the retailer. For our base experiment, we set the total number of customer arrivals in a sample path to equal the total inventory across all products (i.e., $75,352$ units), and assume that each period has exactly one arrival (i.e., we scale up the arrival probabilities so that $\sum_{j\in[J]} \lambda_j = 1$).

On each sample path, we simulate the sequential arrival of customers. After a customer arrives from a location, she is presented with a display of products (determined by the policy); next, she makes a purchase decision based on the MNL choice model; and then, the order is fulfilled by the seller (determined by the policy, and only if there is inventory) before the inventory level is finally updated. After running the simulation on all arrivals of the sample path, we compute the following metrics: the total sales, the revenue-per-sale (total revenue divided by total sales), the

Table 2 Performance of policies: The table summarizes the profit gap, and the average (with 95% confidence intervals) of total sales, revenue-per-sale, cost-per-sale, and total profit.

Framing policy	Fulfill policy	Total sales[1] (units)	Revenue-per-sale ($)	Cost-per-sale ($)	Total profit ($ millions)	Profit gap
Myopic-L	Myopic	$52,858 \pm 73$	52.82 ± 0.04	8.37 ± 0.002	2.350 ± 0.003	20.30%
Myopic-G	Myopic	$61,409 \pm 61$	54.70 ± 0.03	10.86 ± 0.011	2.692 ± 0.001	8.87%
Myopic-IB	Myopic	$69,294 \pm 34$	50.43 ± 0.01	10.79 ± 0.009	2.747 ± 0.001	7.02%
RF	Myopic	$69,188 \pm 49$	51.93 ± 0.03	12.37 ± 0.013	2.737 ± 0.002	7.34%
RF	LP	$68,027 \pm 36$	51.96 ± 0.03	11.62 ± 0.007	2.744 ± 0.002	7.11%
RF (Reopt)	Myopic	$69,573 \pm 35$	51.88 ± 0.02	12.38 ± 0.013	2.749 ± 0.001	6.96%
RF (Reopt)	LP	$69,227 \pm 47$	51.91 ± 0.02	11.61 ± 0.012	2.790 ± 0.001	5.56%
RFF		$66,406 \pm 48$	52.48 ± 0.02	8.97 ± 0.003	2.889 ± 0.002	2.20%

[1] The number of arrivals is equal to the total inventory of 75,352 (i.e. load factor is 1).

cost-per-sale (total cost divided by total sales), and the total profit. Based on the 30 sample paths, we compute the averages of these metrics and the 95% confidence intervals. We also record the profit gap, $(\mathcal{J}^D - \bar{\mathcal{J}})/\mathcal{J}^D$, where $\bar{\mathcal{J}}$ is the sample average total profit, and $\mathcal{J}^D$ is the optimal value of **DET'**. Recall that $\mathcal{J}^D$ is an upper bound to the optimal expected profit.

Table 2 presents simulation results. We observe that the largest profit gap occurs under the myopic framing policies Myopic-L and Myopic-G. This is perhaps not too surprising since both policies ignore the effect of framing decisions on the availability of inventory for future customers. Comparing the two policies, Myopic-L has a significantly lower cost-per-sale since it guarantees that all products displayed to a customer are in-stock at the nearest FC. However, since Myopic-L only recommends products that are available in the nearest FC, it precludes the possibility of selling a higher priced (or more attractive) product that may be in-stock at another FC. Hence, Myopic-L has fewer total sales and a lower revenue-per-sale, resulting in a high profit gap.

Without forecasting future demand, the Myopic-IB policy mimics a forward-looking policy by maintaining an index that it uses as a proxy for the value of each unit of remaining inventory. Hence, among all myopic policies that we tested, Myopic-IB has the lowest profit gap. In fact, this profit gap is slightly lower than that of RF, even though the latter incorporates a forecast of future demand. Note that Myopic-IB computes the index based on current inventory levels, whereas the RF framing decisions ignore this information. If the RF policy is recalibrated weekly with current inventory levels (RF Reopt), its performance is notably superior to that of Myopic-IB. We also observe that the performance of Myopic-IB is significantly worse than that of the joint framing and fulfillment policy RFF. This implies that incorporating a demand forecast can provide significant value in a setting where framing and fulfillment decisions can be made jointly.

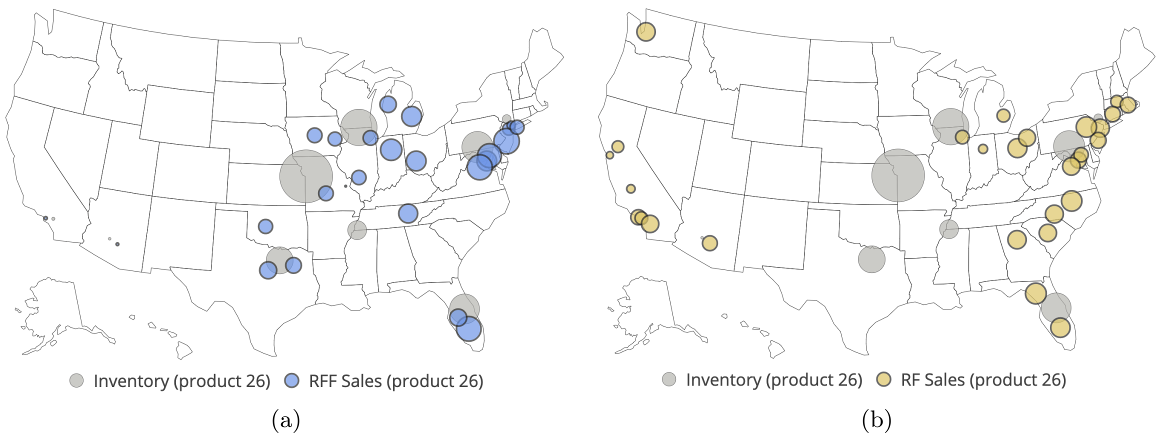


Figure 5 Location and amount (bubble size) of product 26 inventory and expected sales under RFF and RF.

An important observation from Table 2 is that, for the RF and RF (Reopt) policies, there is not a large difference in the cost-per-sale between myopic and LP-based fulfillments. In particular, fixing the framing policy and the expected demand, the LP-based fulfillment can only reduce the average cost-per-sale by 6%. From this, we can infer that any significant reduction in shipping costs does not come from making the fulfillment policy more efficient, but rather has to come from a better *demand management* that takes shipping costs into account. This is further supported by the observation that RFF—which optimizes framings by considering the fulfillment costs—reduces the average cost-per-sale by at least 23% compared to any of the RF policies. The superior performance of RFF is because it can better manage demand (by controlling product framing in each location) so that demands are generated at locations close to the FCs with available inventories. This is demonstrated in Figure 5 which compares the inventory locations of product 26 against the locations of its expected sales under RFF and under RF. In the simulations, only considering product 26, RFF and RF have similar average total sales (4,890 and 4,863, respectively) and average revenue-per-sale ($37.99 for both), but RFF has a significantly lower average cost-per-sale ($9.85) compared to RF ($12.74), which is a 23% cost reduction. This example highlights the value of jointly optimizing product framing and fulfillment decisions, especially when inventory is unevenly distributed and costs are high when shipping to long distances. Crucially, this value is realized because of a better demand management.

We also test the impact of other key problem parameters on the performances of all the policies, including number of display groups, load factor (i.e. the ratio of the total arrivals to the total inventory), total market size, and number of allowed changes to the displays. The results are presented in the Online Supplements (see Lei et al. 2021). Overall, our proposed heuristic exhibits strong and robust performance under a wide variety of system conditions.

7. Conclusion

Motivated by a product framing problem faced by a major U.S. retailer during its clearance sales period, in this paper, we consider a relatively general joint product framing and order fulfillment problem with cardinality and capacity constraints under the MNL model. We propose a randomized policy based on the idea of matrix decomposition. Using the data of a U.S. retailer, we show that our policy significantly reduces shipping costs by successfully managing demands such that they occur mostly at regions closest to the FCs with the most available inventories.

While our model and algorithms are applicable to a wide range of product framing problems, they have some limitations. Investigating how to address these limitations would be important directions for future research. We have focused our analysis on the MNL model. While the MNL model is very popular both in theory and in practice, it suffers from limitations such as the independence of irrelevant alternatives (IIA) assumption. Extending our randomized policy to more general choice models (e.g. nested logit, mixed logit) is not trivial, but it would be an impactful research direction.

In modeling the cardinality constraints in (3a), we have implicitly assumed that the retailer has pre-specified the disjoint sets of framing groups and that each product must be uniquely assigned to one group. In practice, different framing groups may overlap. For example, the retailer may have a local cardinality constraint on each page and a global constraint on the number of products to be promoted, irrespective of on which page the product is displayed. In this example, a product can be displayed in two groups simultaneously (e.g., promoted on the first page). Extending our algorithm to such setting would require a modification in the matrix decomposition algorithm.

In our current model, we have assumed that the initial inventory distribution among FC network is fixed. In reality, the retailer might have the opportunity to decide how to allocate the inventory among his network. Moreover, the retailer may also transship inventory across different FCs to relieve the imbalance in inventory distribution. Extending our model to incorporate periodic replenishment and/or transshipment would be another interesting direction.

Finally, it is also interesting to study the online retailer's optimization problem under the so-called click models instead of the classic choice models. Click models are often used in practice to study customers' click and search behavior in large-scale web analytic applications (see Chuklin et al. (2015) for an overview of the topic). And yet, despite their prevalence in industry, click models have not received as much attention as they should be in the operations community.

Acknowledgments

The authors thank Kamalini Ramdas (the Department Editor), the anonymous associate editor, and the two anonymous referees for their comments and suggestions that have improved this paper.

References

Abeliuk, A, G Berbeglia, M Cebrian, P Van Hentenryck. 2016. Assortment optimization under a multinomial logit model with position bias and social influence. *4OR* **14**(1) 57–75.

Acimovic, J, VF Farias. 2019. The fulfillment-optimization problem. *Operations Research & Management Science in the Age of Analytics*. INFORMS, 218–237.

Acimovic, J, SC Graves. 2014. Making better fulfillment decisions on the fly in an online retail environment. *M & SOM* **17**(1) 34–51.

Acimovic, J, SC Graves. 2017. Mitigating spillover in online retailing via replenishment. *M& SOM* **19**(3) 419–436.

Andrews, JM, VF Farias, AI Khojandi, CM Yan. 2019. Primal–dual algorithms for order fulfillment at urban outfitters, inc. *Interfaces* **49**(5) 355–370.

Aouad, A., D. Segev. 2020. Display optimization for vertically differentiated locations under multinomial logit preferences. *Manag. Sci.* .

Bernstein, F, AG Kök, L Xie. 2015. Dynamic assortment customization with limited inventories. *M & SOM* **17**(4) 538–553.

Birkhoff, G. 1946. Three observations on linear algebra. *Univ. Nac. Tacuman, Rev. Ser. A* **5** 147–151.

Caro, F, J Gallien. 2012. Clearance pricing optimization for a fast-fashion retailer. *Oper. Res.* **60**(6) 1404–1422.

Çetin, O, AJ Mersereau, AK Parlaktürk. 2019. Management and effects of in-store promotional displays. *M& SOM* .

Charnes, A, W W Cooper. 1962. Programming with linear fractional functionals. *Nav. Res. Logis. Q* **9**(3-4) 181–186.

Chen, J. 2014. Pricing strategies in the electronic marketplace. *Handbook of Strategic e-Business Management*. Springer, 489–522.

Chen, Q, S Jasin, I Duenyas. 2015. Real-time dynamic pricing with minimal and flexible price adjustment. *Manage. Sci.* **62**(8) 2437–2455.

Chuklin, A, I Markov, M Rijke. 2015. Click models for web search. *Synthesis Lectures on Information Concepts, Retrieval, and Services* **7**(3) 1–115.

Davis, JM, H Topaloglu, DP Williamson. 2015. Assortment optimization over time. *Oper. Res. Lett.* **43**(6) 608–611.

Désir, A, V Goyal, Zhang J. 2016. Near optimal algorithms for the capacitated assortment optimization problem under random utility models. *Working paper* .

Désir, A, V Goyal, D Segev, C Ye. 2020. Constrained assortment optimization under the markov chain–based choice model. *Manag. Sci.* **66**(2) 698–721.

Duff, IS, J Koster. 2001. On algorithms for permuting large entries to the diagonal of a sparse matrix. *SIAM J Matrix Anal. A* **22**(4) 973–996.

Dufossé, F, K Kaya, I Panagiotas, B Uçar. 2018. Further notes on birkhoff–von neumann decomposition of doubly stochastic matrices. *Linear Algebra Appl.* **554** 68–78.

Dufossé, F, B Uçar. 2016. Notes on birkhoff–von neumann decomposition of doubly stochastic matrices. *Linear Algebra Appl.* **497** 108–115.

Feldman, JB, H Topaloglu. 2015. Capacity constraints across nests in assortment optimization under the nested logit model. *Oper. Res.* **63**(4) 812–822.

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

Ferreira, KJ, BHA Lee, D Simchi-Levi. 2015. Analytics for an online retailer: Demand forecasting and price optimization. *M & SOM* **18**(1) 69–88.

Gallego, G, A Li, VA Truong, X Wang. 2020. Approximation algorithms for product framing and pricing. *Oper. Res.* **68**(1) 134–160.

Gallego, G, R Ratliff, S Shebalov. 2014. A general attraction model and sales-based linear program for network revenue management under customer choice. *Oper. Res.* **63**(1) 212–232.

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

Gallego, G, GJ Van Ryzin. 1994. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Manage. Sci.* **40**(8) 999–1020.

Gallego, G, GJ Van Ryzin. 1997. A multiproduct dynamic pricing problem and its applications to network yield management. *Oper. Res.* **45**(1) 24–41.

Gandhi, R, S Khuller, S Parthasarathy, A Srinivasan. 2006. Dependent rounding and its applications to approximation algorithms. *J. ACM* **53**(3) 324–360.

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

Harsha, P, S Subramanian, J Uichanco. 2019. Dynamic pricing of omnichannel inventories. *M&SOM* **21**(1) 47–65.

Jasin, S. 2014. Reoptimization and self-adjusting price control for network revenue management. *Oper. Res.* **62**(5) 1168–1178.

Jasin, S, A Sinha. 2015. An lp-based correlated rounding scheme for multi-item ecommerce order fulfillment. *Oper. Res.* **63**(6) 1336–1351.

Kök, AG, ML Fisher, R Vaidyanathan. 2015. Assortment planning: Review of literature and industry practice. *Retail supply chain management*. Springer, 175–236.

Kök, AG, Y Xu. 2011. Optimal and competitive assortments with endogenous pricing under hierarchical consumer choice models. *Manage. Sci.* **57**(9) 1546–1563.

Kulkarni, J, E Lee, M Singh. 2017. Minimum birkhoff-von neumann decomposition. *IPCO*. 343–354.

Lei, YM, S Jasin, A Sinha. 2018. Joint dynamic pricing and order fulfillment for e-commerce retailers. *M & SOM* **20**(2) 269–284.

Lei, YM, S Jasin, J Uichanco, A Vakhutinsky. 2021. Joint product framing (display, ranking, pricing) and order fulfillment under the mnl model for e-commerce retailers. *Available at SSRN 3282019* .

Li, G, P Rusmevichientong, H Topaloglu. 2015. The d-level nested logit model: Assortment and price optimization problems. *Oper. Res.* **63**(2) 325–342.

Ma, W, Da Simchi-Levi. 2020. Algorithms for online matching, assortment, and pricing with tight weight-dependent competitive ratios. *Oper. Res.* .

Maddah, B, EK Bish. 2007. Joint pricing, assortment, and inventory decisions for a retailer's product line. *NRL* **54**(3) 315–330.

Rusmevichientong, P, ZJM Shen, DB Shmoys. 2010. Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Oper. Res.* **58**(6) 1666–1680.

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

Sumida, S, G Gallego, P Rusmevichientong, H Topaloglu, J Davis. 2020. Revenue-utility tradeoff in assortment optimization under the multinomial logit model with totally unimodular constraints. *Man. Sci.* .

Talluri, KT, GJ van Ryzin. 2004. Revenue management under a general discrete choice model of consumer behavior. *Manage. Sci.* **50**(1) 15–33.

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

Tversky, A, D Kahneman. 1981. The framing of decisions and the psychology of choice. *Science* **211**(4481) 453–458.

U.S. Census Bureau. 2020. Quarterly retail e-commerce sales. Available at `https://www.census.gov/retail/ecommerce/historic_releases.html` (2020/02/09).

Vakhutinsky, A, K Mihic, SM Wu. 2019. A prescriptive analytics approach to markdown pricing for an e-commerce retailer. *J. Pattern Recogn. Res.* **1** 1–21.

Von Neumann, J. 1953. A certain zero-sum two-person game equivalent to the optimal assignment problem. *Ann. Math. Stud.* **2** 5–12.

Wang, R. 2012. Capacitated assortment and price optimization under the multinomial logit model. *Oper. Res. Lett.* **40**(6) 492–497.

Xu, PJ, R Allgor, SC Graves. 2009. Benefits of reevaluating real-time order fulfillment decisions. *M & SOM* **11**(2) 340–355.

Electronic Companion to "Joint Product Framing (Display, Ranking, Pricing) and Order Fulfillment under the MNL Model for E-Commerce Retailers"

EC 1. Proof of Lemma 1 and 5

We will prove the more general statement Lemma 5, since Lemma 1 can be viewed as a special case with a single FC (i.e. $J=1$).

We first propose a reformulation of **DET**. Note that **DET** is a linear fractional program (LFP) and can be written as a linear program (LP) using the standard Charnes-Cooper variable transformation. Specifically, let $\boldsymbol{v}_j=(v_{ijr})_{i\in[I],r\in[R]}$ and define $\boldsymbol{z}_j^t=\boldsymbol{x}^t/(1+\boldsymbol{v}_j\cdot\boldsymbol{x}_j^t)$ and $w_j^t:=1/(1+\boldsymbol{v}_j\cdot\boldsymbol{x}_j^t)$ where $\boldsymbol{z}_j^t=(z_{ijr}^t)_{i,r}$. Then, we can re-write **DET** as follows:

$$
\begin{aligned}
\textbf{(DET-LP):}\quad \mathcal{J}^D = \max_{\{\boldsymbol{z}_j^t,w_j^t,\boldsymbol{y}_j^t\}}\ & \sum_{t\in[T],j\in[J]}\left[\lambda_j^t\cdot\boldsymbol{\psi}_j\cdot\boldsymbol{z}_j^t-\boldsymbol{c}_j\cdot\boldsymbol{y}_j^t\right] && \\
\text{s.t.}\ & A_1\cdot\boldsymbol{z}_j^t\le\boldsymbol{Q}\cdot w_j^t && \forall j,t \quad \text{(EC.1a)}\\
& \tilde{A}_2\cdot\boldsymbol{z}_j^t=\mathbf{1}\cdot w_j^t && \forall j,t \quad \text{(EC.1b)}\\
& A_3\cdot\boldsymbol{y}_j^t=\lambda_j\cdot\tilde{A}_2\cdot\left(\boldsymbol{v}_j\cdot\boldsymbol{z}_j^t\right) && \forall j,t \quad \text{(EC.1c)}\\
& \sum_{t\in[T],j\in[J]}\boldsymbol{y}_j^t\le\boldsymbol{C} && \text{(EC.1d)}\\
& \boldsymbol{v}^t\cdot\boldsymbol{z}_j^t+w_j^t=1 && \forall j,t \quad \text{(EC.1e)}\\
& \boldsymbol{z}_j^t\ge\mathbf{0},\ w_j^t\ge0,\ \boldsymbol{y}_j^t\in[0,1]^{I(K+1)} && \forall j,t \quad \text{(EC.1f)}
\end{aligned}
$$

where $\boldsymbol{\psi}_j=(\psi_{ijr})_{i\in[I],r\in[R]}$, $\psi_{ijr}=v_{ijr}\cdot p_{ir}$. Matrix $\tilde{A}_2$ in (EC.1b) and (EC.1c) is an $I\times IR$ coefficient matrix whose elements are either zero or one. The i^{th} row of $\tilde{A}_2$ has entries equal to 1 only in the column indices corresponding to product i. (Constraint (EC.1b) in **DET-LP** is due to constraint $\sum_{r\in[R]}x_{ijr}^t=1$, then multiplying both sides by w_j^t.) We can recover $\boldsymbol{x}_j^t$ via $\boldsymbol{x}_j^t=\boldsymbol{z}_j^t/w_j^t$ for all $j\in[J]$. Since w_j^t must be strictly positive, this recovery formula is well-defined.

Let $(\boldsymbol{X}_j^{t,*},\boldsymbol{Y}_j^{t,*})$ denote the online retailer's joint decisions under the optimal control π^* given an arrival of a customer from region j. Define $\boldsymbol{Z}_j^{t,*}:=\boldsymbol{X}_j^{t,*}/(1+\boldsymbol{v}_j^t\cdot\boldsymbol{X}_j^{t,*})$ and $W_j^{t,*}=1/(1+\boldsymbol{v}_j^t\cdot\boldsymbol{X}_j^{t,*})$. We can write the objective function in **OPT** defined in (9) as:

$$\mathcal{J}^*=\sum_{t\in[T],j\in[J]}\mathbb{E}\left[\lambda_j^t\cdot\boldsymbol{\psi}_j\cdot\boldsymbol{Z}_j^{t,*}-\boldsymbol{c}_j\cdot\boldsymbol{Y}_j^{t,*}\right]=\sum_{t\in[T],j\in[J]}\left[\lambda_j^t\cdot\boldsymbol{\psi}_j\cdot\mathbb{E}[\boldsymbol{Z}_j^{t,*}]-\boldsymbol{c}_j\cdot\mathbb{E}[\boldsymbol{Y}_j^{t,*}]\right]. \quad \text{(EC.2)}$$

So, to prove that $\mathcal{J}^*\le\mathcal{J}^D$, it is sufficient that we show $(\boldsymbol{z}_j^t,w_j^t,\boldsymbol{y}_j^t)=(\mathbb{E}[\boldsymbol{Z}_j^{t,*}],\mathbb{E}[W_j^{t,*}],\mathbb{E}[\boldsymbol{Y}_j^{t,*}])$ is a feasible solution for **DET-LP**. Let $\mathbf{1}_j^t\in\{0,1\}$ be an indicator variable for the arrival of a customer from region j in period t and $\boldsymbol{\Gamma}_j^t(\boldsymbol{s}_j^t)=(\Gamma_{ij}^t(\boldsymbol{s}_j^t))$ be a vector of random variables denoting the purchase decision of a customer from region j given purchase probabilities $\boldsymbol{s}_j = (s_{ij}^t) \in [0,1]^I$ with $\sum_{i\in[I]} s_{ij}^t \leq 1$. Formally, $\mathbb{P}(\Gamma_{ij}^t(\boldsymbol{s}_j^t) = 1 \,|\, \boldsymbol{s}_j) = s_{ij}^t$ and $\mathbb{P}(\Gamma_{ij}^t(\boldsymbol{s}_j^t) = 0 \,|\, \boldsymbol{s}_j^t) = 1 - s_{ij}^t$. Under control π^*, the purchase probabilities of a customer from region j in period t are given by $\boldsymbol{v}_j \cdot \boldsymbol{Z}_j^{t,*}$. Then, we can reformulate constraints (9a)-(9e) as the following set of (in-)equalities:

$$A_1 \cdot \boldsymbol{Z}_j^{t,*} \leq \boldsymbol{q} \cdot W_j^{t,*} \qquad \forall j, t \tag{EC.3a}$$

$$\tilde{A}_2 \cdot \boldsymbol{Z}_j^{t} = \mathbf{1} \cdot W_j^{t} \qquad \forall j, t \tag{EC.3b}$$

$$A_3 \cdot \boldsymbol{Y}_j^{t,*} = \mathbf{1}_j^t \cdot \boldsymbol{\Gamma}_j^t(\boldsymbol{v}_j \cdot \boldsymbol{Z}_j^{t,*}) \qquad \forall j, t \tag{EC.3c}$$

$$\sum_{t\in[T], j\in[J]} \boldsymbol{Y}_j^{t,*} \leq \boldsymbol{C} \tag{EC.3d}$$

$$\boldsymbol{v}_j \cdot \boldsymbol{Z}_j^{t,*} + W_j^{t,*} = 1 \qquad \forall j, t \tag{EC.3e}$$

$$\boldsymbol{Z}_j^{t,*} \geq \mathbf{0},\, W_j^{t,*} \geq 0,\, \boldsymbol{Y}_j^{t,*} \in \{0,1\}^{I(K+1)} \qquad \forall j, t \tag{EC.3f}$$

where (EC.3a) and (EC.3b) are obtained by dividing equations (8a) and (8b) by $1 + \boldsymbol{v}_j \cdot \boldsymbol{X}_j^{t,*}$, and (EC.3e) captures the choice behavior under MNL model. Taking expectation on both sides of constraints (EC.3a)-(EC.3f) immediately shows that $(\mathbb{E}[\boldsymbol{Z}_j^{t,*}], \mathbb{E}[W_j^{t,*}], \mathbb{E}[\boldsymbol{Y}_j^{t,*}])$ satisfies constraints (EC.1a)-(EC.1f). We conclude that $(\mathbb{E}[\boldsymbol{Z}_j^{t,*}], \mathbb{E}[W_j^{t,*}], \mathbb{E}[\boldsymbol{Y}_j^{t,*}])$ is feasible for **DET-LP**. ■

EC 2. Proof of Theorem 1 and 2

The proof is done for Theorem 2 only, since Theorem 1 can be viewed as a special case with a single FC (i.e. $J = 1$). Without loss of generality, we assume that $T = 1$. We consider a variant of RPDF (V-RPDF) defined as follow: during period t, always apply framing assignment $\phi_{j(t)}^t$ instead of $\tilde{\phi}_{j(t)}^t$ (i.e. ignore stockout); if there is an order arrives, fulfill the order from region j according to $\boldsymbol{\sigma}_j^t$ regardless of the availability of the corresponding FC. If the FC runs out of inventory, the retailer can still gather a revenue of p_{ir}, but incurs a penalty cost of $\bar{c} := 2 \cdot \max\{\max_{i\in[I], j\in[J], k\in[K]} c_{ijk}, \max_{i\in[I], r\in[R]} p_{ir}\}$. In other words, V-RPDF incurs the same revenue as RPDF if there is no stockout. But, in the case of stock out, RPDF will generate zero profit while V-RPDF generates negative profit. Therefore, the loss of RPDF can be bounded as follows.

$$\begin{aligned}
&\mathcal{J}^D(\theta) - \mathcal{J}^{RPDF}(\theta) \\
\leq\ &\mathcal{J}^D(\theta) - \mathcal{J}^{V-RDPF}(\theta) \\
=\ &\sum_{t\in[T(\theta)], i\in[I], j\in[J]} \left[\mathcal{R}_{ij}(\boldsymbol{x}^{t,D}) - \mathbb{E}\left[\mathcal{R}_{ij}(\boldsymbol{X}^t)\right]\right] - \sum_{t\in[T(\theta)], i\in[I], j\in[J], k\in[K]} c_{ijk} \cdot \left(y_{ijk}^{t,D} - \mathbb{E}\left[Y_{ijk}^t\right]\right) \\
&\quad + \bar{c}\, \mathbb{E}\left[\sum_{i\in[I], k\in[K]} \left(\sum_{t\in[T(\theta)], j\in[J]} Y_{ijk}^t - C_{ik}(\theta)\right)^+\right]
\end{aligned} \tag{EC.4}$$

The first term in (EC.4) equals to zero by Lemma 3; The second term also equals to zero by Lemma 3 and the way we sample fulfillment assignment decisions. Define $\Delta Y_{ijk}^t := Y_{ijk}^t - \mathbb{E}[Y_{ijk}^t] = Y_{ijk}^t - y_{ijk}^{t,D}$, which is a sequence of i.i.d random variables with zero means and bounded variances. The last term in (EC.4) can be bounded from above as follows:

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{i\in[I],k\in[K]}\left(\sum_{t\in[T(\theta)],j\in[J]} Y_{ijk}^t - C_{ik}(\theta)\right)^+\right] \\
&\leq \mathbb{E}\left[\sum_{i\in[I],k\in[K]}\left(\sum_{t\in[T(\theta)],j\in[J]} Y_{ijk}^t - \sum_{t\in[T],j\in[J]} y_{ijk}^t\right)^+\right] + \mathbb{E}\left[\sum_{i\in[I],k\in[K]}\left(\sum_{t\in[T(\theta)],j\in[J]} y_{ijk}^t - C_{ik}(\theta)\right)^+\right] \\
&\leq \sum_{i\in[I],,j\in[J],k\in[K]} \mathbb{E}\left[\left(\sum_{t\in[T(\theta)]} \Delta Y_{ijk}^t\right)^+\right] + 0 \;\leq \sum_{i\in[I],,j\in[J],k\in[K]}\left[\mathbf{Var}\left(\sum_{t\in[T(\theta)]} \Delta Y_{ijk}^t\right)\right]^{1/2} \\
&\leq IJK\sqrt{T(\theta)} = IJK\sqrt{T\cdot\theta}
\end{aligned}
$$

where the first inequality follows from triangular inequality and Jensen's inequality, the second inequality follows from the inventory constraint (5c), the third inequality follows from Jensen's inequality, the fourth inequality holds since the Y_{ijk}^t are independent across periods and is bounded within $[0,1]$. The upper bound on $1 - \mathcal{J}^{RFF}(\theta)/\mathcal{J}^*(\theta)$ is proved in Section EC.3. ■

EC 3. Proof of Corollary 1

Similar with the previous proof, we conduct the analysis for the joint framing and fulfillment setting, which incorporate the framing only setting as a special case. In particular, we want to show that, if $\mathcal{J}^*(1) > 0$, then we have $1 - \mathcal{J}^{RFF}(\theta)/\mathcal{J}^*(\theta) = \mathcal{O}(\theta^{-1/2}) \to 0$ as $\theta \to \infty$.

We first show that, the optimal solution to DET is also stationary. We show this by a constructive proof. Let $\{\bar{\boldsymbol{z}}_j^t, \bar{w}_j^t, \bar{\boldsymbol{y}}_j^t\}$ be any optimal solution of **DET-LP**. Define $(\boldsymbol{z}_j^{1,D}, w_j^{1,D}, \boldsymbol{y}_j^{1,D})$ as follows:

$$
\boldsymbol{z}_j^{1,D} = \frac{\sum_{t=1}^{T(\theta)} \bar{\boldsymbol{z}}_j^t}{T(\theta)},\; w_j^{1,D} = \frac{\sum_{t=1}^{T(\theta)} \bar{w}_j^t}{T(\theta)}, \;\text{ and }\; \boldsymbol{y}_j^{1,D} = \frac{\sum_{t=1}^{T(\theta)} \bar{\boldsymbol{y}}_j^t}{T(\theta)}.
$$

Note that setting $\boldsymbol{z}_j^t = \boldsymbol{z}_j^{1,D}$, $w_j^t = w_j^{1,D}$, and $\boldsymbol{y}_j^t = \boldsymbol{y}_j^{1,D}$ for all t is feasible for **DET-LP**. Moreover, $\{\boldsymbol{z}_j^t, w_j^t, \boldsymbol{y}_j^t\}$ also preserves the optimal objective value of **DET-LP** under $\{\bar{\boldsymbol{z}}_j^t, \bar{w}_j^t, \bar{\boldsymbol{y}}_j^t\}$. This means that $\{\boldsymbol{z}_j^t, w_j^t, \boldsymbol{y}_j^t\}$ is an optimal stationary solution for **DET-LP**. To construct the corresponding $\boldsymbol{x}_j^{1,D}$, we simply let $\boldsymbol{x}_j^{1,D} = \frac{\boldsymbol{z}_j^{1,D}}{w_j^{1,D}}$ for all j (by constraints (EC.1d) and (EC.1e), $w_j^{1,D}$ must be strictly positive, so this is well-defined).

By stationarity of the optimal deterministic solution, we know that $\mathcal{J}^D(\theta) = \theta \cdot \mathcal{J}^D(1) \geq \theta \cdot \mathcal{J}^*(1) > 0$. Since $\mathcal{J}^*(\theta) \geq \mathcal{J}^{RFF}(\theta)$, Theorem 1 implies $\mathcal{J}^D(\theta) - \mathcal{J}^*(\theta) \leq M \cdot \sqrt{\theta}$. Therefore, we know that there must exist some M' such that $\mathcal{J}^*(\theta) \geq \mathcal{J}^D(1) \cdot \theta - M \cdot \sqrt{\theta} \geq M' \cdot \theta$. As a result,

$$
1 - \frac{\mathcal{J}^{RFF}(\theta)}{\mathcal{J}^*(\theta)} \;\leq\; 1 - \frac{\mathcal{J}^{RFF}(\theta)}{\mathcal{J}^*(\theta)} \;\leq\; \frac{M \cdot \sqrt{\theta}}{M' \cdot \theta - M \cdot \sqrt{\theta}} \;=\; \mathcal{O}(\theta^{-1/2}) \;\blacksquare
$$

EC 4. Proof of Lemma 2

Since the sampling distribution for each demand location is constructed independently, it suffices to prove the case where $J=1$. We first argue that B is indeed a framing matrix with respect to $\tilde{Q}$. The column sum condition is verified directly by summing up B_{ir}. The row sum conditions for the first I rows hold because of constraint (5b). For the rows with index $i>I$, we have

$$\sum_{r\in[R]} B_{i,r} = \frac{\sum_{r\in[R]} \lceil q_r \rceil - \sum_{r\in[R],i\in[I]} x_{ir}^D}{\tilde{I}} = \frac{\sum_{r\in[R]} \lceil q_r \rceil - \sum_{i\in[I]} 1}{\tilde{I}} = 1$$

where the second inequality follows from constraint (5b), the third equality follows from the definition of $\tilde{I}$ in Step C1(i).

We now argue that, for any integral assignment matrix corresponding to $\tilde{Q}$, there exist a unique and feasible framing decision ϕ. Firstly, given fixed product $i\in[I]$, there exists a unique r such that $B_{ir}=1$. This means that each product is uniquely assigned to one group (including the no-display group), which confirms that an framing decision ϕ exists uniquely. Secondly, given fixed group r, the maximum amount of product displayed in group r is $\sum_{i\in[I]} \mathbf{1}\{\phi(i)=r\} = \sum_{i\in[I]} B_{ir} = q_r \le Q_r$, where the inequality follows from the cardinality constraint (5a). Moreover, the number of products displayed equal to I. This confirms the feasibility of ϕ. ■

EC 5. Proof of Lemma 3

By abuse of notation, we neglect the superscript t since the construction of $\tilde{\rho}$ is the same for all periods. Instead of directly showing that conditions (7a) - (7c) are satisfied, we characterize a set of conditions and argue that they are sufficient conditions to (7a) - (7c), and then we show that the proposed sufficient conditions are satisfied. This set of conditions are:

$$\sum_{\boldsymbol{\phi}\in\Omega(\boldsymbol{Q})} \tilde{\rho}(\boldsymbol{\phi}) = 1; \tag{EC.5}$$

$$\sum_{\boldsymbol{\phi}\in\Omega(\boldsymbol{Q})} \tilde{\rho}_j(\boldsymbol{\phi}) \cdot v_i(\boldsymbol{\phi}) = \sum_{r\in[R]} v_{ir} \cdot x_{ir}^D \qquad \forall\, i; \tag{EC.6}$$

$$\sum_{\boldsymbol{\phi}\in\Omega(\boldsymbol{Q})} \tilde{\rho}_j(\boldsymbol{\phi}) \cdot r_i(\boldsymbol{\phi}) = \sum_{r\in[R]} v_{ir} \cdot p_{ir} \cdot x_{ir}^D, \qquad \forall\, i. \tag{EC.7}$$

We first argue that conditions (EC.5) - (EC.7) are sufficient. By definition of $\rho(\boldsymbol{\phi})$ in Step C3 of ConstrDist, we have

$$\sum_{\boldsymbol{\phi}\in\Omega(\boldsymbol{Q})} \rho_j(\boldsymbol{\phi}) = \frac{\sum_{\boldsymbol{\phi}\in\Omega(\boldsymbol{Q})} \tilde{\rho}(\boldsymbol{\phi}) \cdot \left(1 + \sum_{i\in[I]} v_i(\boldsymbol{\phi},)\right)}{1 + \sum_{\boldsymbol{\phi}'\in\Omega(\boldsymbol{Q})} \tilde{\rho}(\boldsymbol{\phi}') \cdot \left(\sum_{i\in[I]} v_i(\boldsymbol{\phi}')\right)} = 1$$

where the last equality uses condition (EC.5). This verifies condition (7a).

For condition (7b), we know by the definition of $\theta_i(\boldsymbol{\phi})$ that

$$\begin{aligned}\sum_{\boldsymbol{\phi}\in\Omega(\boldsymbol{Q})} \rho(\boldsymbol{\phi})\cdot\theta_i(\boldsymbol{\phi}) &= \lambda\cdot \sum_{\boldsymbol{\phi}\in\Omega(\boldsymbol{Q})} \rho(\boldsymbol{\phi})\cdot \frac{v_i(\boldsymbol{\phi})}{1+\sum_{k\in[I]} v_k(\boldsymbol{\phi})} \\ &= \lambda\cdot \sum_{\boldsymbol{\phi}\in\Omega(\boldsymbol{Q})} \tilde{\rho}(\boldsymbol{\phi})\cdot \frac{1+\sum_{i\in[I]} v_i(\boldsymbol{\phi})}{1+\sum_{\boldsymbol{\phi}'\in\Omega(\boldsymbol{Q})} \tilde{\rho}(\boldsymbol{\phi}')\cdot\left(\sum_{i\in[I]} v_i(\boldsymbol{\phi}')\right)}\cdot \frac{v_i(\boldsymbol{\phi})}{1+\sum_{k\in[I]} v_k(\boldsymbol{\phi})} \\ &= \lambda\cdot \frac{\sum_{\boldsymbol{\phi}\in\Omega(\boldsymbol{Q})} \tilde{\rho}(\boldsymbol{\phi})\cdot v_i(\boldsymbol{\phi})}{1+\sum_{\boldsymbol{\phi}'\in\Omega(\boldsymbol{Q})} \tilde{\rho}(\boldsymbol{\phi}')\cdot\left(\sum_{i\in[I]} v_i(\boldsymbol{\phi}')\right)} \\ &= \lambda\cdot \frac{\sum_{r\in[R]} v_{ir}\cdot x^D_{ir}}{1+\sum_{k\in[I],r\in[R]} v_{kr}\cdot x^D_{kr}} = \theta_i(\boldsymbol{x}^D)\end{aligned}$$

where last two equalities both follow from condition (EC.6). Following a very similar argument, we can validate that condition (7c) holds under (EC.6) and (EC.7):

$$\begin{aligned}\sum_{\phi\in\Omega(\boldsymbol{Q})} \rho(\phi)\cdot\mathcal{R}_i(\phi) &= \lambda\cdot \sum_{\boldsymbol{\phi}\in\Omega(\boldsymbol{Q})} \tilde{\rho}(\boldsymbol{\phi})\cdot \frac{1+\sum_{i\in[I]} v_i(\boldsymbol{\phi})}{1+\sum_{\boldsymbol{\phi}'\in\Omega(\boldsymbol{Q})} \tilde{\rho}(\boldsymbol{\phi}')\cdot\left(\sum_{i\in[I]} v_i(\boldsymbol{\phi}')\right)}\cdot \frac{r_i(\boldsymbol{\phi})}{1+\sum_{k\in[I]} v_k(\boldsymbol{\phi})} \\ &= \lambda\cdot \frac{\sum_{\boldsymbol{\phi}\in\Omega(\boldsymbol{Q})} \tilde{\rho}(\boldsymbol{\phi})\cdot r_i(\boldsymbol{\phi})}{1+\sum_{\boldsymbol{\phi}'\in\Omega(\boldsymbol{Q})} \tilde{\rho}(\boldsymbol{\phi}')\cdot\left(\sum_{i\in[I]} v_i(\boldsymbol{\phi}')\right)} \\ &= \lambda\cdot \frac{\sum_{r\in[R]} v_{ir}\cdot p_{ir}\cdot x^D_{ir}}{1+\sum_{k\in[I],r\in[R]} v_{kr}\cdot x^D_{kr}} = \mathcal{R}_i(\boldsymbol{x}^D)\end{aligned}$$

We now proceed to verify conditions (EC.5) - (EC.7). Condition (EC.5) follows directly by the definition of $\tilde{\rho}(\boldsymbol{\phi})$. For condition (EC.6), by definition of $v_{ij}(\boldsymbol{\phi})$, we know that

$$\begin{aligned}\sum_{\boldsymbol{\phi}\in\Omega(\boldsymbol{Q})} \tilde{\rho}(\boldsymbol{\phi})\cdot v_i(\boldsymbol{\phi}) &= \sum_{\boldsymbol{\phi}\in\Omega(\boldsymbol{Q})} \tilde{\rho}(\boldsymbol{\phi})\cdot\left(\sum_{r\in[R]} v_{ir}\cdot\mathbf{1}\{\phi(i)=r\}\right) = \sum_{r\in[R]} v_{ir}\cdot\left(\sum_{\boldsymbol{\phi}\in\Omega(\boldsymbol{Q})} \tilde{\rho}(\boldsymbol{\phi})\cdot\mathbf{1}\{\phi(i)=r\}\right) \\ &= \sum_{r\in[R]} v_{ir}\cdot\left(\sum_{\ell=1}^{L} \alpha_\ell\cdot\mathbf{1}\left\{\phi^{(\ell)}(i)=r\right\}\right)\cdot x^D_{ir} = \sum_{r\in[R]} v_{ir}\cdot x^D_{ir}\end{aligned}$$

where the third equality follows from the definition of matrix B is Step C1(ii). The proof of condition (EC.7) is essentially the same as above. ■

EC 6. Improved Framing Decomposition Algorithm

Our proposed framing decomposition algorithm is motivated by improving the basic algorithm (see Section 4.3) in the context of a framing decomposition.

FrameDecomp_Improv

Input: B, a proper framing matrix with respect to $\boldsymbol{Q}$

Output: A decomposition $B=\sum_\ell \alpha^\ell\cdot B^{(\ell)}$

D1. Same as FrameDecomp Step D1.

D2. Initialize $L \leftarrow 0$ (the number of unique integral framing matrices found).

Initialize M to be a bottleneck matching in $G_{B^\dagger}$.

While $B^\dagger$ is not a zero matrix, do:

i. - v. Same as FrameDecomp Step D2 ii. - vi. in the While loop.

vi. Update the matching $M \leftarrow M \cap G_{B^\dagger}$, and construct S_0 as the set of all products not in M;

vii. While S_0 (the unmatched products) is not empty, do:

a. Choose i_0 from S_0 arbitrarily;

b. Apply algorithm AugmentingPath (see below), with inputs $B^\dagger, M, i_0, \phi(i_0), g$, to find M-augmenting path P rooted from i_0;

c. Update matching $M \leftarrow (M \setminus P) \cup (P \setminus M)$;

d. Remove i_0 from S_0;

The iterations in our heuristic begin in the same way as Dufossé and Uçar (2016) (steps D2(i)-(v) above). The key difference is in the choice of the next perfect matching (steps D2(vi)-(vii)). In those steps, the original algorithm chooses a bottleneck matching. But since the matchings in successive iterations often only have few edges in common, this could potentially result in very different framing matrices. In the improved decomposition, the perfect matching in each iteration is chosen to be the one with the least framing group reassignments compared to the old matching. (For any two matchings, the number of framing group reassignments can be computed by counting the number of products that are not assigned to the same framing group.)

We next discuss how the algorithm works by describing what happens in one iteration. Let M be the perfect matching found in the last iteration. From M, we drop the edges that correspond to the entries on $B^\dagger$ that were zeroed out in FrameDecomp Step D2(vi) of the last iteration, and let S_0 be the set of all unmatched products. Note that M is a matching, but it is not a perfect matching. Our aim is to augment M such that the resulting perfect matching has the least group reassignments compared to the old perfect matching. We do this by using the concept of an M-augmenting path.

Definition 1 (M-augmenting path). Given a matching $M \subseteq G_{B^\dagger}$, a path $P \subseteq G_{B^\dagger}$ is said to be *M-alternating* if for any *incident* edges e_1 and e_2 in P (two edges are said to be *incident* if they share a vertex), either $e_1 \in M$ and $e_2 \notin M$, or $e_1 \notin M$ and $e_2 \in M$. Moreover, P is called *M-augmenting* if the first and the last vertices in P are not in M.

Our algorithm chooses a arbitrary product i_0 from S_0, then finds an M-augmenting path P that starts from i_0 and ends with an unmatched subgroup u_0. Note the update $M \leftarrow (M \setminus P) \cup (P \setminus M)$

results in a matching with one more edge where both i_0 and u_0 are now included. It does this until S_0 is empty and all products are matched, so M is a perfect matching.

The choice of augmenting paths is critical to the algorithm. Note that, for an unmatched product i_0, there may be many M-augmenting paths starting from product i_0. The algorithm AugmentingPath (described below) explores M-alternating paths from i_0 until it finds the augmenting path with the minimum number of framing group reassignments. If there are more than one perfect matchings with minimal framing group reassignments, our heuristic simply chooses the one with the largest bottleneck. Using the bottleneck value as a tiebreaker may potentially reduce the number of iterations in FrameDecomp_Improv.

AugmentingPath

Input: A matrix $B^\dagger$; a matching M on $G_{B^\dagger} = (\mathcal{I} \cup \mathcal{U}, E)$; an unmatched product $i_0 \in \mathcal{I}$ and its previous framing group $\phi(i_0) \in [R]$; the mapping $g : \mathcal{U} \mapsto [R]$ of subgroup index to framing group

Output: P, the M-augmenting path with the smallest number of framing group reassignments, where ties are broken by choosing the path with the largest bottleneck of edges not in M

A1. Set $K_1 \leftarrow \emptyset$, $K_0 \leftarrow \emptyset$, $n^* \leftarrow \infty$, $b^* \leftarrow 0$, and for each subgroup $u \in \mathcal{U}$, set $n_u \leftarrow \infty$, $b_u \leftarrow 0$, and $p_u \leftarrow$ null;

A2. Initialize $i \leftarrow i_0$, $r^{\text{old}} \leftarrow \phi(i_0)$, $n \leftarrow 0$, $b \leftarrow \infty$;

A3. While $(n < n^*)$ or $(n = n^*$ and $b > b^*)$, do:

  i. For all $u \in \{u \in \mathcal{U} : (i,u) \in E\} \setminus K_1$, do:
    Set $r^{\text{new}} \leftarrow g(u)$, $n^{\text{new}} \leftarrow n + \mathbf{1}\left\{r^{\text{old}} \neq r^{\text{new}}\right\}$, and $b^{\text{new}} \leftarrow \min(B^\dagger_{iu}, b)$;
    If $(n^{\text{new}} < n^*)$ or $(n^{\text{new}} = n^*$ and $b^{\text{new}} > b^*)$, then:
    a. If u is unmatched, then $n^* \leftarrow n^{\text{new}}$, $b^* \leftarrow b^{\text{new}}$, $u^* \leftarrow u$, and $p_u \leftarrow i$;
    b. Else: If $(n^{\text{new}} < n_u)$ or $(n^{\text{new}} = n_u$ and $b^{\text{new}} > b_u)$, then $n_u \leftarrow n^{\text{new}}$, $b_u \leftarrow b^{\text{new}}$, $p_u \leftarrow i$, and $K_0 \leftarrow K_0 \cup \{u\}$;

  ii. If $K_0 = \emptyset$, then exit while-loop;

  iii. Choose $u \in K_0$ with minimal n_u, where in the case of ties, choose one with the maximal b_u; Set $n \leftarrow n_u$ and $b \leftarrow b_u$;

  iv. Set $K_0 \leftarrow K_0 \setminus \{u\}$ and $K_1 \leftarrow K_1 \cup \{u\}$;

  v. Set $i \leftarrow M(u)$, and $r^{\text{old}} \leftarrow g(u)$;

A4. Construct an M-alternating path by initializing $u \leftarrow u^*$ and $P = \{(p_{u^*}, u^*)\}$
  While $p_u \neq i_0$, do:

  i. $P \leftarrow P \cup (p_u, M(p_u))$;

  ii. Set $u \leftarrow M(p_u)$, then update $P \leftarrow P \cup (p_u, u)$;

The augmenting path algorithm starts from the unmatched product i_0 and then grows M-alternating paths while keeping track of two metrics: (i) the number of products that are reassigned to a different framing group (a product is reassigned if the two subgroup nodes adjacent to it in the path belong to different framing groups) and (ii) the bottleneck value (minimum weight) of all edges not in M. The algorithm explores an M-alternating path until either an M-augmenting path is found or it detects another path with either a smaller number of reassignments or with an equal number of reassignments and a larger bottleneck value. If another path is detected, then the algorithm next grows this other path. The number of reassignments and the bottleneck value in the path currently being grown is n and b, respectively. Of the M-augmenting paths found, n^* is the smallest number of reassignments, b^* is its bottleneck value, and u^* is its unmatched subgroup index. $K_0 \cup K_1$ is the set of all subgroup indices where an M-alternating path to root i_0 has been found; K_1 is the set of subgroup indices where the path has been continued. Of the M-alternating paths grown between subgroup u and product i_0, n_u is the smallest number of reassignments with b_u being its bottleneck value and p_u being its predecessor product in this path.

EC 7. Proof of Lemma 4

For both FrameDecomp and FrameDecomp_Improv (defined formally in Section), the *while* loop in Step D2 is terminated after at most $(I+\tilde{I})^2 \leq 4I^2$ rounds. This is because at least one non-zero element in $B^\dagger$ is eliminated at each round and there at most $(I+\tilde{I})^2$ non-zero elements in $B^\dagger$. Equivalently, there is at least one edge dropped from the products-subgroup graph in each round and there are at most $(I+\tilde{I})^2$ edges in this bipartite graph.

Given that the while loop terminate in polynomial time, the polynomial running time for FrameDecomp follows directly from the fact that bottleneck matching can be found in polynomial time (Duff and Koster 2001). Compared to FrameDecomp, the additional computation of FrameDecomp_Improv lies in the algorithm AugmentingPath, which can be executed in polynomial time.

References

Duff, IS, J Koster. 2001. On algorithms for permuting large entries to the diagonal of a sparse matrix. *SIAM J Matrix Anal. A* **22**(4) 973–996.

Dufossé, F, B Uçar. 2016. Notes on birkhoff–von neumann decomposition of doubly stochastic matrices. *Linear Algebra Appl.* **497** 108–115.

Online Supplements to "Joint Product Framing (Display, Ranking, Pricing) and Order Fulfillment under the MNL Model for E-Commerce Retailers"

In this document, we provide additional details to the numerical experiments in Section 6. We also present additional numerical experiments that the impact of other key problem parameters on the performance of the proposed heuristics, including number of display groups, load factor, total market size, and number of allowed changes to the displays.

O1. Additional details to the computational experiments

We are provided with the estimates of MNL demand models for 98 customer regions ($J = 98$) within the continental US. These estimates are for the arrival rates $(\lambda_j)_{j\in[J]}$ and the MNL parameters $(\eta_{ij})_{i\in[I],j\in[J]}$ and $(\gamma_{ir})_{i\in[I],r\in[R]}$ where $v_{ijr} = \exp(\eta_{ij} + \gamma_{ir})$ is the MNL attraction coefficient. Here, η_{ij} is the utility attributed to the product and γ_{ir} is the utility attributed to the framing decision. For the MNL parameters themselves, we set $\gamma_{i1} = 0$ (this allows us to interpret $(\eta_{ij})_{i\in[I],j\in[J]}$ as the base utilities without promotion effect) and $\gamma_{i3} = -\infty$ (this allows us to turn-off demand when the product is not displayed). This means that, among the framing effect parameters, we only need to estimate γ_{i2}. Below, we provide summary statistics of the MNL coefficients $(\eta_{ij})_{i\in[I],j\in[J]}$ and $(\gamma_{i2})_{i\in[I]}$:

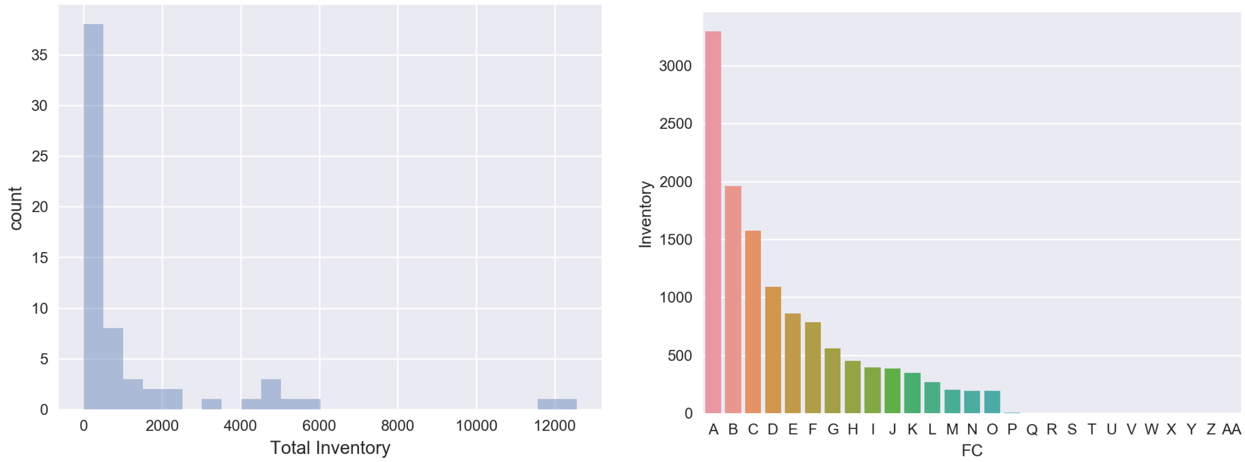
	Average	Median	Stdev	Min	Max
η_{ij}	-3.02	-3.07	1.61	-7.29	1.62
γ_{i2}	1.06	1.08	0.23	-0.09	1.42

By exponentiating the estimated promotion effect γ_{i2}, we can compute the relative risk ratios (RRR) of buying product i with promotion. The relative risk ratios of the promotion coefficient have the following statistics:

	Average	Median	Stdev	Min	Max
RRR of promotion	2.98	2.95	0.59	0.92	4.12

We can infer from the data that promoting a product increases the probability of buying the product (compared to no purchase) by roughly a factor of 3 on average, assuming that the other variables in the model are held constant.

Figure O.1 shows the scarcity of the initial inventory, and the imbalance of inventory throughout the network. Figure O.1a shows the histogram of the total inventory count per product. From this figure, we can see that a majority of the 56 products start the clearance season with less than 500 units of inventory. Figure O.1b shows the inventory level of product 7 in the 27 fulfillment centers.



(a) (b)

Figure O.1 (a) A histogram showing the count of products for each total inventory bin. (b) A barplot illustrating how inventory units of product 7 are distributed across the 27 FCs.

Although product 7 has the most total inventory out of all products, it is only in-stock in roughly half of the fulfillment centers.

We also comment on the computational practicality of RFF for large networks. CPLEX 12.10 with default termination criteria solves the LP counterpart of the deterministic model **DET'** (with 241,570 variables and 12,838 constraints) in 32 seconds on a 2.4 GHz 8-Core Intel Core i9 processor. The decomposition algorithm ConstrDist runs in 6 seconds. Since RFF requires the deterministic model to be solved only once at the start of the selling season, it is very practical for realistic settings. We also test the sensitivity of the run times to the number of display groups in Section of the Online Appendix. With roughly 30 display groups, a solution to **DET'** is found after 3 minutes, and the decomposition algorithm runs in 18 seconds. Table O.1 also shows that the performance of RFF and the best-performing RF policy are robust to the number of groups.

O2. Sensitivity to the number of groups

In Section 6, there are only 3 groups: displayed and promoted, displayed but not promoted, and not displayed. We tested the sensitivity of RFF and RF to the number of groups, while fixing the number of displayed products to be 30. Note that the case with 31 groups is equivalent to the problem of choosing 30 products to be displayed, then ranking them on the display page.

In Table O.1, we can see: (i) the size of the **DET'** model and its solving time, (ii) the runtime of the decomposition algorithm, (iii) the performance of RFF, and (iv) the performance of the best competing heuristic, RF (Reopt) with LP fulfillment. We can observe a few things from this table. First, even though the runtimes increase with the group number, in the worst case, the RFF policy can be determined within 3.5 minutes. Second, the optimality gap of RFF and RF (Reopt) with LP fulfillment are generally unchanging with the number of groups.

	DET' model			RFF			RF (Reopt) + LP	
# Grp.	# Vars.	# Constr.	Time[1] (sec)	Time[2] (sec)	Total profit ($ millions)	Profit gap	Total profit ($ millions)	Profit gap
3	241,570	12,838	32	6	2.89 ± 0.002	2.20%	2.79 ± 0.001	5.56%
4	285,474	12,936	34	7	2.89 ± 0.003	2.20%	2.73 ± 0.003	5.45%
5	329,378	13,034	54	8	2.89 ± 0.002	2.23%	2.80 ± 0.001	5.34%
7	417,186	13,230	92	8	2.89 ± 0.002	2.18%	2.80 ± 0.001	5.23%
16	812,322	14,112	187	11	2.89 ± 0.003	2.20%	2.80 ± 0.001	5.05%
31	1,470,882	15,582	193	18	2.88 ± 0.002	2.44%	2.80 ± 0.001	5.04%

[1] Total time of building the model in CPLEX 12.10 and solving the model on a 2.4 GHz 8-Core Intel Core i9 processor.
[2] Total time of the decomposition algorithm ConstrDist on a 2.4 GHz 8-Core Intel Core i9 processor.

Table O.1 Size and solution time of DET, and performance of policies with different number of groups.

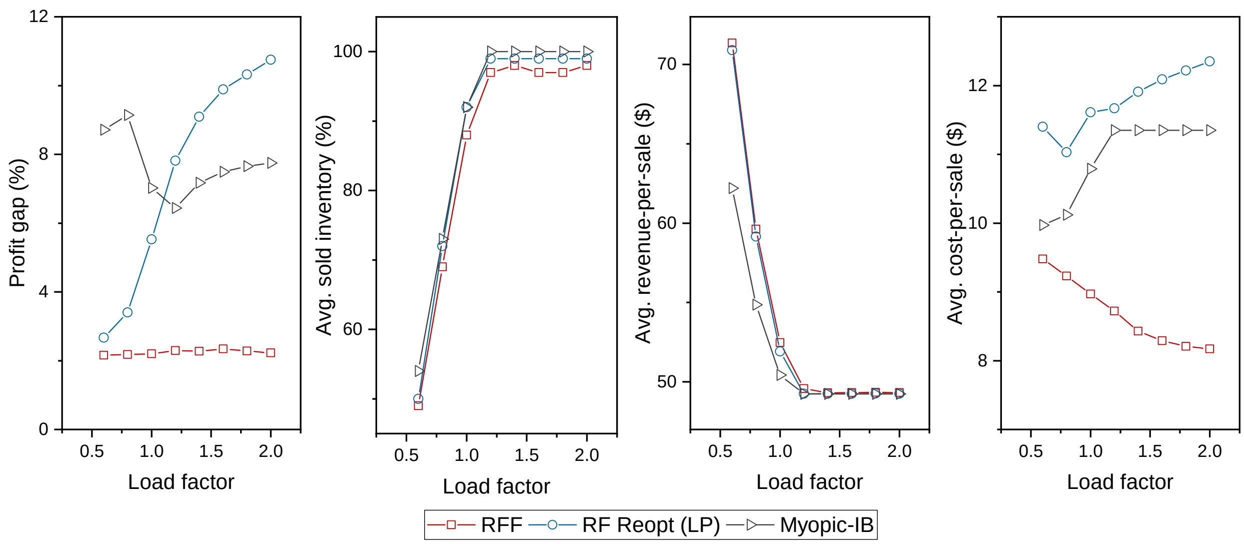


Figure O.2 Policy metrics as a function of the load factor. The metrics plotted are the profit gap, the average sales as a proportion of inventory, the average revenue-per-sale, and the average cost-per-sale.

O3. Sensitivity to the load factor

We tested the sensitivity of the results to the load factor. We define the load factor as the ratio of the total customer arrivals to the total inventory. In the experiments of Section 6, the load factor is 1. We repeated the experiments with load factors 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, and 2. In these experiments, we fixed the total inventory (75,352 units) and varied the total number of arrivals. We compare the performance of the RFF policy against two policies that ignore fulfillment costs when choosing framing decisions: RF (with weekly reoptimization) and Myopic-IB. For the RF policy, the LP heuristic is used for order fulfillment. Figure O.2 summarizes the results of this experiment.

We observe from Figure O.2 that the profit gap of RFF relative to the upper bound is robust against the different values of the load factor. Specifically, its profit gap is consistently below 2.5%. In contrast, the profit gaps of RF and Myopic-IB both are affected by the load factor. By comparing the profit gap of RFF against these two policies, we observe that the value of jointly optimizing framing and fulfillment is highest when inventory is scarce (i.e., high load factor).

Observe that the RF profit gap is increasing with the total customer arrivals (load factor). Recall from Figure 5 that RF does not control the location of demand, whereas RFF ensures that it occurs close to the inventory location. The latter strategy is critical for profitability when a product is out-of-stock in many FCs. When the load factor is low, however, there are fewer FCs that would stock out of a product during the simulation. Hence, the location of the demand would matter less since the sale could be fulfilled from a nearby FCs. This is evident from Figure O.2 that shows the difference in cost-per-sale of RF and RFF is smallest for low values of the load factor. As the load factor increases, an increasing number of FCs stockout of a product during the simulation. Hence, the location of demand becomes increasingly important for profitability. This explains why the RF cost-per-sale (and the profit gap) is increasing with the load factor.

By using a penalty function, Myopic-IB tends to promote and display products with the most inventory left. This explains why in Figure O.2 the cost-per-sale under Myopic-IB is notably lower than that of RF, even though both policies ignore fulfillment costs when making framing decisions. However, for low load factor values, the revenue-per-sale of Myopic-IB is notably lower than those of RF and RFF. As a result, Myopic-IB has the highest profit gap for load factors that are 1 or less. Note that when the load factor is 1.2 or above, the revenue-per-sale under the three policies are comparable. Hence, in that range, Myopic-IB has a better performance than RF due to its lower fulfillment cost-per-sale.

Unlike RF and RFF, Myopic-IB does not use information on the total arrivals when choosing a framing decision. As a result, when the load factor is high, Myopic-IB can run out of inventory much earlier than RF and RFF. The exact time that Myopic-IB stocks out of all products is path-dependent. However, we observe that it occurs between the 81,000$^{\text{th}}$ and 82,000$^{\text{th}}$ customer arrival. Hence, in Figure O.2, we observe that Myopic-IB sells 100% of the inventory when the load factor is 1.2 or higher (i.e., 90,422 arrivals or more). In contrast, RF and RFF utilize arrival rate information, so their framing decisions result in sales occurring throughout the horizon.

O4. Robustness to randomness in the number of arrivals

Aside from the random sequence of arrivals, the total arrivals during the 12-week horizon can also be random. We next conduct experiments where both are random.

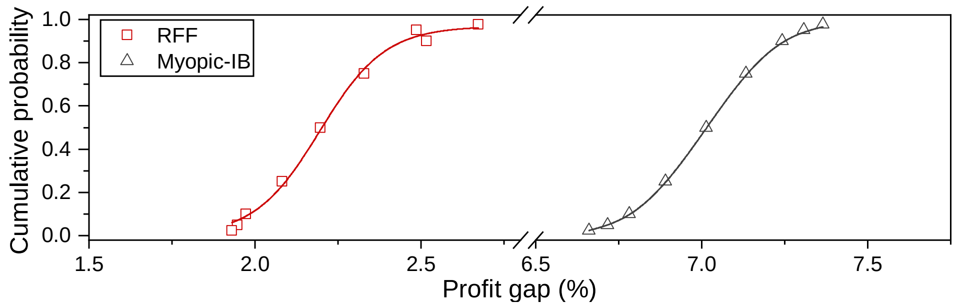


Figure O.3 Cumulative probability distribution of profit gap under RFF and Myopic-IB.

The total arrivals is a Poisson random variable with parameter λT, where we set $\lambda = 1$. Hence, the expected total arrivals is T, which we set to be equal to the total inventory (75,352 units). We use the parameter $T = 75,352$ to compute the RFF policy. To test the robustness of RFF against the number of arrivals, we run experiments where the actual total arrivals deviates from T.

We vary the total arrivals to be 74,814, 74,901, 75,000, 75,167, 75,352, 75,537, 75,704, 75,804, and 75,890. These values correspond to the following quantiles of the Poisson distribution: 2.5%, 5%, 10%, 25%, 50%, 75%, 90%, 95%, and 97.5%. For a fixed the number of arrivals, we randomly generate 30 sample paths with different sequences of customer arrivals. The 30 sample paths are used to compute the expected profit of RFF and its profit gap relative to the deterministic upper bound. We compute the profit gap at different quantiles of the total arrivals (Poisson distribution). Since the profit gap decreases with the number arrivals, our data allows us to estimate the cumulative probability distribution of the RFF profit gap. Note that the profit gap is random due the random total arrivals.

Figure O.3 displays the cumulative distribution of the RFF profit gap. For comparison, the figure also presents the distribution of the Myopic-IB profit gap. We observe that with 97.5% probability, the profit gap of RFF does not exceed 2.7%. In contrast, with probability 97.5%, the Myopic-IB policy has a profit gap larger than 6.7%. Recall that Myopic-IB ignores information about the arrival rate (i.e. T). Hence, we can infer from the figure that, even when RFF mis-specifies the arrival rate, utilizing an arrival rate estimate can lead to significant value in our setting.

O5. Limited Number of Changes to the Displays

We test a variant of RFF where we limit the number of times the policy can change the displays. The results are summarized in Table O.3. Note that, under the RFF policy, **DET'** in (10) is only solved once at the start of the simulation to construct the framing probabilities, then a new display is randomly chosen for each arriving customer according to these probabilities. However, in many realistic cases, firms may prefer that customers in a given location do not see different

Reopt?[1]	Framing changes[2]	Total sales[3] (units)	Revenue-per-sale ($)	Cost-per-sale ($)	Total profit ($ millions)	Profit gap
No	Unlimited	$66,406 \pm 48$	52.48 ± 0.02	8.97 ± 0.003	2.889 ± 0.002	2.20%
No	Weekly	$65,170 \pm 90$	52.67 ± 0.08	8.97 ± 0.005	2.848 ± 0.004	3.72%
Yes	Weekly	$66,954 \pm 72$	52.59 ± 0.03	8.97 ± 0.005	2.921 ± 0.002	1.14%

[1] If "Yes", **DET** is reoptimized weekly based on up-to-date inventory levels.
[2] If "Weekly", framing decision at each location is fixed for one week.
[3] The number of arrivals is equal to the total inventory of 75,325 (load factor is 1).

Table O.3 Performance of RFF variants: The table summarizes the profit gap, and the average (with 95% confidence intervals) of total sales, revenue-per-sale, cost-per-sale, and total profit.

displays if they arrive within the same short time interval. An interesting question then arises: if we use the constructed probabilities to randomly draw the framing assignment for a location, and this assignment is kept the same for all customers arriving within the same week, how would this affect the performance of RFF? As can be seen in Table O.3, this RFF variant has a profit gap that is not much larger than that of the basic RFF policy. This is surprising since the basic RFF policy allows 75,352 display changes whereas the RFF variant discussed above only allows 98 locations $\times$ 12 weeks $=$ 1,176 changes. We also test the variant in which **DET'** is re-optimized weekly based on the up-to-date inventory levels. Consistent with the results in Table 2, we find that re-optimization can improve profits simply by reducing the overall sales that would otherwise be lost due to the random fluctuations in demand.