Personalized Promotions in Practice: Dynamic Allocation and Reference Effects

JACKIE BAEK, WILL MA, and DMITRY MITROFANOV

Partnering with a large online retailer, we consider the problem of sending daily personalized promotions to a userbase of over 20 million customers. We propose an efficient policy for determining, every day, the promotion that each customer should receive (10%, 12%, 15%, 17%, or 20% off), while respecting global allocation constraints. This policy was successfully deployed to see a 4.5% revenue increase during an A/B test, by better targeting promotion-sensitive customers and also learning intertemporal patterns across customers.

We also consider theoretically modeling the intertemporal state of the customer. The data suggests a simple new combinatorial model of pricing with reference effects, where the customer remembers the best promotion they saw over the past ℓ days as the "reference value", and is more likely to purchase if this value is poor. We tightly characterize the structure of optimal policies for maximizing long-run average revenue under this model: they follow an "ℓ-up-1-down" cycle, where each time the offered discount worsens, the new value is repeated ℓ times to reset the customer's reference, and each time the discount improves, the better value is offered only once. As an example, if $\ell = 3$ and there are four distinct discount values 10%, 15%, 12%, 20% in order, then the corresponding ℓ-up-1-down cycle would offer (10%, 10%, 10%, 15%, 12%, 12%, 12%, 20%) repeatedly. This structural characterization allows us to reduce an exponentially-sized MDP to a polynomially-sized one, enabling the computation of the optimal policy. We also prove this characterization is tight: given any ℓ-up-1-down cycle, we can construct an instance for which that cycle is the unique optimal policy.

Contents

Author affiliations: *Stern School of Business, New York University,* baek@stern.nyu.edu (JB); *Graduate School of Business, Columbia University,* wm2428@gsb.columbia.edu (WM); *Carroll School of Management, Boston College,* dmitry.mitrofanov@bc.edu (DM)

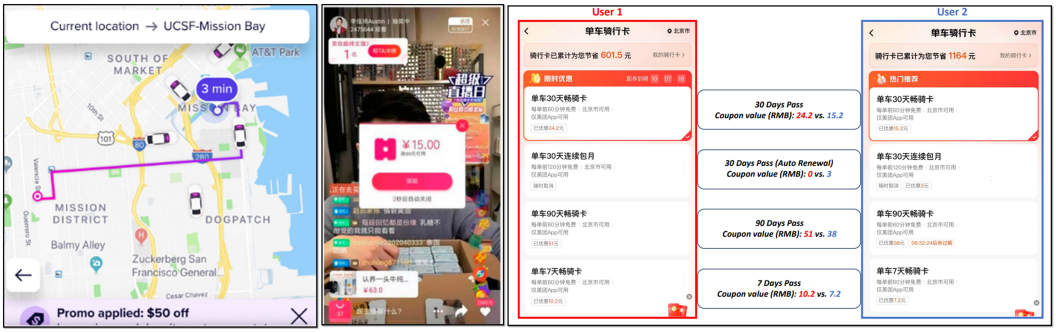

Fig. 1. Examples of personalized promotions on Lyft, Alibaba Livestream Shopping, and Meituan Bike.

1 Introduction

Personalized promotions have become increasingly prevalent, with Lyft offering them in its app as early as 2019 [Shmoys and Wang, 2019], Alibaba Livestream Shopping offering highly personalized deals in real-time to its users watching influencers shop [Liu, 2023], and Meituan explicitly discriminating in the Bike pass discounts offered to different users [Dai et al., 2024]. These practices are illustrated in Figure 1, and are generally allowed so long as they are not deceptive and do not violate anti-discrimination or competition rules. Personalized promotions fall under the grander movement toward gamified shopping, e.g. on Temu [Zhou, 2023], and AI-agent shopping assistants, e.g. on Alibaba Taobao and TMall [Utley, 2024], which provide highly personalized journeys for users on shopping platforms.

Optimizing these personalized promotions from the business's end is a challenging problem, due to context-dependent customer behavior, complex interactions with other (non-personalized) promotions running at the same time, and longer-term reference effects where customers may become insensitive to big discounts if they expect even better ones to come. Moreover, many promotion teams face budget constraints on the total discount redeemed by customers, which creates an allocation problem coupled across customers. Finally, the algorithm must be fast and scalable, because these promotions are sent at a high-frequency (e.g., daily, or in real-time as the user browses the app), and promotions may need to be computed at the individual user level.

To explore this personalized promotion problem, we partner with a large U.S. online retailer in the home decor sector, whose annual revenue is in the billions. They send a daily marketing email to their userbase, each containing a personalized coupon offering "X% off your entire order", where X is either 10%, 12%, 15%, 17%, or 20% and targeted toward the specific user. While the retailer's base prices are set at the product level, these personalized discounts serve as a customer-level tool to drive incremental purchases that would not occur at full price.

Our paper has two main parts. The first, outlined in Subsections 1.1 and 1.2, describes our transformation of the promotion algorithm at our partner retailer to use data and optimization, where we develop a fast and scalable algorithm with one "shadow price" parameter that is manually set every day based on the promotional budget that can be allocated. In an A/B test during May–June 2024 to 20 million customers, we saw a significant 4.5% uplift in revenue, compared to their incumbent algorithm that sent different discounts based on ad-hoc clustering instead of granular personalization. From this excellent result, our algorithm became the default for allocating personalized promotions at our partner retailer in August–September 2024.

The second part of our paper, outlined in Subsection 1.3, studies an abstract theoretical model with the goal of improving upon the deployed algorithm that myopically optimizes for next-day

revenue. Based on evidence from the data, we formulate a new model of pricing with reference effects, where the customer remembers the best promotion they saw over the past ℓ days as the "reference value". We combinatorially characterize the structure of optimal policies for this model, which allows us to computationally solve for optimal promotion cycles.

1.1 Practical Problem and Methodology (details in Section 2)

Data and decisions. The userbase consists of customers i with static features z_i^{Cust} (e.g., join date, shopping channel). Every day t, the retailer offers each customer i a personalized discount value $v_{it} \in \mathcal{V} := \{.10, .12, .15, .17, .20\}$. The customer then makes a purchase with pre-discount spend value $w_{it} \geq 0$ ($w_{it} = 0$ represents no purchase), while also exhibiting auxiliary behavior captured by features z_{it}^{Obs} (e.g., website visits, shopping cart activity, email opens). The objective is to maximize the long-run revenue

$$\sum_{t,i} (1 - v_{it}) w_{it}$$

while also facing a soft budget constraint on the total discounts redeemed $\sum_{t,i} v_{it} w_{it}$.

Solution approach. A customer i's spend w_{it} depends (randomly) on the day t, the personalized discount value v_{it}, as well as z_i^{Cust} and the entire history of $(v_{it'}, w_{it'}, z_{it'}^{\text{Obs}})_{t'<t}$. Because the exact spends w_{it} are noisy and difficult to predict, we focus on the binary purchase indicator $y_{it} = \mathbb{1}(w_{it} > 0)$ and normalize spend to $w_{it} = 1$ conditional on purchase. Maximizing long-run revenue under this assumption is still difficult, and not even formally-defined given the exogenous changes (e.g., the marketing team changing the e-commerce site) that could affect the purchase probabilities on future days. Therefore, every day t we myopically optimize the daily expected revenue

$$\sum_i (1 - v_{it}) \cdot \mathbb{E}[y_{it} | v_{it}].$$

A potential concern is that a myopic policy could assign the same discount to a customer repeatedly, a pattern which our partner preferred to avoid. However, we ensured that our model of $\mathbb{E}[y_{it}|v_{it}]$ depends sufficiently on past decisions $(v_{it'})_{t'<t}$ to induce natural variation in the offered promotions over time. Specifically, $\mathbb{E}[y_{it}|v_{it}]$ becomes less sensitive to v_{it} after a customer has received many strong discounts in recent periods, with customers becoming "complacent" to receiving good discounts (we provide empirical evidence of this in **Subsection 2.2**). In such cases, the myopic rule naturally shifts the customer toward smaller discounts, generating within-customer variation over time, which our partner viewed as a desirable property for deployment. This desirable variation, driven by customers' evolving sensitivity to discounts, inspires the reference effect model studied in the second part of this paper.

Training the model for $\mathbb{E}[y_{it}|v_{it}]$. We extract features to construct a model for $\mathbb{E}[y_{it}|v_{it}]$, accounting for the expert knowledge of our partner's machine learning team, while also ensuring there are features that depend on past decisions $(v_{it'})_{t'<t}$ to potentially induce the desired variation in discounts received. To elaborate, we construct a mapping ϕ that extracts important context from the feature information and history for a customer i on day t, resulting in the (processed) feature

$$x_{it} := \phi(t, z_i^{\text{Cust}}, (v_{it'}, w_{it'}, z_{it'}^{\text{Obs}})_{t'<t}).$$

We then solve a supervised learning problem where the goal is to predict y_{it} based on (x_{it}, v_{it}) over all historical i, t pairs. We impose structure on the prediction model to isolate how decision variable v_{it} affects the probability that $y_{it} = 1$, and find that the direction is intuitively correct (i.e., better discount v_{it} increases purchase probability) in 99.5% of historical i, t pairs (see **Subsection 2.2**). This approach lets us pool data across all customers, rather than fitting separate models for

each individual, while the processed feature representation enables a flexible mapping from x to purchasing behavior. This approach is inspired by a similar methodology for personalizing healthcare interventions [Baek et al., 2025].

Optimization method. After training our model, we let $q(x, v)$ denote the probability that it estimates for a customer with context x making a purchase if they are offered discount value v. On day t, we then send to each customer i

$$v_{it} \in \text{argmax}_{v \in \mathcal{V}}(1 - \lambda_t v) q(x_{it}, v), \tag{1}$$

where x_{it} is their current context, and $\lambda_t > 0$ is a penalty parameter for giving too much discount. The default value for λ_t is 1, in which case (1) corresponds to maximizing expected revenue, but λ_t can be decreased to give discounts more aggressively, or decreased to conserve the promotional budget. In practice at our partner retailer, λ_t is manually set each day t by an internal employee, who is in touch with upper management on the budget to be allocated across customers.

We highlight that our method is fast and scalable: the prediction function $q(x, v)$ once trained is fast to call, and the optimization problem (1) is separately solved for each customer. It is also easily tunable: λ_t can be changed each day to adapt to changing business conditions.

1.2 Deployment and Impact (details in Section 3)

Collaboration details. We were given older data to use academically to develop the algorithm. After testing on this data, we helped our partner write production-level code both for training a model from their most-recent data and for optimizing daily based on the trained model. The model is re-trained periodically, although the most-recent data is not shared with us. This code was deployed in an A/B test during May–June 2024, and aggregate, relative results were shared with us to report. In addition, a small group of customers were reserved to receive independently random discount values every day, and this data was shared for the purpose of further academic testing.

Results from A/B test. In an A/B test on over 20 million customers that were randomly split 50/50 into treatment/control, our algorithm saw a massive 4.5% increase in average revenue per user. The p-value of such an observation if our algorithm was not better is < 0.01. We delve into where the improvement is coming from, and find that our algorithm is able to correctly predict sensitivity to discounts based on a customer's intertemporal state, and thus target the right customers—offer big discounts to customers who need incentive to make a purchase, and send stingy discounts to customers who were going to purchase anyway. Our algorithm yields a more polarized promotion distribution, where it sends either small or large discounts to most customers, compared to the incumbent algorithm that offered more medium discounts.

In terms of who gets the best discounts, aside from the reference effect considerations, our algorithm tends to offer them to the engaged customers who open the most emails, as it believes that these customers are looking for bargains. Thus, our algorithm drives the right incentives for customers to engage with the company's emails. Our algorithm was rolled out to all customers and became the default algorithm for sending personalized promotions during August–September 2024.

1.3 Theoretical Model and Results (details in Section 4)

Our deployed algorithm optimizes myopically each day, for simplicity, scalability, and to cope with uncontrollable exogenous factors such as arbitrarily changing budgets from day to day. We now formally study the long-run optimization problem in a theoretical model with a single customer whose intertemporal context is one-dimensional, and no exogenous changes or budgets. Our model is justified by our data, and its optimal solution provides an explanation for promotion cycling in practice as well as provides structure for how it should be optimized.

Reference value model. We study the problem of maximizing long-run average expected revenue from a single customer i:

$$\sup_{(v_{it})_{t=1}^{\infty}} \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} (1 - v_{it}) q(x_{it}, v_{it})$$

with a one-dimensional context $x_{it} := \max\{v_{i,t-\ell}, \ldots, v_{i,t-1}\}$. This should be viewed as a "reference value", where the customer remembers the best promotion they received over the past ℓ days, for some fixed positive integer ℓ. It is assumed that this is the sole intertemporal state affecting the customer's purchase probabilities $q(\cdot, v)$ under different discounts v, allowing us to construct a deterministic Markov Decision Process (MDP) to model the customer.

We impose only one mild assumption on the function $q(\cdot, v)$, that it is *reference-monotone*: fixing any offered discount $v \in \mathcal{V}$, the purchase probability $q(x, v)$ is decreasing in x. That is, the bigger the customer's reference value x, the more complacent they are to discounts and less likely they are to purchase under any offered v.

Model comparison and justification. In related literature on dynamic pricing with reference effects (reviewed in Subsection 1.4), the reference price is often defined as the (exponentially-weighted) average of past prices. Our definition based on the extremum over a fixed memory length ℓ yields a more combinatorial model with only $|\mathcal{V}|$ possible reference values, which will allow us to derive cleaner cycle structure results than most of this literature.

We also provide some empirical evidence for our definition in **Subsection 3.1**, using the data in which customers were sent random discounts. We find that the reference value $\max\{v_{i,t-\ell}, \ldots, v_{i,t-1}\}$ is negatively correlated with purchases ($y_{it} = 1$), providing empirical evidence of the reference effect. We also find that our definition of the reference value is a better univariate predictor for purchases than using the average coupon value over the past ℓ days, $\frac{1}{\ell}(v_{i,t-\ell} + \cdots + v_{i,t-1})$.

Characterization of optimal policies. For our reference value model, the problem of maximizing long-run average can be formulated as an infinite-horizon undiscounted MDP with $|\mathcal{V}|^{\ell}$ states, where the state needs to include the sequence of past ℓ values offered. A priori, this exponential-sized MDP is computationally intractable to solve. Fortunately, we are able to identify a key "ℓ-up-1-down" structural result that both provides intuition about optimal cycle structure and allows us to solve this infinite-horizon undiscounted MDP problem in polynomial time.

To elaborate, an "ℓ-up-1-down" policy is defined by a cycle of *distinct* values in $\mathcal{V}$, which we call its *generator cycle*. For each value in the generator cycle, we check if it results in a higher price than the previous value (i.e., the discount is worse)—if so (i.e., price is going "up"), then we repeat the value ℓ times; if not (i.e., price is going "down"), then we only offer the value once. The final policy is to offer this cycle with repeats on the "up" values, ad infinitum. For example, if $\mathcal{V} = \{.10, .12, .15, .17, .20\}$ and $\ell = 3$, then generator cycle (.15 .12 .20) would lead to the policy cycling between values (.15 .15 .15 .12 .12 .12 .20), offering the best discount of .20 once a week. As another example, generator cycle (.10 .15 .12 .20) would lead to the policy cycling between values (.10 .10 .10 .15 .12 .12 .12 .20). The intuition behind ℓ-up-1-down policies is that if we were going to make the price go up, then the purpose is to "reset" the reference value of the customer, in which case we need to offer the higher price ℓ times.

We prove that under the reference-monotonicity assumption, there always exists an ℓ-up-1-down policy that is optimal. This drastically reduces the search space, noting that many cycles are not ℓ-up-1-down—e.g. if $\ell = 3$, then (.10, .20) is not ℓ-up-1-down because the .10 is not repeated 3 times; (.10, .10, .10, .20, .20, .20) is not ℓ-up-1-down because the .20 should not be repeated; and (.10, .10, .10, .15, .15, .15, .20, .15) is not ℓ-up-1-down because (.10, .15, .20, .15) is not a valid generator

cycle, as it contains the duplicate value .15. To find an optimal policy, we only need to search over generator cycles with distinct values, which we show can be formulated as maximizing infinite-horizon undiscounted reward in a reduced MDP with only $|\mathcal{V}|$ states, a problem solvable in polynomial time [Puterman, 2014].

Proof technique. General theory about finite deterministic MDP's allows for a reduction to stationary deterministic policies defined by cycles, but this is not enough. Leveraging reference-monotonicity, we further derive a sequence of transformations, none of which worsens the long-run average objective, that allows any policy to be eventually converted into an ℓ-up-1-down cycle. To the best of our understanding, this requires a non-trivial argument that also uses the specific combinatorial structure of our MDP with the "max over past ℓ" reference value. Along with the proof, we provide an example illustrating the variety of transformations that may be needed.

Tightness. We also prove that ℓ-up-1-down is a tight characterization of optimal policies, in that for any generator cycle of distinct values in $\mathcal{V}$, there exists a reference-monotone instance for which the ℓ-up-1-down policy implied by that generator cycle is the unique optimal policy.

1.4 Related Work

Dynamic pricing with reference effects. Early works used numerical methods to optimize price sequences under reference effects, e.g. for peanut butter [Greenleaf, 1995]. Since then, structural results have been established showing the non-necessity of dynamic pricing when customers are loss-averse or loss-neutral [Nasiry and Popescu, 2011, Popescu and Wu, 2007]. Outside of these settings, papers that have tried to formally optimize price sequences found the problem to be generally difficult and exhibit little structure [Chen et al., 2017, Fibich et al., 2003, Hu et al., 2016].

By contrast, our theoretical result establishes a clean structure for optimal price sequences, that moreover allows for efficient computation. This stems from our model being combinatorial than most of these works, using a discrete but general demand function (instead of a specific functional form), a "peak-end" reference model based on the maximum/minimum in recent memory (instead of exponential smoothing), and a long-run steady state objective (i.e., infinite-horizon undiscounted instead of finite-horizon or infinite-horizon discounted). We note that a combinatorial, graph-theoretic approach to dynamic pricing with reference effects was also taken in Cohen et al. [2020], but they do not derive structural results for optimal policies.

The "peak-end" reference model we study has also been considered in Cohen et al. [2020], Cohen-Hillel et al. [2023], Nasiry and Popescu [2011], with Cohen-Hillel et al. [2023] establishing optimality of a high-low pricing structure under certain conditions. Nonetheless, their model is about pricing multiple products under business constraints and quite different in nature from our parsimonious theoretical model, and their structure is also different from our characterization of ℓ-up-1-down price sequences.

Finally, our paper provides some data-driven evidence of this "peak-end" reference model in the context of personalized promotions (see Subsection 3.1), complementing the aforementioned papers, which reference psychology evidence about people remembering peak experiences [Kahneman et al., 1993]. For earlier papers showing empirical evidence of the reference price effect in general, we refer to the survey by Mazumdar et al. [2005]. We should note however that our paper is motivated by *personalized* reference effects, which may be starkly different from estimating reference effects for aggregate demand [Hu and Nasiry, 2018]. Jiang et al. [2024] consider customer heterogeneity when estimating reference effects.

Recently, the literature on dynamic pricing with reference effects has also incorporated having multiple products with logit demand [Guo et al., 2025], and demand function learning [Agrawal and Tang, 2024, den Boer and Keskin, 2022]. At an intuitive level, our ℓ-up-1-down policies that

offer the unattractive value ℓ times to "reset" the reference resembles the ResetRef operation in Agrawal and Tang [2024].

Revenue management with repeated engagements. Our "ℓ-up-1-down" characterization is more similar to the theoretical results derived in the literature on "intertemporal" price discrimination with forward-looking [Besbes and Lobel, 2015] or patient [Liu and Cooper, 2015] customers—see also Lobel [2020], Wang [2016]. However, our state is determined by *past* instead of future prices, and our model is in some sense simpler because it is personalized for a single customer instead of dependent on population arrival rates. This allows for a more exact characterization of optimal price cycles compared to most results in this literature (which only provide a bound on cycle length); we are also able to prove our "ℓ-up-1-down" characterization is tight (see Theorem 4.10).

More broadly, our work relates to revenue management under models of repeated engagements with the customer base or an individual customer. To this end, Calmon et al. [2021] study a model where current prices affect customer goodwill, which in turn affects the future budget they spend on the platform. Freund and Hssaine [2021] study a model where users can be given rewards at each round to stay on the platform, but these rewards must satisfy fairness constraints. Jiang et al. [2022] study a model where if the current price is too high, then customers can register their willingness-to-pay and be alerted if/when the price drops below their registered price. Lei et al. [2023] study how online service providers such as gaming platforms should rotate content to maximize user engagement. Chang et al. [2024] study a model where users form usage habits, and pricing can be used to both maximize revenue and curb addiction. Baek et al. [2026] empirically study the impact of frequent user notifications on short-term revenue and long-term unsubscription risk.

Personalized promotions in practice. Our work is based on a real-world deployment of personalized promotions, which can be compared to the deployments in the following works. Personalized promotion allocation under a promotional budget is formulated as a knapsack problem in Albert and Goldenberg [2022], Shmoys and Wang [2019], which describe real-world applications at Lyft and Booking.com respectively. Our application is similar, but approach is different, as we propose a single "shadow price" parameter to (approximately) solve the allocation problem instead. A similar method is in fact deployed to allocate personalized coupons at Meituan [Dai et al., 2024]. Jagabathula et al. [2022] propose a DAG-based framework for personalized retail promotions, showing how retailers can tailor offers to individual customers in ways that better align with their tastes. Finally, due to the challenges in cleanly modeling intertemporal customer state, Liu [2023] proposes using Deep Reinforcement Learning to black-box learn the best personalized promotion policies for the long-term, which was deployed in real-time at Alibaba Livestream Shopping. However, we were forced to use a more structured and interpretable approach at our partner retailer, which allows management to see which features cause customers to receive better promotions (see Subsection 2.2).

2 Details of Practical Methodology

We provide details of our practical methodology outlined in Subsection 1.1.

We use historical data from 360,000 customers over 90 days. The raw dataset consists of time-stamped customer-level activity logs that track marketing exposure and subsequent engagement and shopping behavior over time. In particular, the data includes whether and when each customer received marketing emails and coupons, and whether those emails were opened or clicked. The dataset also includes transaction history (purchase occurrence, purchase amounts, and other order-level details) and on-site browsing behavior, such as page views, cart views, and checkout-related activity. We note that the promotion decisions in the historical dataset were made by the ad-hoc

incumbent algorithm, which does not have access to any features unobservable to us. Therefore, we do not expect any confounding as long as we control for the observable features.

In collaboration with our partner, we processed the raw data to generate a list of features to use for our estimation model, to capture the majority of the relevant information that may influence a customer's purchasing behavior. We include information about customers' purchasing and browsing activity (recent purchase volume and transaction counts, site and cart views, and cart composition and discounts), their engagement with emails (opens and clicks over multiple recency windows), their coupon exposure and interaction history (coupon values received, and summary statistics of coupons received, opened, and clicked), as well as a day-of-week indicator. Combined, these processed variables yield a total of 61 features, listed in Appendix A.

2.1 Estimation Model and Method

Let $\mathcal{D}$ be the dataset of tuples (x, v, y) of processed features, coupon values, and purchase indicator respectively. The size of dataset $\mathcal{D}$ is approximately $360,000 \times 90$ (one for each customer, day pair). We use $\mathcal{D}$ to estimate $q(x, v)$, the probability that a customer with processed features x makes a purchase when offered coupon value v.

One approach to learn $q(x, v)$ would be to fit a flexible black-box machine learning model that maps (x, v) to y. However, since our downstream goal is to optimize the choice of v rather than to maximize predictive accuracy for y, we need a reliable estimate of how *changes* in v impact y. A black-box approach can yield unstable or even non-monotone responses to changes in v. Therefore, we impose a structure on $q(x, v)$ that makes the dependence of v explicit, where we decompose the purchase probability $q(x, v)$ into a baseline component $\alpha(x)$ and a coupon-sensitivity component $\beta(x)$. Specifically, letting $\sigma(y) := 1/(1 + \exp(-y))$ denote the logistic function, we impose the structure

$$q(x, v) = \sigma\big(\alpha(x) + (v - 0.15)\beta(x)\big), \tag{2}$$

for learned functions $\alpha(\cdot)$ and $\beta(\cdot)$. Here, $\alpha(x)$ captures baseline purchase propensity at the "nominal" coupon value of 0.15, and $\beta(x)$ captures coupon sensitivity: $\beta(x) = 0$ implies no effect of the coupon value v on purchase probability, while larger $\beta(x)$ implies stronger responsiveness to changes in v.

We note that the structure (2) imposes that conditional on x, the coupon value v affects purchase probability *only* through the linear term $(v - 0.15)\beta(x)$, so the marginal effect of increasing v is governed by a single term $\beta(x)$ that does not vary with v. While coupon sensitivity could depend on v in practice, this restriction serves as regularization that enforces a monotone response in v and yields a more stable and interpretable estimate of coupon sensitivity.

We estimate $\alpha(\cdot)$ and $\beta(\cdot)$ using $\mathcal{D}$ via a two-step procedure: we first learn $\alpha(x)$ flexibly, then learn $\beta(x)$ under a linear structure. That is, using the subset of $\mathcal{D}$ consisting only of observations with a coupon value of $v = 0.15$, we first fit a gradient-boosting classifier to predict y from x. Denoting this classifier by $\mathrm{GB}(x)$ as an estimate of $q(x, 0.15)$, and using the fact that $q(x, 0.15) = \sigma(\alpha(x))$, we set $\alpha(x) = \sigma^{-1}(\mathrm{GB}(x))$.

Then, we use the learned $\alpha(x)$ to estimate $\beta(x)$. We impose a linear structure $\beta(x) = \beta^\top x$ (where $\beta, x \in \mathbb{R}^{61}$), which allows us to write

$$q(x, v) = \sigma\big(\alpha(x) + \beta^\top (v - 0.15)x\big) = \sigma\left(\begin{pmatrix}1\\ \beta\end{pmatrix}^\top \begin{pmatrix}\alpha(x)\\ (v - 0.15)x\end{pmatrix}\right).$$

That is, the term inside the logistic function has a linear dependence on the features $\alpha(x)$ and $(v - 0.15)x$. For each sample $(x, v, y) \in \mathcal{D}$, we compute the corresponding features $\alpha(x) = \sigma^{-1}(\mathrm{GB}(x))$ and $(v - 0.15)x$, and then we estimate β using logistic regression on these transformed features.

Feature	Sign of Coefficient
(a) # emails clicked in the last 28 days	+
(b) # shopping cart visits in the last 3 days	+
(c) # shopping cart visits in the last 7 days	+
(d) Average site sale discount in the cart	+
(e) Average percentage of all historical orders where a coupon was used	+
(f) Average coupon use percentage of all orders in the last 30 days	+
(g) Average coupon use percentage of all historical orders	+
(h) Average coupon discount of a coupon clicked in the last 7 days	+
(i) Average coupon discount of a coupon clicked in the last 30 days	+
(j) Maximum coupon discount received in the last 7 days	−

Table 1. The ten most predictive features for β. A positive coefficient implies that a high value for the feature is associated with having a higher sensitivity to the coupon value.

2.2 Empirical Observations of Estimated Model

We fit the model (2) to historical customer-day observations and summarize two empirical patterns.

First, across historical customer-day pairs, we find that the estimated coupon-sensitivity term $\beta(x)$ is positive for 99.5% of samples. This is consistent with the basic monotonicity expectation that a larger discount should weakly increase purchase probability.

Next, to understand which features are most strongly associated with $\beta(x)$, we standardize the 61 processed features and fit an auxiliary L1-regularized logistic regression, tuning the penalty to obtain a sparse set of predictors. Table 1 reports the ten most predictive features and the signs of their coefficients, where a positive sign indicates that larger values of the feature are associated with higher coupon sensitivity.

Features (a)–(c) capture recent engagement, through either email interactions or active cart activity. The positive coefficients imply that customers who were more recently engaged with emails or their shopping cart are more likely to respond to a better discount. Features (d)–(i) capture past reliance on discounts, including frequent coupon use and recent interaction with higher-value coupons. Customers with a high value of these features are ones constantly looking for good discounts, and hence they are not likely to make a purchase with a low-value coupon (i.e., they are price-sensitive customers). Our model associates these features with higher coupon sensitivity.

Finally, feature (j) has a negative sign: customers who recently received a high-value coupon tend to be less responsive to additional increases in v. This represents the reference price effect, where receiving a large discount in the recent past decreases the customer's reference price that they need to pay, which makes a subsequent discount less effective. This mechanism also mitigates repetition in the assigned coupons. If a customer recently received a good discount, their coupon sensitivity decreases, and hence the myopic policy naturally shifts the customer toward worse offers, generating within-customer variation over time.

We provide further evidence of this reference effect in Subsection 3.1. The fact that the *maximum-value* recent coupon is the key predictor of diminished responsiveness motivates our theoretical model in Section 4, where the reference value is defined as the maximum discount in recent memory.

2.3 Optimization Details

Recall from (1) that every day t, our algorithm sends to each customer i the discount value $v_{it} \in \mathcal{V}$ maximizing $(1 - \lambda_t v_{it}) q(x_{it}, v_{it})$, where λ_t is a parameter. We now elaborate on the tuning of λ_t.

Let W denote the average pre-discount spend in a single shopping cart checkout, over all historical orders. For any value of λ_t, we can compute the implied decisions $(v_{it})_i$, under which the expected discount redeemed is

$$\sum_i v_{it} W q(x_{it}, v_{it}). \tag{3}$$

The value of (3) is compared to a promotional budget B_t. If it does not exceed B_t, then the discount values $(v_{it})_i$ are sent to the customers. Otherwise, λ_t is increased until (3) does not exceed B_t.

Proposition 2.1 (proven in Appendix B). *Suppose $q(x_{it}, v)$ is weakly increasing in v for all i. Then* (3) *is weakly decreasing in parameter λ_t.*

Recall from Subsection 2.2 that the predicted purchase probability $q(x_{it}, v)$ is increasing in v for 99.5% of contexts x_{it}, due to the non-negative sign of $\beta(x_{it})$. Therefore, we can essentially think of (3) as being decreasing in λ_t in practice.

At our partner retailer, an internal employee first evaluates (3) under $\lambda_t = 1$ (which would maximize revenue), comparing it to the promotional budget B_t for the day t. If (3) exceeds B_t, then they use bisection search to increase λ_t to the smallest value for which (3) is no greater than B_t.

We note that we tried to get a better prediction of the total discount redeemed by having W depend on the customer i or the discount value v_{it}. However, this failed because exact spend amounts are highly idiosyncratic, which is why we focus on predicting purchase probabilities.

3 Details of Deployment and Impact

An A/B test was ran during May–June 2024, in which 20 million customers were randomly split 50/50 into treatment/control. Customers in the treatment group received personalized discount values determined by our algorithm, while customers in the control group received discount values determined by the incumbent algorithm that had been in production for a couple of years. Once a customer was assigned to either the treatment or control groups, their assignment remained unchanged throughout the experiment.

Overview of incumbent algorithm. The incumbent algorithm focused on what is the correct *distribution* of promotions, i.e. what fraction of customers should receive each of the discounts 10%, 12%, 15%, 17%, and 20%, which could depend on the promotional budget. Meanwhile, the customers were clustered in an ad-hoc fashion, based mostly on tenure and average spend. Customers in the same cluster would generally be sent the same discount value, with arbitrary adjustments as needed to fit the desired promotion distribution. Additional heuristics were inserted to ensure that each customer saw variation in the discounts they received over time.

Overall, the incumbent algorithm processed data manually, to create coarse clusters of customers, based on static features. By contrast, our algorithm used machine learning on the data, to create personalized prediction models for each customer, that also accounted for their intertemporal state including reference effects. To fairly compare our algorithm to the incumbent, the parameter λ_t in our algorithm was tuned daily to match the promotional budget of the incumbent algorithm.

Results. Aggregate, relative results over an 11-day period from the A/B test were shared with us, and displayed in Figure 2. During this 11-day period, our algorithm's prediction model was not re-trained. Our algorithm optimizes for revenue, and we indeed saw a 4.5% higher total revenue over the 11-day period in the treatment group compared to the control group, which have the same size. We also estimate the average treatment effect by regressing each customer's 11-day total revenue on a treatment indicator and conducting a one-sided test using heteroskedasticity-robust standard errors, and observe a p-value less than 0.01 for the null that our algorithm does not increase average revenue.

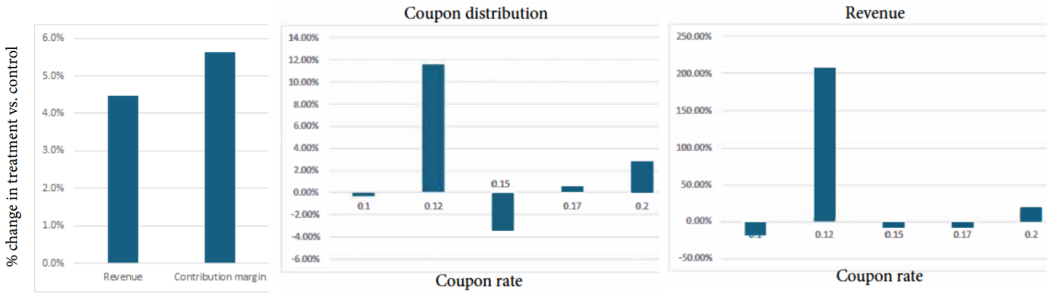

Fig. 2. Deployment results: Left chart is overall % change in revenue and profit between treatment vs. control; Middle chart is % change in total number of times each coupon rate was sent; Right chart is % change in total revenue from coupons of each rate being redeemed.

Our algorithm does not account for cost of goods sold (i.e., the wholesale price our partner retailer paid for its inventory), but we see an even bigger 5.6% increase in contribution margin (i.e., profit). This suggests that our promotion targeting could be enticing the customers to substitute to higher-end goods which have higher margins, although we did not formally test this.

Regardless, the stark results on the Left chart of Figure 2 fully demonstrate the power of machine learning, granular personalization, and intertemporal state—three new elements that our algorithm introduced to our partner's business. We also emphasize that these numbers pertain to overall revenue and contribution margin, without any subdivision of the goods or customers, which was very attractive to our partner's management. We completely changed how they think about allocating coupons, where instead of directly optimizing the distribution of coupon values sent, one should learn individual customer models and personalize coupon values, with the distribution of promotions being a by-product of the individual-level optimization.

Where is the money coming from? The Middle chart of Figure 2 shows how our algorithm changed the distribution of promotions. In particular, there is an 11.56% increase in the number of 0.12 coupons sent, and a 2.91% increase in the number of .2 coupons sent, mostly at the expense of a 3.44% decrease in the number of .15 coupons sent. Overall, our algorithm seems to think that the 0.12 coupon rate strikes a nice balance, where it is not as unattractive to the customer as the 0.1 rate, but preserves more revenue than the 0.15 rate.

The Right chart of Figure 2 shows massive changes in the total revenue from coupons of each rate being redeemed, suggesting that our algorithm is significantly shifting which customers receive each coupon rate, and overall enticing more purchases. In particular, there is a 208% higher (i.e., tripled) revenue from 0.12 coupons being redeemed, with only an 11.56% increase in the number of 0.12 coupons sent, suggesting that our algorithm is correctly targeting this smaller discount toward customers who were going to make a purchase regardless of the coupon rate. At the same time, there is a 20.5% higher revenue from 0.2 coupons being redeemed, with only a 2.91% increase in the number of 0.2 coupons sent, suggesting that our algorithm is also correctly targeting the best discount at customers who need the bargain to make a purchase.

Which customers receive the best promotions? Generally, the structure of our estimation and optimization is such that every day t, customers i in an intertemporal state x_{it} with a higher value of $\beta^\top x_{it}$ tend to receive a bigger discount v_{it}. Looking back at Table 1, this suggests that engaged customers who have redeemed more coupons in the past would receive better coupons in the future.

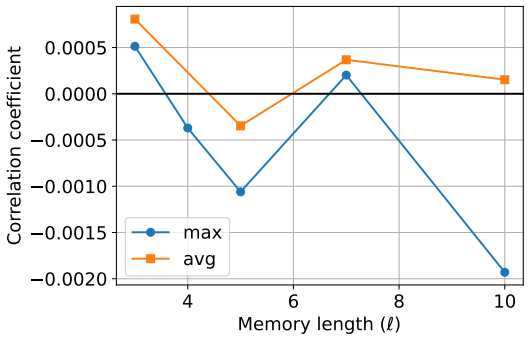

Fig. 3. Correlation between reference coupon metrics and the purchase indicator y_{it} for various memory lengths ℓ. The reference metric "max" uses $\max\{v_{i,t-\ell}, \ldots, v_{i,t-1}\}$, and "avg" uses $\frac{1}{\ell}(v_{i,t-\ell} + \cdots + v_{i,t-1})$. We note that the absolute magnitude of the correlation coefficient is generally small because over 99.9% of y_{it} values are 0.

However, this may not happen if they have already received the biggest coupon in the past 7 days, inducing some natural cycling in coupons received even for these customers.

3.1 Empirical Evidence for Theoretical Model

We provide empirical support for the key ingredients of our theoretical model (Section 4), including reference effects and the reference-monotonicity assumption. We leverage data from a group of 150,000 customers who, for an 11-day period during the A/B test, were assigned an independent random coupon each day, drawn uniformly from $\{0.12, 0.15, 0.17, 0.20\}$. The randomized assignment allows us to directly measure correlations without controlling for any factors.

Evidence of reference effect. For various memory lengths ℓ, we define the reference value for a customer i on day t as $\max\{v_{i,t-\ell}, \ldots, v_{i,t-1}\}$, and compute its empirical correlation with the purchase indicator y_{it} across customer-day observations from the randomized dataset. Figure 3 shows that this correlation is generally negative across ℓ, suggesting that a bigger recent-best discount (a bigger reference value) is associated with a lower propensity to purchase, consistent with a reference effect under our definition.

As a comparison, we also consider an average-based reference metric, $\frac{1}{\ell}(v_{i,t-\ell} + \cdots + v_{i,t-1})$. The corresponding correlations are not as strong or negative, which suggests that the maximum over recent history is a more predictive univariate proxy for intertemporal effects than the average.

Evidence for reference-monotonicity. Next, we test the reference-monotonicity assumption that, under any discount v_{it} offered today, purchase propensity is higher when the reference discount is smaller. To do this, we group coupon values into *small* discounts $(0.12, 0.15)$ vs. *large* discounts $(0.17, 0.20)$. For each case of v_{it} being small or large, we consider different memory lengths $\ell \in \{3, 4, 5, 7\}$, and report the percent change in purchase rate when the reference $\max\{v_{i,t-\ell}, \ldots, v_{i,t-1}\}$ lies in the small set $(0.12, 0.15)$ instead of the large set $(0.17, 0.20)$. Table 2 shows that the change is positive in nearly all cases, providing support for the reference-monotonicity assumption.

4 Theoretical Model and Results

We provide the theoretical model and results outlined in Subsection 1.3. To be consistent with the literature on dynamic pricing with reference effects, just for this section, we switch the language from offering a sequence of *discounts* (where bigger is better for the customer) to offering a sequence

ℓ	$v_{it} \in \{0.12, 0.15\}$	$v_{it} \in \{0.17, 0.20\}$
3	+7.1%	-1.1%
4	+30.1%	+16.5%
5	+70.3%	+55.9%
7	+137.1%	+12.6%

Table 2. Percent changes in purchase rate when the reference value $\max\{v_{i,t-\ell}, \ldots, v_{i,t-1}\}$ lies in $\{0.12, 0.15\}$ instead of $\{0.17, 0.20\}$, reported separately for different memory lengths ℓ and different values of v_{it}.

of *prices* (where lower is better for the customer). Therefore, the reference value is defined to be the *minimum (instead of maximum)* of recent offerings. We omit the index i of the single customer.

4.1 Model and MDP Formulation

Model. There is a finite set of feasible prices $\mathcal{P}$ from which an infinite sequence of prices $(p_t)_{t=1}^{\infty}$ is offered to the customer ($\mathcal{P} = \{.80, .83, .85, .88, .90\}$ would correspond to the discount values $\mathcal{V} = \{.10, .12, .15, .17, .20\}$ from our partner retailer). At each time t, the *reference* r_t of the customer is $\min\{p_{t-\ell}, \ldots, p_{t-1}\}$, the best price seen in the ℓ previous time steps (if $t \leq \ell$, then r_t is understood to be $\min\{p_1, \ldots, p_{t-1}\}$, with $r_1 = \overline{p}$ where $\overline{p} := \max\{p : p \in \mathcal{P}\}$ denotes the highest price in $\mathcal{P}$). Here ℓ is a positive integer denoting the *memory length* of the customer.

We note that both the price p_t and the reference r_t always lie in $\mathcal{P}$. We let $g(r, p)$ denote the immediate "gain", e.g. revenue, from offering price $p \in \mathcal{P}$ to the customer when their reference is $r \in \mathcal{P}$. We allow for an arbitrary gain function g as long as it satisfies the following assumption.

Definition 4.1. A gain function $g : \mathcal{P}^2 \to \mathbb{R}$ is said to be *reference-monotone* if for any fixed $p \in \mathcal{P}$, the gain $g(r, p)$ is weakly increasing in r.

Definition 4.1 is a mild assumption that is almost axiomatic in reference price models, where regardless of what price p the seller plans on offering, they are better off if the reference price r they are competing against is higher (and hence worse in the customer's mind). Importantly, we do not make any assumptions on how $g(r, p)$ changes with p, which allows for different objectives such as market share (in which case $g(r, p)$ is generally decreasing in p) or revenue (in which case $g(r, p)$ is often increasing and then decreasing in p).

MDP formulation. The problem of maximizing long-run average gain

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} g(r_t, p_t) \tag{4}$$

can be formulated as the following MDP. The state space is $\mathcal{P}^\ell$, with a state $s = (s^1, \ldots, s^\ell)$ denoting the ℓ previous prices seen by the customer, where s^ℓ is most recent. (Although the reference only depends on the minimum of the ℓ previous prices, the evolution of this minimum depends on the exact sequence of ℓ previous prices, which must all be tracked in the state.) The action space from any state is $\mathcal{P}$, denoting the next price to offer. When any action $p \in \mathcal{P}$ is taken from any state $s \in \mathcal{P}^\ell$, the reward is $g(\min\{s^1, \ldots, s^\ell\}, p)$, and the state deterministically transitions to $(s^2, \ldots, s^\ell, p)$. The starting state is $s_1 = (\overline{p}, \ldots, \overline{p})$.

Standard theory for finite MDP's [Puterman, 2014] confirms that (4) can be maximized using a policy which is stationary and deterministic, defined by a mapping $\pi : \mathcal{P}^\ell \to \mathcal{P}$ specifying the price to offer from any state. However, optimizing over such policies is still difficult, because our MDP is exponential-sized. Therefore, we consider the following reformulation instead.

Reduction to optimization over price cycles. Because the transitions are deterministic in our MDP, a stationary deterministic policy π induces a periodic steady state, where for some cycle length $c \leq |\mathcal{P}|^\ell$, there exists a time T_0 after which the states visited will cycle between $s(0), \ldots, s(c-1)$. The prices offered would cycle between $\pi(s(0)), \ldots, \pi(s(c-1))$, which we denote using $\pi_0, \ldots, \pi_{c-1}$ respectively. Given $\pi_0, \ldots, \pi_{c-1}$, the cycle of states can be reconstructed to be $s(t) = (\pi_{t-\ell \mod c}, \ldots, \pi_{t-1 \mod c})$ for all $t = 0, \ldots, c-1$ (noting that $c < \ell$ is possible), and hence the objective (4) (irrespective of starting state) equals

$$\frac{1}{c} \sum_{t=0}^{c-1} g\Big(\min\{\pi_{t-\ell \mod c}, \ldots, \pi_{t-1 \mod c}\}, \pi_t \Big). \tag{5}$$

4.2 Characterization of Optimal Policies

Maximizing objective (5) over positive integers c and prices cycles $(\pi_0, \ldots, \pi_{c-1}) \in \mathcal{P}^c$ is still computationally challenging, because a priori, c could be as long as $|\mathcal{P}|^\ell$, with the same price appearing multiple times in different contexts. For an arbitrary gain function $g : \mathcal{P}^2 \to \mathbb{R}$, this could indeed be the case (see Example 4.6 below). However, under the reference-monotonicity assumption, our main result drastically reduces the search space, to "ℓ-up-1-down" price cycles.

Definition 4.2 (Notation). We hereafter represent prices cycles using strings over the finite alphabet $\mathcal{P}$, and let p^k denote the concatenation of k copies of token p, for any positive integer k and $p \in \mathcal{P}$. For example, if $\mathcal{P} = \{1, 2, 3\}$, then 12^3 denotes the string 1222, which is the price cycle with $c = 4$ and $(\pi_0, \ldots, \pi_3) = (1, 2, 2, 2)$. We note that this cycle can be equivalently represented using the strings 2221, 2212, or 12221222.

Definition 4.3. A price cycle is said to be *ℓ-up-1-down* if it can be represented by a string of the form $\rho_0^{k_0} \cdots \rho_{d-1}^{k_{d-1}}$, where $d \leq |\mathcal{P}|$ is a positive integer, $\rho_0, \ldots, \rho_{d-1}$ are *distinct* prices in $\mathcal{P}$, and

$$k_t = \begin{cases} \ell, & \rho_t > \rho_{t-1 \mod d} \\ 1, & \rho_t < \rho_{t-1 \mod d} \end{cases} \qquad \forall t = 0, \ldots, d-1.$$

We call $(\rho_0, \ldots, \rho_{d-1})$ the *generator cycle*, with length d. The price cycle itself has length $c = \sum_{t=0}^{d-1} (1 + (\ell - 1)\mathbb{1}(\rho_t > \rho_{t-1 \mod d}))$.

Example 4.4 (Examples of ℓ-up-1-down Price Cycles). Let $\mathcal{P} = \{1, 2, 3\}$ and $\ell = 3$. Then 1222, 1222333, and 13332 represent ℓ-up-1-down price cycles, induced by generator cycles $(\rho_0, \rho_1) = (1, 2)$, $(\rho_0, \rho_1, \rho_2) = (1, 2, 3)$, and $(\rho_0, \rho_1, \rho_2) = (1, 3, 2)$ respectively. Meanwhile, 12 does not represent an ℓ-up-1-down price cycle because the "2" does not have ℓ copies despite being greater than 1. Also, 111222 does not represent an ℓ-up-1-down cycle because the "1" has multiple copies despite being less than 2. Finally, 12223332 has the correct number of copies but does not represent an ℓ-up-1-down cycle because copies of "2" occur on two different occasions in the cycle.

Theorem 4.5. *For any finite $\mathcal{P}$, memory length ℓ, and reference-monotone gain function g, there exists an ℓ-up-1-down price cycle that maximizes* (5).

Theorem 4.5 shows that the optimization problem can be restricted to ℓ-up-1-down price cycles, and we will use this result in Subsection 4.4 to construct polynomial-time algorithms. We first provide some intuition for Theorem 4.5 by showing that without the reference-monotonicity assumption, the unique optimal price cycle can be quite complex.

$g(r,p)$	$p=1$	$p=2$	$p=3$	$p=4$
$r=1$	0	1	0	1
$r=2$	0	0	1	1
$r=3$	1	0	0	1
$r=4$	0	0	0	0

Table 3. The gain function g used for Example 4.6. A reference-monotone g would be weakly increasing down each column; this gain function is not.

Example 4.6 (Necessity of Reference-monotonicity). Let $\ell = 2, \mathcal{P} = \{1,2,3,4\}$, and $g(r,p)$ be defined as in Table 3. The optimal price cycle for this example is 414243, whose objective (5) equals

$$\frac{1}{6}\big(g(3,4) + g(3,1) + g(1,4) + g(1,2) + g(2,4) + g(2,3)\big) = 1.$$

It is easy to check that no other price cycle (ℓ-up-1-down or not) can have an average gain of 1.

4.3 Proof of Theorem 4.5

Our goal is to prove that for any given c and price cycle $\pi_0, \ldots, \pi_{c-1}$, its objective value (5) can be upper-bounded by that of an ℓ-up-1-down price cycle. Although our proof is non-constructive, the result of Theorem 4.5 itself can be used formulate a reduced MDP that allows for polynomial-time algorithms, as we show in Subsection 4.4.

Our proof proceeds in two steps. Lemma 4.8 first reduces the given price cycle to an intermediate form, and the proof then repeatedly applies Lemma 4.9 until we end up with an ℓ-up-1-down price cycle, whose objective value has not decreased. To aid the reader, we provide the following example illustrating the two steps of the reduction, that can be referenced while reading the proofs.

Example 4.7. Let $\mathcal{P} = \{1,2,3,4,5,6\}$, $\ell = 3$, and suppose we are given the price cycle below. Its *low points* (as defined in the proof of Lemma 4.8) are underlined.

$$\underline{1}2345\underline{3}4\underline{2}653453$$

The substrings between the low points are replaced in the following way:

- Substring 2345 between the first two low points is replaced with either 222555, 333, or 444;
- Substring 4 is replaced with the empty string, or it is concluded that the objective value of the price cycle is worse than that of constantly offering price 4;
- Substring 65345 is replaced with either 666444, 555555, or 333;
- The empty substring between the final $\underline{3}$ that wraps around to the initial $\underline{1}$ is unchanged.

Lemma 4.8 shows that at least one combination of these replacements would not decrease the objective value. Suppose for illustration this replacement is

$$\mathbf{\underline{1}}222555\mathbf{\underline{32}}666\mathbf{444\underline{3}}$$

where the original low points are still underlined and now we have also bolded the *reset points* (as defined in Lemma 4.9).

The second part of the proof finds two distinct reset points with the same number, and breaks the cycle into two starting from these points, wrapping around as necessary. For example, taking the 3's as the reset points, we can consider the two shorter cycles below:

$$\mathbf{\underline{32}}666\mathbf{444}; \qquad \mathbf{\underline{31}}222\mathbf{555}.$$

By Lemma 4.9, the better of these cycles has objective value no less than that of the original cycle. We take the better cycle and repeat this process until there are no distinct reset points with the

same number. For example, if the better cycle was 32666444, then we might end up with the cycle 326664, or just 4 (both of which are ℓ-up-1-down price cycles). On the other hand, if the better cycle was 31222555, then there is no further reduction. In either case, the process must terminate with an ℓ-up-1-down price cycle. □

We now proceed with the formal proof.

LEMMA 4.8. *For any price cycle $\pi_0, \ldots, \pi_{c-1}$, its objective value* (5) *is upper-bounded by that of a price cycle represented by a string of the form $\rho_0^{k_0} \cdots \rho_{d-1}^{k_{d-1}}$, where d is a positive integer, and for all $t = 0, \ldots, d-1$ we have $\rho_t \in \mathcal{P}$, $k_t \in \{1, \ell\}$, and $k_t = 1$ implying $\rho_t \leq \rho_{t-1 \mod d}$.*

PROOF OF LEMMA 4.8. In the initial price cycle $\pi_0, \ldots, \pi_{c-1}$, define a time $t \in \{0, \ldots, c-1\}$ to be a *low point* if $\pi_t \leq \min\{\pi_{t-\ell \mod c}, \ldots, \pi_{t-1 \mod c}\}$. That is, t is a low point if it is at most the reference price at time t. Let $t_1, \ldots, t_m$ denote the low points in the initial cycle, with $0 \leq t_1 < \cdots < t_m \leq c-1$, noting that there must be at least one low point.

We consider the substrings of prices between low points:

$$\pi_{t_m+1 \mod c} \cdots \pi_{t_1-1 \mod c}; \quad \pi_{t_1+1 \mod c} \cdots \pi_{t_2-1 \mod c}; \quad \cdots \quad ; \quad \pi_{t_{m-1}+1 \mod c} \cdots \pi_{t_m-1 \mod c}.$$

Note that there are m such substrings, some of which may be empty. We iteratively replace these substrings with new substrings in which every token appears ℓ consecutive times, showing that the objective value (5) does not go down, and that all low points are preserved. This would complete the proof because after the m replacements, all tokens would either appear ℓ consecutive times, or be a low point which implies that it is no greater than the immediately preceding price.

To ease notation, at each iteration we re-index (rotate) the cycle so that the current substring under consideration is $\pi_0 \cdots \pi_{t'-1}$, preceded by a low point at time $c-1$ and succeeded by a low point at time t'. None of $0, \ldots, t'-1$ being low points guarantees that $\pi_t > \pi_{c-1}$ for all $t = 0, \ldots, t'-1$, because otherwise the smallest $t \in \{0, \ldots, t'-1\}$ for which $\pi_t \leq \pi_{c-1}$ would be a low point.

Case 1: $t' < \ell$. Because $t' < \ell$, the reference price for each time $t = 0, \ldots, t'-1$ is at most (in fact equal to) π_{c-1}. Therefore, we can express the objective value of the current price cycle as

$$\begin{aligned}
&\frac{1}{c}\left(\sum_{t=0}^{t'-1} g(\pi_{c-1}, \pi_t) + \sum_{t=t'}^{c-1} g(\min\{\pi_{t-\ell \mod c}, \ldots, \pi_{t-1 \mod c}\}, \pi_t)\right) \\
&\leq \frac{\sum_{t=0}^{t'-1} g(\pi_t, \pi_t) + \sum_{t=t'}^{c-1} g(\min\{\pi_{t-\ell \mod c}, \ldots, \pi_{t-1 \mod c}\}, \pi_t)}{t' + (c - t')} \\
&\leq \max\left\{g(\pi_0, \pi_0), \ldots, g(\pi_{t'-1}, \pi_{t'-1}), \frac{\sum_{t=t'}^{c-1} g(\min\{\pi_{t-\ell \mod c}, \ldots, \pi_{t-1 \mod c}\}, \pi_t)}{c - t'}\right\}
\end{aligned}$$

where the first inequality follows from reference-monotonicity because $\pi_t > \pi_{c-1}$ for all $t = 0, \ldots, t'-1$, and the second inequality follows elementarily[1]. If the max in the final expression is attained at any of the first t' arguments, then Lemma 4.8 would be immediately proven, because we have upper-bounded the objective value by a price cycle that is the constant price π_t for some $t \in \{0, \ldots, t'-1\}$.

Otherwise, if the max in the final expression is attained at the final argument, then we claim this is equal to the objective value of the price cycle $\pi_{t'}\pi_{t'+1} \cdots \pi_{c-1}$ (formed by replacing $\pi_0 \cdots \pi_{t'-1}$ with the empty substring). To see this, note that the reference price at any time $t \geq t' + \ell$ is unchanged. For time steps $t = t', \ldots, t' + \ell - 1$, the reference price is still $\pi_{t'}$ because $\pi_{t'} \leq \pi_{c-1}$ by virtue of $t' < \ell$ and t' being a low point, and $\pi_{c-1} \leq \min\{\pi_{c-1-\ell \mod c}, \ldots, \pi_{c-2 \mod c}\}$ by virtue of

[1]For real numbers $a_1, \ldots, a_n$ and $b_1, \ldots, b_n > 0$, it holds that $(a_1 + \cdots + a_n)/(b_1 + \cdots + b_n) \leq \max\{\frac{a_1}{b_1}, \ldots, \frac{a_n}{b_n}\}$.

c being a low point (which can be checked to hold under the new cycle even if $\pi_{c-1-\ell \mod c}$ wraps around). That is, both t' and $c-1$ are still low points in the new price cycle. This completes the proof of Case 1.

Case 2: $t' \geq \ell$. We consider replacing $\pi_0 \cdots \pi_{t'-1}$ with a substring of the form

$$\pi_j^\ell \pi_{\ell+j}^\ell \pi_{2\ell+j}^\ell \cdots \pi_{\lfloor (t'-1-j)/\ell \rfloor \ell + j}^\ell, \qquad j \in \{0, \ldots, \ell-1\}. \tag{6}$$

For each substring $j = 0, \ldots, \ell-1$, note that it has length $(1 + \lfloor (t'-1-j)/\ell \rfloor)\ell$. The reference price for the ℓ copies of its first price π_j is $\pi_{c-1} = \min\{\pi_{c-1}, \pi_j\}$, because $c-1$ is a low point and $\pi_j > \pi_{c-1}$. Meanwhile, for all $j' = 1, \ldots, \lfloor (t'-1-j)/\ell \rfloor$, the reference price for the ℓ copies of its price $\pi_{j'\ell+j}$ is at least $\min\{\pi_{(j'-1)\ell+j}, \pi_{j'\ell+j}\}$. Finally, for any replacement substring j the reference prices at all times $t \geq t'$ remain unchanged, because $\pi_{t'} \leq \pi_{\lfloor (t'-1-j)/\ell \rfloor \ell + j}$ by virtue of t' being a low point and $\pi_{\lfloor (t'-1-j)/\ell \rfloor \ell + j}$ existing (where $\lfloor (t'-1-j)/\ell \rfloor \geq 0$ due to $t' \geq \ell$). Therefore, the objective value under each replacement substring $j = 0, \ldots, \ell-1$ is at least

$$\frac{\ell g(\pi_{c-1}, \pi_j) + \ell \sum_{j'=1}^{\lfloor (t'-1-j)/\ell \rfloor} g(\min\{\pi_{(j'-1)\ell+j}, \pi_{j'\ell+j}\}, \pi_{j'\ell+j}) + \sum_{t=t'}^{c-1} g(\min\{\pi_{t-\ell}, \ldots, \pi_{t-1}\})}{(1 + \lfloor (t'-1-j)/\ell \rfloor)\ell + c - t'}.$$

Now, the objective value of the current price cycle with substring $\pi_0 \cdots \pi_{t'-1}$ can be bounded:

$$\begin{aligned}
&\frac{1}{c}\left(\sum_{t=0}^{\ell-1} g(\pi_{c-1}, \pi_t) + \sum_{t=\ell}^{t'-1} g(\min\{\pi_{t-\ell}, \ldots, \pi_{t-1}\}, \pi_t) + \sum_{t=t'}^{c-1} g(\min\{\pi_{t-\ell}, \ldots, \pi_{t-1}\}, \pi_t)\right) \\
&\leq \frac{1}{c}\left(\sum_{t=0}^{\ell-1} g(\pi_{c-1}, \pi_t) + \sum_{t=\ell}^{t'-1} g(\min\{\pi_{t-\ell}, \pi_t\}, \pi_t) + \sum_{t=t'}^{c-1} g(\min\{\pi_{t-\ell}, \ldots, \pi_{t-1}\}, \pi_t)\right) \\
&= \frac{1}{c}\left(\sum_{j=0}^{\ell-1}\left(g(\pi_{c-1}, \pi_j) + \sum_{j'=1}^{\lfloor (t'-1-j)/\ell \rfloor} g(\min\{\pi_{(j'-1)\ell+j}, \pi_{j'\ell+j}\}, \pi_{j'\ell+j})\right) + \sum_{t=t'}^{c-1} g(\min\{\pi_{t-\ell}, \ldots, \pi_{t-1}\}, \pi_t)\right) \\
&= \frac{1}{c}\left(\frac{1}{\ell}\sum_{j=0}^{\ell-1}\left(\ell g(\pi_{c-1}, \pi_j) + \ell \sum_{j'=1}^{\lfloor (t'-1-j)/\ell \rfloor} g(\min\{\pi_{(j'-1)\ell+j}, \pi_{j'\ell+j}\}, \pi_{j'\ell+j}) + \sum_{t=t'}^{c-1} g(\min\{\pi_{t-\ell}, \ldots, \pi_{t-1}\}, \pi_t)\right)\right) \\
&= \frac{\sum_{j=0}^{\ell-1}\left(\ell g(\pi_{c-1}, \pi_j) + \ell \sum_{j'=1}^{\lfloor (t'-1-j)/\ell \rfloor} g(\min\{\pi_{(j'-1)\ell+j}, \pi_{j'\ell+j}\}, \pi_{j'\ell+j}) + \sum_{t=t'}^{c-1} g(\min\{\pi_{t-\ell}, \ldots, \pi_{t-1}\}, \pi_t)\right)}{\ell(c-t') + \ell \sum_{j=0}^{\ell-1}(1 + \lfloor (t'-1-j)/\ell \rfloor)} \\
&\leq \max_{j=0,\ldots,\ell-1} \frac{\ell g(\pi_{c-1}, \pi_j) + \ell \sum_{j'=1}^{\lfloor (t'-1-j)/\ell \rfloor} g(\min\{\pi_{(j'-1)\ell+j}, \pi_{j'\ell+j}\}, \pi_{j'\ell+j}) + \sum_{t=t'}^{c-1} g(\min\{\pi_{t-\ell}, \ldots, \pi_{t-1}\}, \pi_t)}{1 + (\lfloor (t'-1-j)/\ell \rfloor)\ell + c - t'}.
\end{aligned}$$

The first inequality uses the fact that $\pi_t > \min\{\pi_{t-\ell}, \ldots, \pi_{t-1}\}$ because t is not a low point, and hence $g(\min\{\pi_{t-\ell}, \ldots, \pi_{t-1}\}, \pi_t) = g(\min\{\pi_{t-\ell}, \ldots, \pi_t\}, \pi_t) \leq g(\min\{\pi_{t-\ell}, \pi_t\}, \pi_t)$ (with the inequality applying reference-monotonicity). The first equality partitions the integers $0, \ldots, t'-1$ based on their remainder j when divided by ℓ, noting that $\lfloor (t'-1-j)/\ell \rfloor \ell + j$ is the largest integer less than t' with a remainder of j when divided by ℓ. The third equality holds because $\sum_{j=0}^{\ell-1}(1 + \lfloor (t'-1-j)/\ell \rfloor) = t'$, which is easiest to see from the fact that there are $1 + \lfloor (t'-1-j)/\ell \rfloor$ integers in $\{0, \ldots, t'-1\}$ with remainder j when divided by ℓ. The final inequality follows from Footnote 1.

Therefore, we can always replace $\pi_0 \cdots \pi_{t'-1}$ with one of the substrings from (6) without decreasing the objective value. Moreover, t' remains a low point because it is now preceded by ℓ

copies of $\lfloor (t' - 1 - j)/\ell \rfloor \ell + j$, which appeared in $\{\pi_{t'-\ell}, \ldots, \pi_{t'-1}\}$ and hence is at least $\pi_{t'}$. This completes the proof of Case 2 and the overall proof of Lemma 4.8. □

Lemma 4.9. *In a price cycle $\pi_0, \ldots, \pi_{c-1}$, define a time $t \in \{0, \ldots, c-1\}$ to be a reset point if $\pi_t = \min\{\pi_{t-\ell+1 \mod c}, \ldots, \pi_{t \mod c}\}$. Suppose the price cycle $\pi_0, \ldots, \pi_{c-1}$ contains two distinct reset points with the same price, relabeled to be $t'-1$ and $c-1$ (i.e., $t'-1, c-1$ are both reset points with $t' < c$ and $\pi_{t'-1} = \pi_{c-1}$). Then the objective value* (5) *of the price cycle must be upper-bounded by that of either the price cycle $\pi_0, \ldots, \pi_{t'-1}$ or the price cycle $\pi_{t'}, \ldots, \pi_{c-1}$.*

Proof of Lemma 4.9. The objective value of the original price cycle $\pi_0, \ldots, \pi_{c-1}$ equals

$$\frac{1}{t' + (c - t')}\left(\sum_{t=0}^{\min\{t'-1,\ell-1\}} g(\min\{\pi_{c-1}, \pi_0, \ldots, \pi_{t-1}\}, \pi_t) + \sum_{t=\min\{t'-1,\ell-1\}+1}^{t'-1} g(\min\{\pi_{t-\ell}, \ldots, \pi_{t-1}\}, \pi_t) \right.$$
$$\left. + \sum_{t=t'}^{\min\{c-1,\ell-1\}} g(\min\{\pi_{t'-1}, \pi_0, \ldots, \pi_{t-1}\}, \pi_t) + \sum_{t=\min\{c-1,\ell-1\}+1}^{c-1} g(\min\{\pi_{t-\ell}, \ldots, \pi_{t-1}\}, \pi_t) \right)$$

where we have used the fact that both $t'-1$ and $c-1$ are reset points. Using the same fact, the objective value of price cycle $\pi_0, \ldots, \pi_{t'-1}$ equals

$$\frac{1}{t'}\left(\sum_{t=0}^{\min\{t'-1,\ell-1\}} g(\min\{\pi_{t'-1}, \pi_0, \ldots, \pi_{t-1}\}, \pi_t) + \sum_{t=\min\{t'-1,\ell-1\}+1}^{t'-1} g(\min\{\pi_{t-\ell}, \ldots, \pi_{t-1}\}, \pi_t) \right)$$

while the objective value of price cycle $\pi_{t'}, \ldots, \pi_{c-1}$ equals

$$\frac{1}{c - t'}\left(\sum_{t=t'}^{\min\{c-1,\ell-1\}} g(\min\{\pi_{c-1}, \pi_0, \ldots, \pi_{t-1}\}, \pi_t) + \sum_{t=\min\{c-1,\ell-1\}+1}^{c-1} g(\min\{\pi_{t-\ell}, \ldots, \pi_{t-1}\}, \pi_t) \right).$$

Because $\pi_{t'-1} = \pi_{c-1}$, it follows from Footnote 1 that the objective value of the original price cycle is upper-bounded by the maximum of the latter two expressions. □

Completing the proof of Theorem 4.5. Consider the upper-bounding price cycle that is the result of the reduction in Lemma 4.8. For $t = 0, \ldots, d-1$, first suppose $k_t = 1$. Then it is guaranteed that $\rho_t \le \rho_{t-1 \mod d}$, which means that ρ_t corresponds to a reset point. Indeed, this is immediate if $k_{t-1 \mod d} = \ell$, and if $k_{t-1 \mod d} = 1$ then $\rho_{t-1 \mod d} \le \rho_{t-2 \mod d}$ so we can iteratively apply the same argument. On the other hand, now suppose $k_t = \ell$. In this case, if $\rho_t \le \rho_{t-1 \mod d}$, then the ℓ copies of ρ_t all correspond to reset points, by the same argument as before; if otherwise $\rho_t > \rho_{t-1 \mod d}$, then only the final copy of ρ_t corresponds to a reset point.

All in all, we have proven that every $t = 0, \ldots, d-1$ corresponds to at least 1 reset point, and corresponds to ℓ reset points if $k_t = \ell$ and $\rho_t \le \rho_{t-1 \mod d}$. We now iteratively apply Lemma 4.9 to reduce the price cycle whenever distinct reset points have the same price, noting that after each reduction the upper-bounding price cycle still takes the form described in Lemma 4.8 and the same properties still hold. Let $\rho_0^{k_0} \cdots \rho_{d-1}^{k_{d-1}}$ represent the final price cycle which no longer has distinct reset points with the same price. It must be the case that $\rho_0, \ldots, \rho_{d-1}$ are distinct prices in $\mathcal{P}$, and moreover, if $k_t = \ell$ then we must have $\rho_t > \rho_{t-1 \mod d}$, because otherwise all ℓ copies of ρ_t would be reset points. (In the case where $\ell = 1$, this argument is irrelevant.) The original property from Lemma 4.8 that if $k_t = 1$ then $\rho_t \le \rho_{t-1 \mod d}$ also still holds; in fact we would have $\rho_t < \rho_{t-1 \mod d}$, by distinctness. This shows that the the final price cycle must satisfy precisely the definition of ℓ-up-1-down, completing the proof of Theorem 4.5. □

4.4 Computational Consequences; Tightness of ℓ-up-1-down Characterization

Theorem 4.5 shows that to maximize long-run average gain (5) over all positive integers $c \le |\mathcal{P}|^\ell$ and price cycles $(\pi_0, \dots, \pi_{c-1}) \in \mathcal{P}^c$, it suffices to search over ℓ-up-1-down price cycles, which are defined by a generator cycle of at most d distinct prices in $\mathcal{P}$ (see Definition 4.3). We now show that the latter problem can be formulated using a reduced MDP.

For any generator cycle $\rho_0, \dots, \rho_{d-1}$ of distinct prices in $\mathcal{P}$, its long-run average gain (5) equals

$$\frac{\sum_{t=0}^{d-1} g(\rho_{t-1 \mod d}, \rho_t)(1 + (\ell-1)\mathbb{1}(\rho_t > \rho_{t-1 \mod d}))}{\sum_{t=0}^{d-1}(1 + (\ell-1)\mathbb{1}(\rho_t > \rho_{t-1 \mod d}))}, \tag{7}$$

because the reference price is always $\rho_{t-1 \mod d}$ while offering ρ_t, for all $t = 0, \dots, d-1$, due to the ℓ-up-1-down structure that the previous price $\rho_{t-1 \mod d}$ is repeated ℓ times if it is higher than the price before it.

We now construct an MDP where both the state and action space is $\mathcal{P}$. When any action $p \in \mathcal{P}$ is taken from any state $r \in \mathcal{P}$, the next state is always p; the reward is $g(r,p)$ if $r \ge p$, and $\ell g(r,p)$ if $r < p$ but the *transition takes ℓ steps* (instead of 1 step, so the reward-per-step is still $g(r,p)$). The objective is to maximize long-run average reward per step. We again restrict without loss to stationary deterministic policies[2], to see that the optimal policy is defined by a cycle $\rho_0, \dots, \rho_{d-1}$ of distinct states in $\mathcal{P}$. Moreover, the long-run average reward per step of this policy equals exactly (7).

Therefore, the optimization problem over ℓ-up-1-down price cycles is captured by this reduced MDP, which can be solved via the following system of infinite-horizon undiscounted Bellman's equations:

$$h(r) = \max_{p \in \mathcal{P}} \Big(\big(g(r,p) - \mathsf{OPT}\big)\big(1 + (\ell-1)\mathbb{1}(r < p)\big) + h(p) \Big) \qquad \forall r \in \mathcal{P}. \tag{8}$$

Here, variable OPT denotes the optimal long-run average reward, and $h(p)$ denotes the bias for each state p, one of which can be normalized to 0 so that there are both $|\mathcal{P}|$ variables and equations. The multiplication by the factor of $1 + (\ell-1)\mathbb{1}(r < p)$ accounts for the transition times. For further details and algorithms that solve this in polynomial-time, we defer to Puterman [2014].

We can use (8) to also prove that our characterization of ℓ-up-1-down policies is tight.

Theorem 4.10 (proven in Appendix C). *Let $\rho_0, \dots, \rho_{d-1}$ be any cycle of distinct prices in $\mathcal{P}$. Under any memory length ℓ, there exists a reference-monotone gain function $g : \mathcal{P}^2 \to \mathbb{R}$ for which the unique optimal price cycle is $\rho_0^{k(\rho_{d-1},\rho_0)} \rho_1^{k(\rho_0,\rho_1)} \cdots \rho_{d-1}^{k(\rho_{d-2},\rho_{d-1})}$, where $k(r,p) := 1 + (\ell-1)\mathbb{1}(r < p)$.*

5 Conclusion and Post-mortem

Personalizing promotions is more possible than ever, yet optimizing them for the long-term remains an unsolved business problem. Our paper combines a successful real-world deployment of personalized promotions with a theoretical model and structural results for optimizing promotion cycles. Unfortunately, a change in management at our industry partner caused the department responsible for personalized promotions to disperse at the end of 2024. The new management favored offering the biggest discount every day, removing all personalization; this prevented us from testing a more sophisticated ℓ-up-1-down policy, and our algorithm was phased out by early 2025. More encouragingly, a growing literature on promotion fatigue and "annoyance" identifies the long-term loss from overly myopic promotion offerings (see Subsection 1.4). We believe that our empirical study and structural price cycling results will prove useful for researchers and practitioners.

[2]Technically this is a Semi-Markov Decision Process due to the inhomogeneous transition times, but the optimality of stationary deterministic policies still holds. We can alternatively have homogeneous transition times if for all transitions from states r to p with $r < p$, we add $\ell - 1$ dummy states in the middle with a single action for proceeding.

Acknowledgments

The authors thank Chamsi Hssaine, Zhenyu Hu, Hanwei Li, and Weiming Zhu for early feedback on a draft.

References

Shipra Agrawal and Wei Tang. 2024. Dynamic Pricing and Learning with Long-term Reference Effects. In *Proceedings of the 25th ACM Conference on Economics and Computation*. 72–72.

Javier Albert and Dmitri Goldenberg. 2022. E-commerce promotions personalization via online multiple-choice knapsack with uplift modeling. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2863–2872.

Jackie Baek, Justin J Boutilier, Vivek F Farias, Jonas Oddur Jonasson, and Erez Yoeli. 2025. Policy optimization for personalized interventions in behavioral health. *Manufacturing & Service Operations Management* 27, 3 (2025), 770–788.

Jackie Baek, Daniel Chen, Will Ma, and Dmitry Mitrofanov. 2026. Balancing Customer Engagement and Annoyance in Online Retail: Insights from a Field Experiment. (2026).

Omar Besbes and Ilan Lobel. 2015. Intertemporal price discrimination: Structure and computation of optimal policies. *Management Science* 61, 1 (2015), 92–110.

Andre P Calmon, Florin D Ciocan, and Gonzalo Romero. 2021. Revenue management with repeated customer interactions. *Management Science* 67, 5 (2021), 2944–2963.

Jiacheng Chang, Xiao Lei, and Feng Tian. 2024. Pricing and Addiction Control for Digital Services. *Available at SSRN 4962550* (2024).

Xin Chen, Peng Hu, and Zhenyu Hu. 2017. Efficient algorithms for the dynamic pricing problem with reference price effect. *Management Science* 63, 12 (2017), 4389–4408.

Maxime C Cohen, Swati Gupta, Jeremy J Kalas, and Georgia Perakis. 2020. An efficient algorithm for dynamic pricing using a graphical representation. *Production and Operations Management* 29, 10 (2020), 2326–2349.

Tamar Cohen-Hillel, Kiran Panchamgam, and Georgia Perakis. 2023. High-low promotion policies for peak-end demand models. *Management Science* 69, 4 (2023), 2016–2050.

Jinglong Dai, Hanwei Li, Weiming Zhu, Jianfeng Lin, and Binqiang Huang. 2024. Data-Driven Real-time Coupon Allocation in the Online Platform. *arXiv preprint arXiv:2406.05987* (2024).

Arnoud V den Boer and N Bora Keskin. 2022. Dynamic pricing with demand learning and reference effects. *Management Science* 68, 10 (2022), 7112–7130.

Gadi Fibich, Arieh Gavious, and Oded Lowengart. 2003. Explicit solutions of optimization models and differential games with nonsmooth (asymmetric) reference-price effects. *Operations Research* 51, 5 (2003), 721–734.

Daniel Freund and Chamsi Hssaine. 2021. Fair incentives for repeated engagement. *arXiv preprint arXiv:2111.00002* (2021).

Eric A Greenleaf. 1995. The impact of reference price effects on the profitability of price promotions. *Marketing science* 14, 1 (1995), 82–104.

Mengzi Amy Guo, Hansheng Jiang, and Zuo-Jun Max Shen. 2025. Multiproduct dynamic pricing with reference effects under logit demand. *Manufacturing & Service Operations Management* 27, 5 (2025), 1645–1663.

Zhenyu Hu, Xin Chen, and Peng Hu. 2016. Dynamic pricing with gain-seeking reference price effects. *Operations Research* 64, 1 (2016), 150–157.

Zhenyu Hu and Javad Nasiry. 2018. Are markets with loss-averse consumers more sensitive to losses? *Management Science* 64, 3 (2018), 1384–1395.

Srikanth Jagabathula, Dmitry Mitrofanov, and Gustavo Vulcano. 2022. Personalized retail promotions through a directed acyclic graph–based representation of customer preferences. *Operations Research* 70, 2 (2022), 641–665.

Bo Jiang, Zizhuo Wang, and Nanxi Zhang. 2022. Revenue Management Under a Price Alert Mechanism. *Available at SSRN 4154861* (2022).

Hansheng Jiang, Junyu Cao, and Zuo-Jun Max Shen. 2024. Intertemporal pricing via nonparametric estimation: Integrating reference effects and consumer heterogeneity. *Manufacturing & Service Operations Management* 26, 1 (2024), 28–46.

Daniel Kahneman, Barbara L Fredrickson, Charles A Schreiber, and Donald A Redelmeier. 1993. When more pain is preferred to less: Adding a better end. *Psychological science* 4, 6 (1993), 401–405.

Xiao Lei, Beichen Wan, and Shixin Wang. 2023. Content rotation in the presence of satiation effects. *Available at SSRN 4593945* (2023).

Xiao Liu. 2023. Dynamic coupon targeting using batch deep reinforcement learning: An application to livestream shopping. *Marketing Science* 42, 4 (2023), 637–658.

Yan Liu and William L Cooper. 2015. Optimal dynamic pricing with patient customers. *Operations research* 63, 6 (2015), 1307–1319.

Ilan Lobel. 2020. Dynamic pricing with heterogeneous patience levels. *Operations Research* 68, 4 (2020), 1038–1046.

Tridib Mazumdar, Sevilimedu P Raj, and Indrajit Sinha. 2005. Reference price research: Review and propositions. *Journal of marketing* 69, 4 (2005), 84–102.

Javad Nasiry and Ioana Popescu. 2011. Dynamic pricing with loss-averse consumers and peak-end anchoring. *Operations research* 59, 6 (2011), 1361–1368.

Ioana Popescu and Yaozhong Wu. 2007. Dynamic pricing strategies with reference effects. *Operations research* 55, 3 (2007), 413–429.

Martin L Puterman. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

David Shmoys and Shujing Wang. 2019. How to solve a linear optimization problem on incentive allocation? (2019). https://eng.lyft.com/how-to-solve-a-linear-optimization-problem-on-incentive-allocation-5a8fb5d04db1 Lyft Engineering blog.

Elizabeth Utley. 2024. Taobao and Tmall Upgrades Consumer Shopping Experience and Merchant Support Through AI. (2024). https://www.alizila.com/taobao-and-tmall-upgrades-consumer-shopping-experience-and-merchant-support-through-ai/ Alizila.

Zizhuo Wang. 2016. Intertemporal price discrimination via reference price effects. *Operations research* 64, 2 (2016), 290–296.

Viola Zhou. 2023. How Temu topped the U.S. app charts by turning shopping into a game. (2023). https://restofworld.org/2023/temu-mobile-gaming/ Rest of World.

A Processed Features for Prediction Model

B Proof of Proposition 2.1

Suppose $0 \le \lambda \le \lambda'$ and $v, v' \in \mathcal{V}$ with $v \le v'$. If for customer i, the bigger discount v' is preferred over v under λ', i.e. $(1-\lambda' v')q(x_{it}, v') \ge (1-\lambda' v)q(x_{it}, v)$, then we have

$$\frac{1-\lambda' v'}{1-\lambda' v} \ge \frac{q(x_{it}, v)}{q(x_{it}, v')}.$$

We know however that $\frac{1-\lambda v'}{1-\lambda v} \ge \frac{1-\lambda' v'}{1-\lambda' v}$ by the rearrangement inequality, which implies that $(1-\lambda v')q(x_{it}, v') \ge (1-\lambda v)q(x_{it}, v)$, i.e. v' is also preferred over v under λ. Therefore, increasing λ_t from λ to λ' cannot cause v_{it} to increase for any customer i.

Under the assumption that $q(x_{it}, v)$ is increasing in v, it is clear that (3) is increasing in v_{it} for all i, and hence decreasing in λ_t.

C Proof of Theorem 4.10

We can without loss assume $\mathcal{P} = \{\rho_0, \dots, \rho_{d-1}\}$, by creating arbitrarily negative values for $g(r, p)$ when $p \notin \{\rho_0, \dots, \rho_{d-1}\}$. For convenience, we can denote the price set to be $\mathcal{P} = \{1, \dots, d\}$.

Take any real values $h(1) < \cdots < h(d)$. Define

$$g(\rho_t) := \frac{h(\rho_{t-1 \mod d}) - h(\rho_t)}{k(\rho_{t-1 \mod d}, \rho_t)} + C \qquad \forall t = 0, \dots, d-1, \tag{9}$$

where $k(\cdot, \cdot)$ is defined as in the statement of Theorem 4.10, and C is a large positive constant that ensures $g(\rho_t) > 0$ for all $t = 0, \dots, d-1$ (it suffices if $\frac{h(1)-h(d)}{\ell} + C > 0$). Finally, define the reference-monotone gain function g to be

$$g(r, \rho_t) = g(\rho_t)\mathbb{1}(r \ge \rho_{t-1 \mod d}) \qquad \forall t = 0, \dots, d-1; r \in \mathcal{P}. \tag{10}$$

We show for this gain function g that the unique optimal price cycle is $\rho_0^{k(\rho_{d-1},\rho_0)} \rho_1^{k(\rho_0,\rho_1)} \cdots \rho_{d-1}^{k(\rho_{d-2},\rho_{d-1})}$, as required for the statement of Theorem 4.10. It suffices to show that if substitute these values of $h(1), \dots, h(d)$ into the optimality condition (8), along with $\mathrm{OPT} = C$ (the long-run average gain of the optimal price cycle), then the "max" is achieved if and only if $r = \rho_{t-1 \mod d}, p = \rho_t$ for some $t = 0, \dots, d-1$. In other words, we need to prove

$$h(\rho_{t-1 \mod d}) \ge (g(\rho_{t-1 \mod d}, \rho_{t'}) - C)k(\rho_{t-1 \mod d}, \rho_{t'}) + h(\rho_{t'}) \quad \forall t, t' \in \{0, \dots, d-1\}$$

Category	Features
Purchasing	Total purchase amount in the last 30/360 days
	Number of transactions in the last 3/7/30/360 days
	Average site sale discount of purchases in the last 30/360 days
	Number of items in the cart
	Total value of products in the cart
	Average site sale discount in the cart
Coupon	Number of free shipping coupons in last 7/28 days
	Number of stackable coupons in last 7/28 days
	Coupon value received 1/2 day ago
	Largest coupon in the last 3/7/28 days
	Average coupon in the last 3/7/28/30 days
	Median coupon in the last 28 days
	Variance of coupon values in the last 7/28 days
	Average coupon clicked in the last 7/30 days
	Average coupon opened in the last 7/30 days
	Average coupon discount of purchases in the last 30/360 days
	Average coupon discount of all historical purchases
	Percentage of historical orders where a coupon was used
Email	Email opened 1/2 day ago
	Number of emails opened in last 3/7/28 days
	Whether all emails were opened in the last 28 days
	Email clicked 1/2 day ago
	Number of emails clicked in last 3/7/28 days
Website	Number of times viewed webpage in the last 1/3/7/30 days
	Number of times viewed cart in the last 1/3/7/30 days
	Number of days viewed cart in the last 1/3/7/30 days
	Number of times viewed a product in the last 7 days
	Number of times viewed checkout in the last 7 days
Time	Day of week

Table 4. List of features used in the prediction model. Time windows written with slashes (e.g., 3/7/28 days) indicate multiple distinct features, one computed for each listed window length.

or equivalently

$$g(\rho_{t-1 \mod d}, \rho_{t'}) \leq \frac{h(\rho_{t-1 \mod d}) - h(\rho_{t'})}{k(\rho_{t-1 \mod d}, \rho_{t'})} + C \qquad \forall t, t' \in \{0, \dots, d-1\} \tag{11}$$

with equality if and only if $t' = t$.

If $t' = t$, then equality holds by the definitions in (9)–(10). Now suppose $t' \neq t$, and first consider the case where $\rho_{t-1 \mod d} < \rho_{t'-1 \mod d}$. We have

$$g(\rho_{t-1 \mod d}, \rho_{t'}) = 0 < \frac{h(1) - h(d)}{\ell} + C \leq \frac{h(\rho_{t-1 \mod d}) - h(\rho_{t'})}{k(\rho_{t-1 \mod d}, \rho_{t'})} + C$$

as desired, where the final (weak) inequality holds because the smallest possible value of $\frac{h(\rho_{t-1 \mod d})-h(\rho_{t'})}{k(\rho_{t-1 \mod d},\rho_{t'})}$ is $\frac{h(1)-h(d)}{\ell}$. In the other case where $\rho_{t-1 \mod d} > \rho_{t'-1 \mod d}$, we have

$$g(\rho_{t-1 \mod d}, \rho_{t'}) = g(\rho_{t'}) = \frac{h(\rho_{t'-1 \mod d}) - h(\rho_{t'})}{k(\rho_{t'-1 \mod d}, \rho_{t'})} + C < \frac{h(\rho_{t-1 \mod d}) - h(\rho_{t'})}{k(\rho_{t-1 \mod d}, \rho_{t'})} + C$$

as desired, where the inequality holds because $\frac{h(p)-h(\rho_{t'})}{k(p,\rho_{t'})}$ is a strictly increasing function over the prices $p \in \mathcal{P}$ and $\rho_{t-1 \mod d} > \rho_{t'-1 \mod d}$. This completes the proof of Theorem 4.10.