

An Efficient Algorithm for Dynamic Pricing Using a Graphical Representation

Maxime C. Cohen* 

Desautels Faculty of Management, McGill University, Montreal, Quebec H3A 1G5, Canada, maxime.cohen@mcgill.ca

Swati Gupta

Georgia Institute of Technology, Industrial and Systems Engineering, Atlanta, Georgia 30332, USA, swatig@gatech.edu

Jeremy J. Kalas, Georgia Perakis 

Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA, kalas.jeremy@gmail.com, georgiap@mit.edu

We study a multi-period, multi-item dynamic pricing problem faced by a retailer. The objective is to maximize the total profit by choosing prices, while satisfying several business rules. The strength of our work lies in our graphical model reformulation, which allows us to use ideas from combinatorial optimization. We do not make any assumptions on the structure of the demand function. The complexity of our method depends linearly on the number of time periods but is exponential in the memory of the model (number of past prices that affect current demand) and in the number of items. We prove that the profit maximization problem is NP-hard by showing an approximation preserving reduction from the weighted Max-3-SAT problem. We next introduce the discrete reference price model which is a discretized version of the reference price model, accounting for an exponentially smoothed contribution of all past prices. We show that our problem can be solved efficiently under this model. We then approximate common demand functions using the discrete reference price model. To handle cross-item effects among multiple items, we propose to use a virtual reference price that assigns a reference price for each category of items (as opposed to a reference price for each item). To enhance the tractability of our approach, we cluster items into blocks and show how to adapt our method to include business constraints across blocks. Finally, we apply our solution approach using demand models calibrated with supermarket data and validate its practical performance.

Key words: retail pricing; layered graph; reference price model; multi-item pricing

History: Received: May 2019; Accepted: May 2020 by Kalyan Singhal after 1 revision.

1. Introduction

Pricing decisions play an important role in determining the profit of any commodity-based industry. In supermarkets, dynamic pricing translates to promoting the right product(s) at the right time using the appropriate promotion depth. Price promotions can increase product visibility, store traffic, and sometimes even induce customers to switch brands. A study by A.C. Nielsen in 2004 estimated that 12%–25% of supermarket sales in five European countries were made during promotion (Gedenk et al. 2006).

Price variations do not only help retailers meet their sales targets but also substantially contribute to the total profit. One of the supermarket industry characteristics lies in the small profit margins earned by retailers for most items. A report published by the Community Development Financial Institutions (CDFI) Fund states that the average profit margin in the supermarket industry was 1.9% in 2010, and Yahoo! Finance data concluded that the average net

profit margin for publicly traded US-based grocery stores in 2012 was also 1.9%. These studies provide evidence that setting the right prices can be a good way for retailers to increase their sales and profits. However, many retailers are still currently using a manual process for planning prices. The natural question is: Can we develop efficient optimization models that cater to various business requirements and can be used to solve realistic instances?

In this study, we study the profit maximization problem faced by a retailer who needs to decide the prices of several items over the selling season (e.g., 13 weeks). In practice, retailers need to satisfy several business rules such as respecting a limited number of price changes (a comprehensive discussion is presented in section 2.1). Our approach can handle general non-linear demand models that capture behavioral effects observed in practice (and supported by actual data). For example, our demand models account for seasonality effects, the post-promotion dip effect (induced by consumer stockpiling),

and cross-item effects (substitution and complementarity among different items). We consider demand models that are expressed as non-linear and time-dependent functions of the current and past prices and seek to solve the profit maximization problem faced by the retailer. We develop an efficient method to solve this problem, allowing the retailer to test several “what-if” scenarios to better infer the impact of different pricing policies. We introduce a graphical representation that allows us to cast the dynamic pricing problem as solving a maximum weighted path on a layered graph. We then use this representation to derive complexity results and to develop an efficient method for solving our problem.

1.1. Contributions

Maximizing profits using dynamic pricing is an important problem that has captured the attention of retailers and researchers. Our contributions can be summarized as follows.

- *Formulating the pricing problem as a layered graph and deriving an NP-hardness result.*

We present a dynamic programming formulation of the profit maximization problem as a maximum weighted path on a layered graph. This representation holds for any non-linear and time-dependent demand function that depends on current and past prices. One can also easily incorporate pricing business rules by adapting the graph structure. We provide an NP-hardness result by reducing the weighted maximum 3-satisfiability problem to the profit maximization problem. This reduction is approximation preserving, thus showing that there cannot exist a polynomial-time approximation scheme (for any factor better than $7/8$) for our problem. We then derive complexity results implying that our method scales linearly with the number of time periods but is exponential in the model memory (number of past prices that affect current demand) and the number of items.

- *Introducing and studying the discrete reference price model.*

We introduce the discrete reference price model, which is a discretized version of the commonly used reference price model. First, we show that this model yields a good approximation of the (continuous) reference price model both in terms of demand and profit. Second, we develop an efficient algorithm (quasi-polynomial-time approximation scheme) for the profit maximization problem. Third, we propose a procedure to approximate several common demand functions using the discrete reference price model—allowing us to efficiently solve instances with large memory parameters.

- *Extending our approach for multiple items and incorporating global constraints.*

We extend our results to the setting with multiple items. Inspired by the discrete reference price model, we propose two solution approaches: consumers form a reference price for each product separately, or a joint *virtual reference price* for the entire category of items. To our knowledge, this study is the first to consider the concept of a virtual reference price for multiple items. To increase the tractability of our approach, we introduce the notion of blocks and organize items into smaller clusters so that cross-item interactions across blocks are negligible. However, it now becomes challenging to impose global pricing constraints across blocks. Using ideas from combinatorial optimization, we limit the total number of promotions across blocks by solving a multi-choice knapsack problem. We also develop ways to handle price-ordering and exclusivity constraints.

- *Testing our methods on realistic-size instances using supermarket data.*

Using data from Oracle Retail, we evaluate the method proposed in this study using supermarket coffee data. We convey that our solution approach can solve realistic-size instances in a few minutes.

1.2. Literature Review

Dynamic pricing and sales promotions are extensively studied in the literature (see, e.g., Blattberg and Neslin 1990, Özer et al. 2012, and the references therein). A recent related work on planning price promotions can be found in Cohen et al. (2017), where the authors provide an optimization formulation and propose an efficient approximation method for solving the problem based on linearizing the objective and solving a linear program. Our study has a similar motivation but bears several key differences. First, we model the problem as a directed layered graph and use dynamic programming instead of a linear programming approximation. This graphical representation allows us to easily capture pricing business rules, while providing an access to combinatorial techniques to solve the problem. Second, we show a reduction from maximum satisfiability to our problem, allowing us to formally show that the problem is NP-hard. Third, the algorithms developed in this study yield exact solutions and we extend our approach to multiple items by proposing the concept of a virtual reference price. Finally, the problem considered in this study is not restricted to promotions as we study a general multi-period pricing problem. To our knowledge, there are no existing approaches that can solve our problem optimally (under a non-linear demand model, discrete prices, and the presence of business rules).

Sales promotions are also well-studied in marketing (see Blattberg and Neslin (1990) and the references therein). Several retailers (e.g., Walmart) employ an everyday-low-price strategy (see, e.g., Lal and Rao 1997), whereas many others use temporary price reductions (i.e., promotions) on selected items (examples of such works include Blattberg et al. 1995, Nijs et al. 2001). For a comparison between the two pricing strategies, see Ellickson and Misra (2008). In marketing, the focus is often on estimating demand models (e.g., linear regression or choice models) to draw managerial insights on the impact of promotions. For example, Foekens et al. (1998) study econometrics models based on scanner data to examine the dynamic effects of promotions. It has been observed that promotions may lead to a decrease in future sales, a phenomenon referred to as the *post-promotion dip* effect. One way to capture the post-promotion dip effect is to model demand as a function of the price in the current period and the prices in the most recent periods (Heerde et al. 2000, Macé and Neslin 2004).

Our work is related to the field of dynamic pricing (see, e.g., Talluri and van Ryzin 2005). A common approach used in the dynamic pricing literature is to model consumers using a reference price model (Fibich et al. 2003, Kopalle et al. 1996, Popescu and Wu 2007). The reference price model assumes that past prices affect the consumers' willingness to pay. Then, consumers compare the current price to the reference price as a benchmark. Prices above the reference reduce demand, whereas prices below the reference lead to a demand increase. Kopalle et al. (1996), Fibich et al. (2003), and Popescu and Wu (2007) study an infinite-horizon dynamic pricing problem with a reference price model. Our paper differs from the models in the dynamic pricing literature in that our problem is directly inspired by practical models tailored to setting promotions for supermarkets and includes important business rules. In addition, we extend the model of a reference price to the context of multiple items, where a virtual reference price captures the cross-item effects on demand. This extension is important as it allows to capture substitution and complementarity effects, while deciding the promotions for several items simultaneously. Chen et al. (2016) consider the asymmetric reference price model for a single item and presents an exact algorithm to solve the continuous pricing problem under some technical conditions. For problems where these conditions do not hold, the authors develop an approximation algorithm using dynamic programming. However, the approach in Chen et al. (2016) cannot easily handle price-dependent business constraints and does not extend to multiple items. On the other hand, our paper can handle pricing business rules, provides

NP-hardness and complexity results, and extends to multiple items.

In the literature on item pricing, one can find two streams of studies. On the one hand, papers such as Hartline and Koltun (2005) and Balcan et al. (2008) rely on the modeling assumption that customers have a valuation for the items (typically modeled as a probability distribution function). This assumption leads to a pricing problem that is hard to solve. In fact, even optimizing over two prices is not an easy problem (it was mentioned as an open question in Balcan et al. 2008). Other related papers in this context are Chakraborty et al. (2013) and den Boer (2015). On the other hand, our paper assumes that demand functions are estimated at an aggregate level, using historical purchase data. In other words, we estimate a price-demand function for the different items, instead of customer valuation distributions. Using supermarket retail data, we find that demand functions with a memory parameter (i.e., the current demand depends explicitly on past prices) yield a high out-of-sample prediction accuracy. To our knowledge, our paper is the first to show the NP-hardness of the price optimization problem, under aggregate demand functions and prices constrained to lie in a discrete set.

Finally, our work is related to the field of retail operations and more specifically, to pricing problems. One of the constraints considered in this study imposes the prices to lie in a discrete set. Zhao and Zheng (2000) consider a dynamic pricing problem for a fixed-inventory perishable product sold over a finite (continuous) time horizon. For the special case of a discrete price set, the authors solve the continuous time dynamic program by applying a discretization approach and using backward recursion. Our approach is different in nature as it considers a general demand function form that depends on current and past prices. Subramanian and Sherali (2010) study a pricing problem for retailers, where prices are subject to inter-item constraints. Due to the nonlinearity of the objective, they propose a linearization technique to solve the problem. In our study, we also consider a model for multiple items that includes several global constraints.

On the technical side, we use concepts related to graph theory and combinatorial algorithms. We assume the reader to be familiar with the theory of computational complexity and hardness of problems. Apart from pointers to relevant references pertaining to specific combinatorial algorithms and complexity results in the paper, Schrijver (2003) and Vazirani (2013) are great references to review these concepts.

Structure of the paper. In section 2, we introduce our model and assumptions. In section 3, we focus on the single item setting and present our graphical

representation, the NP-hardness, and the complexity results. We then introduce the discrete reference price model in section 4. Section 5 extends our results to the setting with multiple items. Computational experiments using supermarket data and our conclusions are presented in sections 6 and 7, respectively.

2. Model and Assumptions

Given a set of n items and a finite planning horizon of T time periods, the profit maximization problem aims to set the price of each item at each period to maximize the total profit. The prices are assumed to come from a discrete set (e.g., must end with 9 cents). The demand is assumed to be a time-dependent function that depends on the current price and on a constant number m (referred to as the *memory* parameter) of past prices, see Equation (1). We first consider the setting with a single item and discuss the extension with multiple items in section 5.

We denote the unit cost of the item at time $t \in \{1, \dots, T\}$ by c_t and the discrete set of admissible prices, called the *price ladder*, by $\mathcal{Q}_p = \{q^0 > q^1 > \dots > q^k > \dots > q^Q\}$. The regular price (i.e., the maximum price) is denoted by q^0 and the minimum price by q^Q .

It is well known that when the price is reduced, consumers tend to purchase larger quantities. Nevertheless, this can also induce a post-promotion dip effect (see, e.g., Macé and Neslin 2004) due to the stockpiling behavior of consumers. In other words, for some items, customers will purchase larger quantities toward future consumption (e.g., toiletries and non-perishable goods). Due to the consumer stockpiling behavior, a price reduction increases the demand at the current period but also reduces the demand in subsequent periods, with the demand slowly recovering to the nominal level. We propose to capture this effect by a demand model that explicitly depends on the current price p_t and on the m past prices $p_{t-1}, p_{t-2}, \dots, p_{t-m}$. We denote the vector $\mathbf{p}_t = (p_t, p_{t-1}, \dots, p_{t-m})$. In addition, our models have the flexibility to assign different weights to reflect how strongly a past price affects the current demand. The parameter $m \in \mathbb{N}_0$ represents the memory of consumers with respect to past prices and varies depending on several features of the item. In practice, m is estimated from data. We consider a general time-dependent demand function denoted by $d_t(\mathbf{p}_t)$ that explicitly depends on the current price and m past prices. Mathematically, the demand at time t is given by:

$$d_t(\mathbf{p}_t) = h_t(p_t, p_{t-1}, \dots, p_{t-m}). \quad (1)$$

We consider solving a finite horizon profit maximization problem given by:

$$\max_{p_t} \sum_{t=1}^T (p_t - c_t) d_t(\mathbf{p}_t), \quad (2)$$

s.t Several business rules.

We next describe the business rules we incorporate in our formulation.

2.1. Business Rules

1. *Prices are chosen from a discrete price ladder.* For each product, there is a finite set of permissible prices. For example, prices may have to end with 9 cents. The price ladder for an item can be time-dependent, but for simplicity we assume that the elements of the price ladder are time-independent (our results still hold when relaxing this assumption).
2. *Limited number of price changes.* The retailer may want to limit the frequency of price changes for a product.¹ This requirement is motivated from the fact that retailers wish to preserve the image of their store and not train customers to be deal seekers. For example, the number of prices changes for a particular product during the quarter may be required to remain below $L = 3$.
3. *Separating periods between successive price changes.* A common requirement is to space out two successive price changes by a minimal number of separating periods, denoted by S . Indeed, if successive price changes are too close together, this may hurt the store image and incentivize consumers to be deal seekers. This business requirement may be dictated by the brand manufacturer that often restricts the frequency of promotions to preserve its brand image.
4. *Inter-price constraints.* The retailer often wants to impose constraints on the prices at the different periods. For example, prices can only decrease in time as the item is very seasonal (markdown strategy). Alternatively, the first and last prices can be required to be the same.
5. *Inter-item constraints.* The retailer may need to satisfy constraints that link the prices of the different items. A common business rule is to limit the total number of price changes for all items during the season. We provide a more detailed discussion on this type of constraints in section 5.4.

2.2. Assumptions

As mentioned, we first consider the setting with a single item and present the extension with multiple items in section 5. We assume that at each period, the

retailer orders the item from the supplier at a linear ordering cost that can vary over time, c_t . This assumption holds under the conventional wholesale price contract which is frequently used (see, e.g., Porteus 1990).

We also consider the demand to be a deterministic function $h_t(\cdot)$ of current and past prices. This assumption is supported by the fact that we capture the most important factors that affect demand: seasonality as well as current and past prices. Using supermarket data, we found that our estimated demand models have a low forecast error: the out-of-sample R^2 was between 0.85 and 0.96 (some estimation results can be found in Cohen et al. 2017). For the multiple items setting, we consider cross-item effects on the demand function (i.e., the price of item i can affect the demand of item $j \neq i$). We assume that the estimation-optimization process is sequential: we first estimate the demand model from historical data and then compute the optimal prices.

Finally, we assume that the retailer always carries enough inventory to meet demand in each period. This assumption does not apply to all products and retail settings. For example, it is common practice in the fashion industry to intentionally produce limited inventory. Nevertheless, the assumption that the retailer carries enough inventory is reasonable in settings such as supermarket items such as coffee and soft drinks. These products are called fast-moving consumer goods and are available all year round. Since their shelf lives are usually greater than six months, customers have been conditioned to expect that these products would always be in stock at retail stores.

Starting with the single item setting allows us to present our approach in a more concise way. Interestingly, several item categories do not have significant cross-item effects so that effectively, the items can be treated as independent and one can solve the problem for each item separately. We extend our approach to multiple items in section 5 by capturing cross-item effects.

3. Graphical Representation

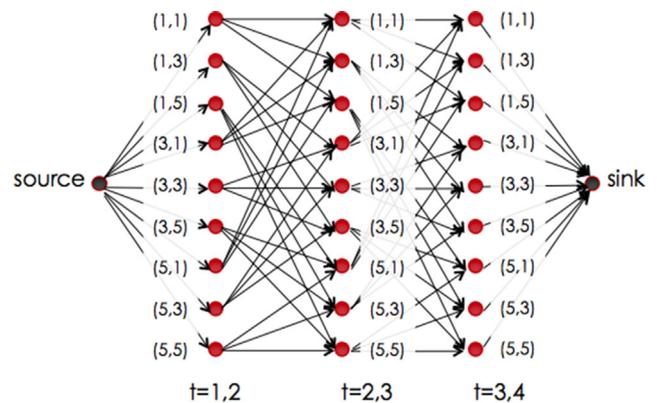
In this section, we present our graphical representation to model the profit maximization problem as a maximum weighted path problem on a layered graph. We then report the complexity results and conclude by the NP-hardness result.

3.1. Constructing the Graph

Recall that we denote our planning horizon by T and the price ladder by $\mathcal{Q}_p = \{q^0, q^1, \dots, q^Q\}$. For each

period $t \in \{m, \dots, T\}$, we construct the nodes (\mathbf{x}, t) where $\mathbf{x} \in \mathcal{Q}_p^m$, that is, all possible m -tuples of prices \mathcal{Q}_p . When the price ladder is time-varying (i.e., $p_t \in \mathcal{Q}_{t,p}$), we construct nodes (\mathbf{x}, t) such that $\mathbf{x} \in \mathcal{Q}_{t,p}^m$. As discussed, for ease of exposition, we consider a static price ladder. In Figure 1, we illustrate the layered graph for $\mathcal{Q}_p = \{5, 3, 1\}$, $m = 2$, and $T = 4$. The label $(t = 1, 2)$ represents the concatenation of both the first and second time periods. More precisely, the layer $(t = 1, 2)$ aims to capture the connection between the first two periods, while encoding the compatibility restrictions of the prices between both periods. We add two special nodes to the graph: the *source* and the *sink* (the total number of nodes in the graph is then $2 + (T - m + 1) \cdot |\mathcal{Q}_p|^m$). We call an ordered pair of m -tuples, (\mathbf{x}, \mathbf{y}) , *price-compatible* if $(x_2, x_3, \dots, x_m) = (y_1, y_2, \dots, y_{m-1})$, that is, the prices are consistent at the overlapping time periods. The graph edges are given by $A = \{(\mathbf{x}, t), (\mathbf{y}, t + 1) | \mathbf{x}, \mathbf{y} \in \mathcal{Q}_p^m, 1 \leq t \leq T - 1\}$ such that (\mathbf{x}, \mathbf{y}) are price-compatible. We define weights on these edges as $w((\mathbf{x}, t), (\mathbf{y}, t + 1)) = (y_m - c_{t+1})d_{t+1}(y_m, \dots, y_1, x_1)$. In other words, the weight is equal to the profit obtained by setting the price y_m at time $t + 1$. Finally, we add arcs from the *source* node to all nodes at time $t = m$ with weight equal to the profit obtained from the first m prices: $w((\text{source}, (\mathbf{x}, m))) = \sum_{t=1}^m (x_t - c_t)d_t(x_t, \dots, x_0)$. We also connect all nodes in the last period, $t = T$, to the *sink* with zero weight. Note that the graph we constructed forms a directed layered graph since the edges exist only between nodes in consecutive periods. Having constructed the layered graph for a given instance of the profit maximization problem, Proposition 1 summarizes the equivalence between the profit maximization problem and finding the maximum weighted path in the graph.

Figure 1 Layered Graph When $\mathcal{Q}_p = \{q^0 = 2, q^1 = 1\}$, $x_r = 2$ for $t = 1$, $\theta = 0.5$, $\varepsilon = 0.25$, and $T = 4$, with Reference Prices Rounded Down to the Nearest Multiple of 0.25 [Color figure can be viewed at wileyonlinelibrary.com]



PROPOSITION 1. Consider the layered graph construction, as explained above. Any path P from the source to the sink corresponds to a price assignment (p_1, p_2, \dots, p_T) . Moreover, the sum of the weights of the edges on P corresponds to the total profit in the profit maximization problem. As a result, the two problems are equivalent.

PROOF. Since we have constructed a directed layered graph, any path from the source to the sink must use exactly one node from each layer for periods $\{m, \dots, T\}$. Since the edges are only between price-compatible nodes, there is exactly one price used at each period $\{1, \dots, T\}$. Consequently, this leads to a price value for each period. In addition, summing up the weights of the edges on the path yields the total profit. \square

This shows that the profit maximization problem is equivalent to finding the maximum weighted path (MWP) in the layered graph. The MWP problem is in general NP-hard (see Karp (1972) or Theorem 8.11 in Schrijver 2003)). However, the graph in our case is a directed acyclic graph, so that one can use dynamic programming to find the maximum weight path (Morávek 1970) in linear time in the number of edges in the graph, namely, $O(T|Q_p|^{m+1})$ for the unconstrained version of the problem. Note that to solve the MWP between two nodes $s, t \in V$ in a directed layered graph $D = (V, A, w)$, where V denotes the set of vertices, A the set of arcs, and w the vector of weights, one can solve a compact linear program (see, e.g., Ahuja et al. 1988).

3.2. Complexity Results

So far, we have considered the profit maximization problem without any constraints on prices. As discussed, the runtime complexity for the unconstrained version of the problem is $O(T|Q_p|^{m+1})$. We next present appropriate modifications to the graphical representation that allow us to capture the business rules presented in section 2.1.

Case 1. Constraining prices by restricting the graph. Very often, practical requirements dictate specific rules that prohibit certain price variations. For example, a markdown policy requires the prices in subsequent periods to always be non-increasing. An additional example is to restrict the price to be below a certain value. In such cases, one can simply delete the set of nodes and arcs that violate the rules of interest. In the markdown policy, all arcs connecting a lower price to a higher price in the subsequent time period are deleted. Such deletions decrease the graph size relative to the unconstrained case and hence, may improve the runtime.

Case 2. Limiting the number of price changes. As discussed, an important business rule is to restrict the

number of price changes to L . For each period $t \in \{m, \dots, T\}$, we construct the nodes (x, l, t) where $x \in Q_p^m$, $l \in \{0, 1, \dots, L\}$ and $t \in \{1, \dots, T\}$. In this case, we maintain a parameter $l \in \{1, \dots, L\}$ that counts the number of price changes used so far. An edge between nodes (x, l_1, t) and $(y, l_2, t + 1)$ exists if and only if:

1. (x, y) are price-compatible, and
2. $l_2 = l_1 + 1$ when $y_m \neq x_m$ and $l_2 = l_1$ otherwise.

In other words, the graph edges ensure that we correctly count the number of price changes used so far. The edge weights are: $w((x, l_1, t), (y, l_2, t + 1)) = (y_m - c_{t+1})d_{t+1}(y_m, \dots, y_1, x_1)$. As before, the path from the source to the sink with the maximum weight yields the optimal price values. Since the graph size increases by L , the runtime complexity is $O(TL|Q_p|^{m+1})$.

Case 3. No-touch constraints. We want to restrict the minimal duration S between two successive price changes. One can maintain a parameter at each node to denote the number of periods before which a price change can occur. For each period $t \in \{1, \dots, T\}$, we construct the nodes (x, s, t) with $x \in Q_p^m$, while satisfying the no-touch constraints (i.e., for any $x = (x_1, \dots, x_m)$ if $x_i \neq x_{i-1}$ for $i > 1$, then $x_{i+1}, \dots, x_{\min(i+S, m)}$ should be set to x_i) and $s \in \{0, 1, \dots, S\}$. Here, the parameter s in the node denotes the number of time periods away from t until a price change is allowed. Namely, to ensure that the no-touch constraint is satisfied (i.e., we have a minimum of S periods between two successive price changes), we need to ensure that the price during the next S periods is x_i . An edge between (x, s_1, t) and $(y, s_2, t + 1)$ exists if and only if all of the following are satisfied:

1. (x, y) are price-compatible,
2. $s_2 = \max(s_1 - 1, 0)$ if $s_1 \geq 0$ and $y_m = x_m$ (i.e., no price change at $t + 1$), and
3. $s_2 = S$ if $s_1 = 0$ and $y_m \neq x_m$ (i.e., a price change at $t + 1$).

The edges ensure that we correctly count the number of periods before a price variation can occur. The edges' weights are defined, as before, to capture the profit at time $t + 1$: $w((x, s_1, t), (y, s_2, t + 1)) = (y_m - c_{t+1})d_{t+1}(y_m, \dots, y_1, x_1)$. In addition, we delete the nodes with price vectors that do not satisfy the no-touch constraint. Deleting these nodes can significantly reduce the size of the graph, as we explain next. Recall that we maintain a tuple of m prices in each node (x, s, t) . The vector x captures the prices in the past $m - 1$ periods and the current price (i.e., at time t). The no-touch constraint requires that if one of

the prices $x_i \neq x_{i+1}$ (for $i < t - 1$), we then know that a valid node must have $x_{i+1+k} = x_{i+1}$, for $k \leq t, 1 \leq k \leq S$. As a result, we can have at most $\lceil \frac{m}{S} \rceil$ price changes in any given node. Consequently, the total number of nodes in any time period is of the order $O(|Q_p|^{\lceil \frac{m}{S} \rceil})$.² Note that the nodes in which s is positive have exactly one outgoing edge, whereas the nodes with $s = 0$ have up to $T(S + 1)|Q_p|$ outgoing edges. The total number of edges is thus $O(TS|Q_p|^{\lceil \frac{m}{S} \rceil} + T|Q_p|^{\lceil \frac{m}{S} \rceil + 1})$, corresponding to the running time to solve the profit maximization problem.

Recall that in our model, the demand at time t depends on the current price as well as on the m past prices. Such a model often suffers from the end-of-horizon effects (see, e.g., Herer and Tzur 2001). In particular, the optimization creates an artificial advantage to schedule price reductions toward the end of the horizon, as the price effect on future demand is overlooked. A possible way to address this issue is to consider a rolling horizon such that the price at time T affects demand at times 1 to m (similarly, the price at time $T - 1$ affects demand at time T but also at times $1, 2, \dots, m - 1$). This modification is equivalent to replicating the horizon of T periods an infinite number of times. Note that the modified problem (actually, we only modify m demand functions) does not suffer from the end-of-horizon effect anymore. One can solve this version of the problem by using our graphical representation with a planning period of $2T$. This only doubles the graph size, so that the run time complexity remains the same as the unconstrained case.

We summarize the runtimes in Table 1. Incorporating business rules such as markdown prices or no-touch constraints improve the optimization problem by reducing the instance size. As mentioned, our approach is independent of the demand structure so that all complexity results hold for any non-linear demand function. Finally, for the case with no-touch constraints, when $S \geq m$, the algorithm is very efficient (no longer exponential in the memory parameter). However, when m is large (e.g., polynomial in T), these algorithms are no longer tractable and may take several hours to solve (more details are reported in section 6). We next discuss the hardness of the profit maximization problem for the case with a large memory parameter m . Note that the model studied in this

section including the different business rules can be written and solved as a dynamic program (details are omitted for conciseness).

3.3. Hardness Result

We show that when the memory parameter is large, the profit maximization problem is NP-hard. We present a reduction of the weighted maximum 3-satisfiability (Max-3-SAT) (see, e.g., Karp 1972, Schrijver 2003) to our problem. Max-3-SAT is a classical NP-hard problem. We next introduce some terminology. A literal x is a variable that can take a boolean value of true (1) or false (0), and \bar{x} denotes the negation of x . A clause c is a disjunction of three literals, that is, $c = (x_i \vee x_j \vee \bar{x}_k)$. A clause is said to be satisfied if an assignment of boolean values to the literals x_i makes the clause true (i.e., at least one of the literals in the clause is true). The maximum satisfiability problem seeks to find an assignment of booleans to the literals such that the maximum number of clauses are satisfied. In the weighted version, each clause c_s is given a non-negative weight w_s . The weighted Max-3-SAT problem is to find an assignment of boolean values that maximizes the weighted sum of satisfied clauses.

THEOREM 1. *The profit maximization problem is NP-hard when the memory parameter $m = \Omega(T)$,³ unless $P = NP$. Moreover, it is NP-hard to approximate the profit maximization problem to $7/8 + \varepsilon$ factor, for any $\varepsilon > 0$.*

PROOF. To prove the NP-hardness of the profit maximization problem, we consider an arbitrary instance of the weighted Max-3-SAT, I_{sat} , and construct a corresponding instance I_{Π} of the profit maximization problem with a general demand function d_{Π} . We show that any solution of I_{Π} corresponds to a solution of I_{sat} , thereby proving that a polynomial-time algorithm for the profit maximization problem when m is $\Omega(T)$ is not possible, unless $P = NP$. Moreover, since it is NP-hard to approximate the Max-3-SAT to any approximation factor better than $7/8$ (see, e.g., Hastad 2001), the same inapproximability result applies to the profit maximization problem. \square

Let the Max-3-SAT instance, I_{sat} , be defined using n literals x_i ($i = 1, \dots, n$) and m clauses c_j ($j = 1, \dots, m$) each of which has a non-negative weight w_j associated with it. Note that the total number of clauses can only be at most $\binom{2n}{3} = \Theta(n^3)$ (which is polynomial in n).

We construct an instance of the profit maximization problem, I_{Π} , as follows. We let $T = n$ and $m = n - 1$.

Table 1 Summary of Complexity Results Under Different Business Rules

Business rules	Runtime complexity
Unconstrained problem	$O(T Q_p ^{m+1})$
Limited number of price changes	$O(TL Q_p ^{m+1})$
Markdown prices	$O(T Q_p ^{m+1})$
No-touch constraints	$O(T Q_p ^{\lceil \frac{m}{S} \rceil} \max(Q_p , S))$
End-of-horizon effects	$O(T Q_p ^{m+1})$

The price ladder is simply $\mathcal{Q}_p = \{2, 1\}$. The price at any period t will correspond to the literal $x_t = 0$ (if $p_t = 2$) or $x_t = 1$ (if $p_t = 1$) (and correspondingly, $\bar{x}_t = 1$ or 0). Consider the following demand function for the profit maximization problem:

$$d_t(\mathbf{p}_t) = \sum_{j \in J_t} w_j c_j(\mathbf{p}_t) / p_t,$$

where J_t is the set of indices of clauses such that the maximum indexed literal in the clause is x_t or \bar{x}_t , and $c_j(\mathbf{p}_t) = 1$ if the corresponding assignment of the literals in the clause satisfies it and otherwise $c_j(\mathbf{p}_t) = 0$. Given a price vector (i.e., an assignment of literals), it is easy to evaluate the demand. Moreover, the demand function depends exactly on two past prices for every time period $t > 2$ and can be specified using a polynomial number of inputs.

Since we did not impose any assumption on the demand function in the profit maximization formulation, one can encapsulate the above weighted Max-3-SAT objective into the profit maximization problem. Thus, our finite horizon profit maximization problem reduces to:

$$\max_{\mathbf{p}_t} \sum_{t=1}^T p_t d_t(\mathbf{p}_t) = \max_{\mathbf{p}_t} \sum_{j \in J_t} w_j c_j(\mathbf{p}_t) \stackrel{(a)}{=} \max_{\mathbf{p}_T} \sum_{j=1}^m w_j c_j(\mathbf{p}_T), \quad (3)$$

where the equality in (a) holds since $\{J_t\}$ partitions the set of clauses. Therefore, solving the profit maximization problem in polynomial time would contradict the NP-hardness of the Max-3-SAT—concluding the proof. Since the reduction is exact, any inapproximability bound for Max-3-SAT also applies to the profit maximization problem when m is $\Omega(T)$.

Having shown that the profit maximization problem cannot, in general, admit a polynomial-time approximation algorithm with a factor better than $7/8 + \varepsilon$ (for any $\varepsilon > 0$), a natural question is: What assumptions on the demand function would render the problem approximable to an arbitrary constant factor (in polynomial space) under a large memory parameter? In the next section, we study the reference price model and show that there exists an efficient approximation method to solve the problem up to $(1 + \varepsilon)$ accuracy, for an arbitrary $\varepsilon > 0$.

4. Reference Price Model

In this section, we introduce a discrete version of the commonly used reference price model (see, e.g., Popescu and Wu 2007). In the continuous reference price model, the demand at time t is assumed to depend on the current price p_t and the reference price r_t . The latter represents the baseline price that

consumers are forming based on past prices. Recall that our setting focuses on discrete prices. In addition, customers may not form a reference price with infinite precision. Consequently, we propose to only allow reference prices that belong to a discrete ladder denoted by \mathcal{Q}_r (e.g., 5 cents intervals for an item that costs 1 dollar). We call this the *discrete reference price model*. Note that this is a special case of our general demand in Equation (1), where the memory parameter is large (i.e., $m = T$) but the contributions of past prices are decaying by a constant factor (see more details below). We first develop an exact algorithm that runs in polynomial time with respect to the input price ladder. Subsequently, we approximate a general demand with linear past price effects using the discrete reference price model and derive bounds on the profit performance. These results will prove useful in the context of multiple items (see section 5).

4.1. Discrete Reference Price Model

As discussed, instead of considering a general demand model that depends on the current and m past prices, the reference price model depends on the current price and on the (continuous) reference price r_t . Specifically, the reference price follows the following update equation:

$$r_t = (1 - \theta)p_{t-1} + \theta r_{t-1}, \quad (4)$$

where $0 \leq \theta < 1$ represents the weight that consumers allocate to past prices. For example, the demand model in Fibich et al. (2003) with a (linear) symmetric reference price effect is given by:

$$d_t = a_t - \beta^0 p_t - \phi(p_t - r_t). \quad (5)$$

The parameter ϕ denotes the price sensitivity with the reference price, and $(\beta^0 + \phi)$ corresponds to the price sensitivity with p_t . The reference price at time t can be rewritten as follows:

$$\begin{aligned} r_t &= (1 - \theta)p_{t-1} + \theta(1 - \theta)p_{t-2} + \theta^2(1 - \theta)p_{t-3} + \dots \\ &= (1 - \theta) \sum_{k=1}^T \theta^{k-1} p_{t-k}. \end{aligned}$$

Thus, the demand at time t from Equation (5) can also be written in terms of the prices:

$$\begin{aligned} d_t(p_t, p_{t-1}, \dots, p_{t-T}) &= a_t - (\beta^0 + \phi)p_t + \phi \sum_{k=1}^T (1 \\ &\quad - \theta)\theta^{k-1} p_{t-k}. \end{aligned} \quad (6)$$

Equation (6) indeed depicts a model that depends on the current and m past prices (with $m = T$). More generally, we consider a non-linear reference price-demand model of the form:

$$d_t(p_t, r_t) = f_t(p_t) + g(p_t - r_t). \tag{7}$$

Here, the demand model includes two additive parts: (i) the price and seasonality effects captured by the function $f_t(\cdot)$, and (ii) the reference price effect modeled by the function $g(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ that depends on the difference between the price and the reference price. We assume that the function $g(\cdot)$ is G -Lipschitz (i.e., $|g'(x)| \leq G$ for all $x \in \mathbb{R}$). The assumption that the function $g(\cdot)$ is G -Lipschitz is a common assumption that yields analytical tractability (see, e.g., Chaib-draa and Müller 2006, Dockner et al. 2000). In our context, this assumption means that a small price change will not increase or decrease the demand by an arbitrary amount, which is reasonable in most practical settings. We note that the demand model in Equation (7) includes cases where the reference price effect is asymmetric.

Recall that the reference price represents the price that consumers are willing to pay for the item based on past prices. Given that consumers may not form a reference price with infinite precision, it seems reasonable to assume that r_t lies in a discrete set where the values are rounded to some finite level (e.g., 5 cents). We denote the discrete reference price by \hat{r}_t , and its set of values by $\mathcal{Q}_r = \{r^0 > r^1 > \dots > r^N > \dots > r^N\}$. We have: $r^0 = q^0$ and $r^N = q^Q$ (the smallest element of the price ladder). We next introduce the discrete reference price at time t which is obtained by:⁴

$$\hat{r}_t = \text{round} \left[(1 - \theta)p_{t-1} + \theta\hat{r}_{t-1} \right]. \tag{8}$$

We next show that the discrete reference price model approaches the continuous reference price model in the limit of the discretization of the price ladder. We choose \mathcal{Q}_r such that $r^k - r^{k-1} = \epsilon, \forall k = 1, 2, \dots, N$, for some given $\epsilon > 0$.

PROPOSITION 2. Consider the continuous reference price model in Equation (4) and the proposed discrete reference price model with precision $\epsilon > 0$ as in Equation (8). Then, the difference between the continuous and discrete reference prices at time t is bounded by:

$$|\hat{r}_t - r_t| \leq \frac{1 - \theta^{t-1}}{1 - \theta} \epsilon, \tag{9}$$

where r_t and \hat{r}_t denote the continuous and the discrete reference prices at time t , respectively.

PROOF. For the first period, we have: $\hat{r}_1 = \text{round} \left[(1 - \theta)p_0 + \theta r_0 \right] = r_1 \pm \epsilon$. Then, for $t = 2$, one can write: $\hat{r}_2 = \text{round} \left[(1 - \theta)p_1 + \theta\hat{r}_1 \right]$

$= \text{round} \left[(1 - \theta)p_1 + \theta(r_1 \pm \epsilon) \right]$. Recall that $r_2 = (1 - \theta)p_1 + \theta r_1$ and therefore, one can bound the rounding error in \hat{r}_2 as follows: $\hat{r}_2 = \text{round} \left[r_2 \pm \theta\epsilon \right] = r_2 \pm (\theta + 1)\epsilon$. We next proceed by induction on t . We assume that the claim is true for $t = k$: $\hat{r}_k = r_k \pm (\sum_{u=0}^{k-1} \theta^u)\epsilon$. We next show that the claim holds for $t = k + 1$. We have:

$$\begin{aligned} \hat{r}_{k+1} &= \text{round} \left[(1 - \theta)p_k + \theta\hat{r}_k \right] \\ &= \text{round} \left[(1 - \theta)p_k + \theta(r_k \pm (\sum_{u=0}^{k-1} \theta^u)\epsilon) \right]. \end{aligned}$$

Using the fact that $r_{k+1} = (1 - \theta)p_k + \theta r_k$, we obtain:

$$\begin{aligned} \hat{r}_{k+1} &= \text{round} \left[r_{k+1} \pm \theta(\sum_{u=0}^{k-1} \theta^u)\epsilon \right] = r_{k+1} \pm (\sum_{u=0}^k \theta^u)\epsilon \\ &= r_{k+1} \pm \left(\frac{1 - \theta^{k+1}}{1 - \theta} \right) \epsilon, \end{aligned}$$

concluding the proof. □

Since $0 \leq \theta < 1$, the difference is guaranteed to be within a constant factor of ϵ . We next show that in the limit of the reference price ladder discretization, the demand and total profit also approach their continuous model counterparts. This shows that the discrete reference price model is a good approximation of the continuous model, while providing the benefit of modeling customer behavior more realistically. We propose two simple ways of selecting the discrete reference price ladder: either set $\mathcal{Q}_r = \mathcal{Q}_p$ or consider a discretization of 1 cent. The former choice models the scenario when customers remember past prices and select one of the past prices to be the reference. The latter choice captures the fact that consumer preferences can only be granular up to 1 cent.

COROLLARY 1. Consider the demand model in Equation (7) and the discrete reference price model with precision $\epsilon > 0$. Then, the difference between the demand value and the total profit are bounded by:

$$\begin{aligned} |\hat{d}_t - d_t| &= |g(p_t - \hat{r}_t) - g(p_t - r_t)| \leq G|\hat{r}_t - r_t| \\ &\leq G \frac{1 - \theta^{t-1}}{1 - \theta} \epsilon, \end{aligned} \tag{10}$$

$$|\hat{\Pi} - \Pi| \leq T(q^0 - c_{min})G \frac{1 - \theta^{T-1}}{1 - \theta} \epsilon, \tag{11}$$

where \hat{d}_t and $\hat{\Pi}$ denote the demand at time t and the total profit from the discrete model, respectively. Here, c_{min} denotes the minimal cost value, $c_{min} = \min_t c_t$.

Corollary 1 follows from Proposition 2 and from the fact that $g(\cdot)$ is G -Lipschitz. For the special case with a linear demand model as in Equation (5), we have:

$$|\hat{d}_t - d_t| = \phi |\hat{r}_t - r_t| \leq \phi \frac{1 - \theta^{t-1}}{1 - \theta} \epsilon. \quad (12)$$

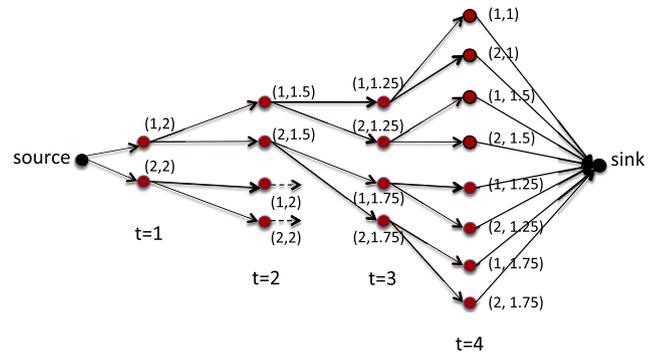
One can also consider a linear asymmetric reference price model (see, e.g., Popescu and Wu 2007), that is, $d_t = f_t(p_t) - \phi^{\text{loss}}(p_t - r_t)^+ + \phi^{\text{gain}}(r_t - p_t)^+$. This captures the fact that a price reduction does not have the same effect as a price increase. In this case, one can rewrite the bound in Equation (12) with $\phi = \max(\phi^{\text{gain}}, \phi^{\text{loss}})$.

We next present a polynomial algorithm for the profit maximization problem under the discrete reference price model (for a given reference price ladder). More precisely, we propose a dynamic program algorithm that is polynomial in $|\mathcal{Q}_p|$ and $|\mathcal{Q}_r|$. Note that \mathcal{Q}_r can be an input to the algorithm (with any precision) and that the algorithm is independent of the way of rounding.

Graphical Representation. Suppose we are given a discrete reference price ladder $\mathcal{Q}_r = \{r^0 > r^1 > r^2 > \dots > r^N\}$, where $r^0 = q^0$, $r^N = q^N$ such that $r^k - r^{k-1} = \epsilon; \forall k = 1, 2, \dots, N$, for some $\epsilon > 0$. We can construct a layered graph such that computing a maximum weighted path yields the optimal prices. For each period $t \in [1, \dots, T]$, we construct the nodes (\mathbf{x}, t) , where $\mathbf{x} = (x_p, x_r) \in \mathcal{Q}_p \times \mathcal{Q}_r$. We add two special nodes to the graph: the source and the sink (the total number of nodes in the graph is then $2 + T|\mathcal{Q}_p||\mathcal{Q}_r|$). As before, we call a pair of nodes, $(\mathbf{x}, t), (\mathbf{y}, t + 1)$, price-compatible if $(1 - \theta)x_p + \theta x_r - \epsilon \leq y_r \leq (1 - \theta)x_p + \theta x_r + \epsilon$ for $1 \leq t \leq T - 1$, that is, the reference prices are updated in a consistent manner in consecutive periods. We then add a set of arcs to the graph given by $A = \{(\mathbf{x}, t), (\mathbf{y}, t + 1) | \mathbf{x}, \mathbf{y} \in \mathcal{Q}_p \times \mathcal{Q}_r, 1 \leq t \leq T - 1\}$ such that $(\mathbf{x}, t), (\mathbf{y}, t + 1)$ are price-compatible. We define weights on these edges as $w((\mathbf{x}, t), (\mathbf{y}, t + 1)) = (y_p - c_{t+1})d_{t+1}(y_p, y_r)$. In other words, this weight is equal to the profit obtained by introducing price y_p at time $t + 1$. Finally, we add arcs from the source to all nodes at time $t = 1$ with weight $(x_p - c_1)d_1(x_p, x_r)$, where we assume that $x_r = 1$. We also connect all the nodes at $t = T$ to the sink with zero weight. Note that the graph forms a directed layered graph since the edges exist only between nodes in consecutive periods. In Figure 2, we illustrate the layered graph for $\mathcal{Q}_p = \{q^0 = 2, q^1 = 1\}$, $x_r = 2$ for $t = 1$, $\theta = 0.5$, $\epsilon = 0.25$, and $T = 4$ with the reference prices rounded down in $\mathcal{Q}_r = \{1, 1.25, 1.5, 1.75, 2\}$.

In the same spirit as in Proposition 1, optimizing prices under the discrete reference price model and finding the maximum weighted path in the graph are equivalent.

Figure 2 Layered Graph When $\mathcal{Q}_p = \{q^0 = 2, q^1 = 1\}$, $x_r = 2$ for $t = 1$, $\theta = 0.5$, $\epsilon = 0.25$, and $T = 4$, with Reference Prices Rounded Down to the Nearest Multiple of 0.25 [Color figure can be viewed at wileyonlinelibrary.com]



PROPOSITION 3. Consider the layered graph for the discrete reference price model (8). Any path P from the source to the sink corresponds to a price assignment (p_1, p_2, \dots, p_T) , and the sum of the weights of the edges on P corresponds to the total profit. Thus, the two problems are equivalent.

The proof of Proposition 3 follows directly from the way we constructed the layered graph and is omitted for conciseness. The time complexity of finding the maximum profit path is $O(T|\mathcal{Q}_p|^2|\mathcal{Q}_r|)$. In the special case where $\mathcal{Q}_r = \mathcal{Q}_p$ (i.e., the reference price is rounded to prices in the price ladder), this reduces to $O(T|\mathcal{Q}_p|^3)$. Recall that our goal is to optimize the profit for a demand model with a large memory parameter (i.e., $m = T$). We have proposed an alternate discrete reference price model which effectively behaves as if the memory parameter was equal to two by merging all the required information in a single quantity (the discrete reference price at each time). Finally, one can easily incorporate business constraints in the same way as in section 3.2. As shown in section 3.2, this method runs in < 0.1 second for realistic-size instances.

4.2. Reference Price Model Approximation

As discussed in section 3.3, for a general demand function with a large m , our method may not be applicable as the time complexity is exponential in m . Without specific assumptions on the structure of the problem, it is not clear how one can develop an efficient algorithm to solve the problem. However, as we saw in the previous section, the discrete reference price model can be solved efficiently. In this section, we present a way to approximate several demand functions using the discrete reference price model and we derive bounds on the quality of the approximation. We consider three common demand models that can be estimated from

data: linear past prices, log-log past prices, and log-linear past prices. We next present the analysis for demand models with linear past prices effects as in Equation (13) and include the analysis for the log-log and log-linear models in Appendix 8. In all three cases, we develop a procedure to approximate the true demand function with a large m and provide an efficient algorithm to solve the problem.

We consider a model where the demand at time t depends on the current and past m prices, where $m = T$ (i.e., a setting with a large memory parameter):

$$d_t(\mathbf{p}_t) = f_t(p_t) + \beta^1 p_{t-1} + \dots + \beta^T p_{t-T}. \quad (13)$$

The parameters β^1, \dots, β^T and the functions $f_t(\cdot)$ are estimated from data. A common assumption—validated by our data—requires the gain parameters to be non-negative and non-increasing: $\beta^1 \geq \dots \geq \beta^T \geq 0$. In addition, β^1 is often much larger relative to the other coefficients. Intuitively, the most recent price has a higher impact relative to prices further in the past. Recall that the time complexity of our dynamic program is exponential in m , and thus not tractable. Our goal is to use a similar methodology as for the discrete reference price model and approximate the demand function in Equation (13) to depend only on the current price p_t and on a single additional variable \tilde{r}_t , called the modified reference price. We require \tilde{r}_{t+1} to depend only on \tilde{r}_t and p_t . We naturally impose the following relation:⁵

$$\tilde{r}_t = (1 - \tilde{\theta})p_{t-1} + \tilde{\theta}\tilde{r}_{t-1}, \quad (14)$$

where $\tilde{\theta}$ is a design parameter that aims to approximate the true demand from Equation (13) using the following approximated demand model:

$$\tilde{d}_t(p_t, \tilde{r}_t) = f_t(p_t) + \tilde{\phi}p_t - \tilde{\phi}(p_t - \tilde{r}_t) = f_t(p_t) + \tilde{\phi}\tilde{r}_t. \quad (15)$$

In other words, Equations (14) and (15) aim to approximate the true demand from Equation (13) by carefully choosing $\tilde{\theta}$ and $\tilde{\phi}$. Specifically, we need to approximate a linear function with T coefficients by using an approximation with only two parameters. As in Equation (6), one can write:

$$\tilde{d}_t(p_t, \tilde{r}_t) = f_t(p_t) + \tilde{\phi} \sum_{k=1}^T (1 - \tilde{\theta})\tilde{\theta}^{k-1} p_{t-k}. \quad (16)$$

Observe that if the coefficients β^i are decreasing by a constant factor, that is, β^{i+1}/β^i is constant for all i , we then obtain as a special case, the (linear) reference price model from Equation (5) and the approximation is exact. When the ratios are not constant, we obtain an approximation.

Given that β^1 is often larger relative to the other coefficients (i.e., the last price has a higher effect relative to further past prices), we want to ensure that we match the effect of the most recent past price p_{t-1} . We thus require: $\tilde{\phi}(1 - \tilde{\theta}) = \beta^1$, or equivalently $\tilde{\phi}$ is always set such that $\tilde{\phi} = \beta^1/(1 - \tilde{\theta})$. This leaves us with a single degree of freedom ($\tilde{\theta}$) to approximate the $T-1$ remaining factors. Equation (16) becomes:

$$\tilde{d}_t(p_t, \tilde{r}_t) = f_t(p_t) + \beta^1 p_{t-1} + \beta^1 \sum_{k=2}^T \tilde{\theta}^{k-1} p_{t-k}. \quad (17)$$

We next propose three different approximations that depend on the value of $\tilde{\theta}$:

1. $\tilde{\theta}_{\min} = \min_{i=1, \dots, T-1} \beta^{i+1}/\beta^i$.
2. $\tilde{\theta}_{\max} = \max_{i=1, \dots, T-1} \beta^{i+1}/\beta^i$.
3. $\tilde{\theta}_{LS}$ is defined so that it minimizes the following error function:

$$\sum_{k=2}^T [\beta^k - \beta^1 \tilde{\theta}^{k-1}]^2. \quad (18)$$

The value of $\tilde{\theta}_{\min}$ (resp. $\tilde{\theta}_{\max}$) can be interpreted as a lower (resp. upper) envelope of the true demand function that belongs to all the linear demand models with a reference price.

One can compute $\tilde{\theta}_{LS}$ numerically by searching, as minimizing the function in Equation (8) is a single-dimensional optimization problem. A more conventional way to find the least-squares estimate is not to impose $\tilde{\phi}(1 - \tilde{\theta}) = \beta^1$, but instead estimate both parameters $\tilde{\theta}_{LS}$ and $\tilde{\phi}$. In our model, however, we are exploiting the fact that β^1 is higher than the other coefficients and want to ensure that the past-price effect is fully matched by the approximation. More importantly, imposing $\tilde{\phi}(1 - \tilde{\theta}) = \beta^1$ allows us to reduce the problem to a single-dimensional problem, which is easier to solve.

The corresponding demand approximations are labeled as d_t^{\min} , d_t^{\max} , and d_t^{LS} . Similarly, the optimal profits using the different approximations are denoted by Π^{\min} , Π^{\max} , and Π^{LS} . For instance, Π^{\min} corresponds to solving the approximated problem with $\tilde{\theta} = \tilde{\theta}_{\min}$ (recall that the problem becomes tractable since one can apply the efficient algorithm from section 4.1).

PROPOSITION 4. Consider the true demand function from Equation (13) with a large memory parameter (i.e., $m = T$). Consider also the approximation using the continuous reference price model with the three different $\tilde{\theta}$ defined above. Then, we have:

$$\begin{aligned} \Pi^{\min} &\leq \Pi^{LS} \leq \Pi^{\max}, \\ \Pi^{\min} &\leq \Pi^{\text{True}} \leq \Pi^{\max}, \end{aligned}$$

where Π^{True} corresponds to the optimal profit using the true demand function from Equation (13).

PROOF. We show that using the approximation with $\tilde{\theta}_{\min}$ (resp. $\tilde{\theta}_{\max}$) provides a lower (resp. upper) bound relative to any other approximation within the class of approximations we are considering (in particular, the one using $\tilde{\theta}_{LS}$), as well as relative to the true demand function. For any given price vector \mathbf{p}_t , we have:

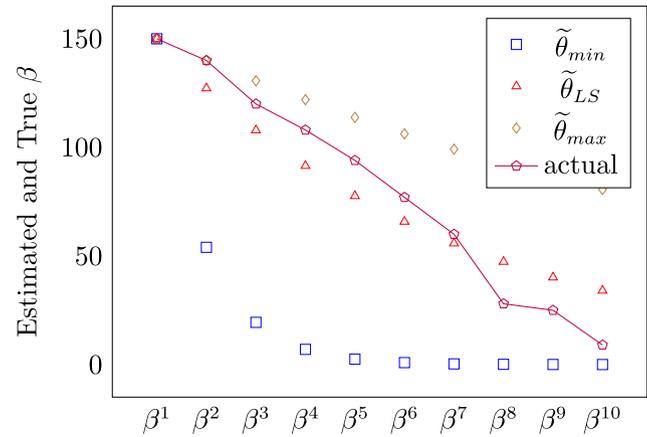
$$\begin{aligned} \Pi^{\min} &= \sum_{t=1}^T (p_t - c_t) d_t^{\min} \leq \sum_{t=1}^T (p_t - c_t) d_t^{LS} \\ &\leq \sum_{t=1}^T (p_t - c_t) d_t^{\max} = \Pi^{\max}, \\ \Pi^{\min} &= \sum_{t=1}^T (p_t - c_t) d_t^{\min} \leq \sum_{t=1}^T (p_t - c_t) d_t^{\text{True}} \\ &\leq \sum_{t=1}^T (p_t - c_t) d_t^{\max} = \Pi^{\max}. \end{aligned}$$

The four inequalities follow from the fact that β^i ($\forall i = 1, \dots, T$) are all non-negative and that under the same price vector, using a smaller value of $\tilde{\theta}$ can only decrease the demand and hence the profit (under a given price vector, $d_t^{\min} \leq d_t^{LS} \leq d_t^{\max}$ and $d_t^{\min} \leq d_t^{\text{True}} \leq d_t^{\max}$ for all t). \square

Note that the results in Propositions 2 and 4 imply that one can approximate the problem with the demand function in Equation (13) by using the discrete reference price model. This allows us to solve the problem efficiently (polynomial time), while having a guarantee on the quality of the approximation. In Figure 3, we consider a specific instance with $T = 10$ and plot the demand approximations using $\tilde{\theta}_{\min}$, $\tilde{\theta}_{LS}$, and $\tilde{\theta}_{\max}$. The values in the y-axis correspond to the true and approximated (using the three approximations) values of β^t for $t = 1, 2, \dots, 10$. One can see that both the true values and the ones obtained using $\tilde{\theta}_{LS}$ are lower bounded by the $\tilde{\theta}_{\min}$ approximation and upper bounded by the $\tilde{\theta}_{\max}$ approximation. In addition, the approximation based on $\tilde{\theta}_{LS}$ yields a good approximation. As we show in section 6, for randomly generated instances, the 25th percentile and median are 97.7% and 99.4% relative to the optimal profit, respectively.

To summarize, one can solve the problem with $\tilde{\theta}_{LS}$ and obtain an approximation solution efficiently. In addition, we have lower and upper bounds on the profit performance by solving the problem with $\tilde{\theta}_{\min}$ and $\tilde{\theta}_{\max}$. As discussed, for the reference price model,

Figure 3 An Example of How the Estimated Parameters Compare to the True Parameters [Color figure can be viewed at wileyonlinelibrary.com]



we have $\Pi^{\min} = \Pi^{\max}$ so that the approximation is exact, whereas for any other demand function as in Equation (13), the approximation is not exact but provides a good computational performance (see section 6).

5. Multiple Items

We extend the graphical model for the profit maximization problem with multiple items. We consider a setting with $n > 1$ items for which the retailer needs to set the prices of each item at each period. As before, we assume that the current demand of item i depends on the current price of item i , p_t^i , and the past self prices $p_{t-1}^i, \dots, p_{t-m_i}^i$. In addition, the demand of item i now also depends on the vector of current other prices: p_t^j for $j \neq i$. This aims to capture cross-item effects on demand, that is, a price variation on one item can affect the sales of other items. For example, a price reduction on item j may decrease the sales of item i (in this case, items i and j are substitutes) or increase the sales of item k (in this case, items j and k are complements).

We first observe that one can extend the graphical representation from section 3 to the case with n items by expanding the size of each node in the graph. Namely, in each node, we maintain a tuple of $m \times n$ prices: m prices for each items. The total number of nodes at each period is then $|\mathcal{Q}_p|^{mn}$. To simplify the exposition, we assume that each item has the same memory parameter, $m_i = m \forall i = 1, 2, \dots, n$, and the same price ladder \mathcal{Q}_p . Consequently, we obtain a naive extension of the previous results. For example, the time complexity of the unconstrained problem for n items is $O(T|\mathcal{Q}_p|^{n(m+1)})$. Unfortunately, this approach is intractable (the time complexity grows exponentially with m and n) and even for instances

with relatively small values of m and n , this may take several hours to solve (for more details, see section 6). To address this issue, we propose two alternative approaches that allow us to solve the problem more efficiently. Both methods borrow the intuitions and results developed in section 4. The first method is based on having a reference price for each item, whereas the second considers a single virtual reference price for all the items.

5.1. Model with n Reference Prices

We consider that consumers form a reference price based on past prices for each item separately. Accordingly, the demand of item i at time t is given by:

$$d_t^i(p_t^1, p_t^2, \dots, p_t^n, r_t^i) = f_t^i(p_t^1, p_t^2, \dots, p_t^n) + g_t^i(p_t^i - r_t^i),$$

where the first term represents the effect of all current prices, and the second term captures the effect of the reference price of item i at time t , r_t^i , which can be written as

$$r_t^i = \theta_i r_{t-1}^i + (1 - \theta_i) h^i(p_{t-1}^1, p_{t-1}^2, \dots, p_{t-1}^n).$$

In this case, the function $h^i(\cdot)$ can depend on all prices at the previous period. In its simplest form, $h^i(p_{t-1}^1, p_{t-1}^2, \dots, p_{t-1}^n)$ would be equal to p_{t-1}^i . More generally, $h^i(\cdot)$ can depend on all prices at the previous period (e.g., a weighted average of all the prices). Note that the parameters $0 < \theta_i \leq 1$ can be different for each item and are estimated from data.

Using this representation, we conclude that the discrete reference price model for n items has time complexity of $O(T|Q_p|^{n+1}Q_r^n)$, where Q_r is the reference price ladder (see section 4.1). This is a significant improvement relative to the naive extension discussed above with a time complexity of $O(T|Q_p|^{n(m+1)})$.

Interestingly, one can extend the result of Proposition 4 for this model. Specifically, if the function $h^i(p_{t-1}^1, p_{t-1}^2, \dots, p_{t-1}^n) = p_{t-1}^i$ for each item $i = 1, \dots, n$, we can compute $\tilde{\theta}_{min}^i$ and $\tilde{\theta}_{max}^i$ for each i in the same way as for the setting with a single item. Hence, the result of Proposition 4 still holds, that is, $\Pi^{min} \leq \Pi^{LS} \leq \Pi^{max}$ and $\Pi^{min} \leq \Pi^{True} \leq \Pi^{max}$. More generally, when $h^i(p_{t-1}^1, p_{t-1}^2, \dots, p_{t-1}^n)$ is a general function of the prices in the previous period, the minimum (resp. maximum) bound is obtained by using $\tilde{\theta}_{min}$ (resp. $\tilde{\theta}_{max}$) together with $h_{min} = \min_{i, p_{t-1}} \{h^i(p_{t-1}^1, p_{t-1}^2, \dots, p_{t-1}^n)\}$ (resp. $h_{max} = \max_{i, p_{t-1}} \{h^i(p_{t-1}^1, p_{t-1}^2, \dots, p_{t-1}^n)\}$). Note that when the function $h^i(\cdot)$ is the average, minimum, or maximum (or any convex combination of the prices in the previous period), one can use $h_{min} = \min\{p_{t-1}^1, p_{t-1}^2, \dots, p_{t-1}^n\}$ (resp. $h_{max} = \max\{p_{t-1}^1, p_{t-1}^2, \dots, p_{t-1}^n\}$).

5.2. Model with a Single Reference Price

Given a category of items in a supermarket (e.g., coffee), customers often have a notion of how much they are willing to spend for buying one pack of coffee. They do not form a reference price for each item separately but instead, consider a reference price for the category. We call this notion of aggregate baseline the *virtual reference price*. To our knowledge, we are the first to formally introduce this concept. We consider that consumers form a single reference price r_t^V for the entire category of n items. In this case, we assume that the demand of item i at time t is given by:

$$d_t^i(p_t^1, p_t^2, \dots, p_t^n, r_t^V) = f_t^i(p_t^1, p_t^2, \dots, p_t^n) + g_t^i(p_t^i - r_t^V),$$

where the first term represents the effect of all current prices, and the second term captures the effect of the virtual reference price at time t , which is given by:

$$r_t^V = \theta_V r_{t-1}^V + (1 - \theta_V) h(p_{t-1}^1, p_{t-1}^2, \dots, p_{t-1}^n),$$

where the function $h(\cdot)$ can depend on all prices at the previous period. For example, the function $h(\cdot)$ can be a weighted average of the n past prices, the minimum, or the maximum. The parameter $0 < \theta_V \leq 1$ represents the memory of past prices and is estimated from data.

Using this representation, one can also consider discretized virtual reference prices. Then, the time complexity of the model reduces to $O(T|Q_p|^{n+1}Q_r)$, where Q_r is the number of elements in the discrete ladder of the virtual reference price. Note that if n is large or if Q_r includes many elements (i.e., the precision parameter ε is small), then this model is the only one (out of three approaches we considered) that is tractable. We compare the time complexity of the three approaches in Table 2. We further compare the three approaches computationally in section 6.5.

We next show how to extend the result of Proposition 4 for this model. This is not straightforward since the past prices of the different items can directly affect the demand of *all* items. Fortunately, we can still find the minimum and maximum profit bounds within the virtual reference price family of functions. We define $\tilde{\theta}_{min}$ (resp. $\tilde{\theta}_{max}$) to be the minimum (resp. maximum) ratio across both time periods and items, that is:

$$\tilde{\theta}_{min} = \min_{k=1, \dots, n} \min_{i=1, \dots, T-1} \beta_k^{i+1} / \beta_k^i,$$

$$\tilde{\theta}_{max} = \max_{k=1, \dots, n} \max_{i=1, \dots, T-1} \beta_k^{i+1} / \beta_k^i.$$

We also define h_{min} (resp. h_{max}) in the same way as in section 5.1. Finally, we impose the following equation: $\tilde{\phi}(1 - \theta_V) = \min_{i=1, \dots, n} \beta_i^1$. We can then conclude that the result of Proposition 4 also holds

Table 2 Summary of Time Complexity Results for Multiple Items

Model	Time complexity	$\mathcal{Q}_r = \mathcal{Q}_p$
Naive	$O(T \mathcal{Q}_p ^{n(m+1)})$	$O(T \mathcal{Q}_p ^{n(m+1)})$
n reference prices	$O(T \mathcal{Q}_p ^{n+1} \mathcal{Q}_r ^n)$	$O(T \mathcal{Q}_p ^{2n+1})$
Single reference price	$O(T \mathcal{Q}_p ^{n+1} \mathcal{Q}_r)$	$O(T \mathcal{Q}_p ^{n+2})$

for the setting with multiple items under a single virtual reference price.

Interestingly, all three methods can easily be extended to incorporate business rules. First, one can satisfy pricing constraints for each item as in section 2.1. Second, one can consider business rules that link the price changes of the different items. The most common constraint is to impose a limitation on the total number of price changes during the selling season for the entire category. Alternative examples include enforcing a relationship between prices of the different items (e.g., smaller formats should be always cheaper than larger formats) and exclusivity deals. We discuss the implementation of this type of global constraints in section 5.4. We next describe a practical model that makes the approach presented in this section even more efficient.

5.3. Practical Model with Blocks

The demand of a product is often affected only by a relatively small subset of other items' prices in the category. A typical category in a supermarket can include between 30 and 250 different items and it is clear that price variations of certain items will have no impact on the sales of other items. For instance, two items may be in the same category but have a very weak connection in terms of substitution or complementarity (e.g., dark roast and decaf products). Ultimately, the demand of a product is typically affected only by a few prices. We use this observation to refine the time complexity of the three previous methods and obtain a more efficient solution approach. Specifically, out of the n items, one can potentially cluster the products into different clusters (also called blocks) that share similar features. In practice, one can estimate the cross elasticities between prices and very often, a large portion are not statistically significant. Alternatively, one can first run a clustering algorithm (e.g., K -means) to partition the n items into K clusters (the value of K depends on the setting) and then, for each cluster, one can estimate the demand of each item while assuming there is no cross price effects between two different clusters. To identify the clusters, one can use data attributes such as price range (e.g., low/medium/high), brand, functionality features (e.g., color, type of product, category, flavor), location of the item on the shelf (for brick-and-mortar stores), and velocity (i.e., how fast the product has been selling in the past). In what

follows, we assume that the n items can be clustered in K blocks. We then consider that only the items in the same block affect each other, that is, the demand of item i depends only on prices of the items in the same block. Using this assumption, one can take advantage of this structure to improve the time complexity of our methods by solving for each block separately. More precisely, when $\mathcal{Q}_r = \mathcal{Q}_p$ and assuming that each cluster is composed of n/K items (for ease of exposition), the time complexities are given by:

- Naive model: $O(KT|\mathcal{Q}_p|^{n(m+1)/K})$.
- Model with n reference prices: $O(KT|\mathcal{Q}_p|^{1+2n/K})$.
- Model with a single reference price: $O(KT|\mathcal{Q}_p|^{2+n/K})$.

A more concrete comparison of the runtimes for the three methods is presented in section 6.5 using a realistic instance with 100 items. As we will see, the model with a single reference price solves in minutes even for instances with a large number of items. We next discuss how to adapt our approach to incorporate global business rules.

5.4. Business Constraints with Blocks

Motivated by practical retail settings, we consider global pricing rules that restrict the permissible price changes. For example, the retailer may have a restriction on the total number of price changes in the category. This rule may be imposed by a store manager who is concerned about preserving its store image. A second common restriction is called price-ordering constraints that dictate the price relationship of two items. Finally, we also consider exclusive deals offered by brand manufacturers.

We first illustrate the combinatorial blow up that occurs when trying to satisfy global business constraints. Consider two blocks of items with no cross-item effects between the blocks. For instance, one could consider two sets of items: dark roast coffee (block 1) and decaf coffee (block 2). Suppose that by optimizing the prices independently for each block, the profits are maximized in block 1 by setting 6 price changes, whereas block 2 requires 4 price changes. However, consider the global constraint in which the store enforces a limit of 8 price changes for all coffee items. We then need to decide which price changes to remove from each block so that the overall profits are maximized while satisfying the global limit of 8 price changes. Note that the number of options increases exponentially with the number of blocks. We next show how to adapt our approach to avoid such a combinatorial blow up and to efficiently incorporate global constraints.

Case 1. Limiting the total number of price changes. Our goal is to restrict the total number of price changes to

be at most L_{total} , overall K blocks. First, we compute the function $Y(\cdot) : \{1, \dots, K\} \times \{0, \dots, L_{total}\} \rightarrow \mathbb{R}$ that calculates the maximum profit achievable in block B_i by using at most j price changes. This function can be computed using the methods from section 5 (a single run of the algorithm is required for each block B_i and each value of j). Let $x_{(i,j)}$ be binary variables for $i \in \{1, \dots, K\}$ and $j \in \{0, \dots, L_{total}\}$ such that $x_{(i,j)} = 1$ if and only if a price assignment (i.e., the price of each item at each time period) with at most j price changes is selected for block B_i (and $x_{(i,j)} = 0$, otherwise). To find the optimal prices for all blocks that satisfy the global constraint of at most L_{total} price changes, one can solve the following optimization problem:

$$\max \sum_{i=1}^K \sum_{j=0}^{L_{total}} Y_i(j)x_{(i,j)}$$

$$\sum_{j=0}^{L_{total}} x_{(i,j)} = 1 \forall i \in \{1, \dots, K\} \tag{19}$$

$$\sum_{i=1}^K \sum_{j=0}^{L_{total}} jx_{(i,j)} \leq L_{total} \tag{20}$$

$$x_{(i,j)} \in \{0, 1\} \forall i \in \{1, \dots, K\}, j \in \{1, \dots, L_{total}\}.$$

Here, the decision variables are $x_{(i,j)}$. The variables $Y_i(j)$ are computed beforehand. Constraints (19) ensure that for any block B_i , a specific number of price changes is selected, whereas Constraints (20) ensure that the total number of price changes across all blocks does not exceed L_{total} . This is an instance of the well-known multiple-choice knapsack problem which is NP-hard (see, e.g., Pisinger 1995). Nevertheless, it can be solved using dynamic programming (Dudziński and Walukiewicz 1987), with running time $O(KL_{total}^2)$ —allowing us to handle the restriction on the total number of price changes efficiently.

Case 2. Price-ordering constraints. Retailers may need to satisfy price orderings on certain sets of items. A concrete example is the same brand with two different formats. In this case, the price of the smaller format needs to be cheaper than the larger format, at all times. Consider items i and j (possibly in different blocks) such that the price of item i is required to be lower than the price of j at all times, that is, we want to ensure that

$$\max_{t=1, \dots, T} p_t^i \leq \min_{t=1, \dots, T} p_t^j. \tag{21}$$

Let Q_i and Q_j be the price ladders of items i and j , respectively. To impose Equation (21), we compute the maximum achievable profit for each of the two blocks (and the corresponding price vectors) such that $p_t^i \leq v$ and $p_t^j \geq v$ for all $t = 1, \dots, T$ and for each value $v \in Q_i \cup Q_j$. One can do so by simply deleting

the nodes that violate these inequalities. Then, one can obtain the optimal prices that satisfy Equation (21) by selecting the value of v that maximizes the total profit. In the worst case, we require solving the profit maximization problem $2|Q_i \cup Q_j|$ times.

Case 3. Exclusivity constraints. Trade funds dictated by manufacturers may impose an “exclusivity deal” that prohibits retailers from decreasing prices of competing items at the same time. For example, a manufacturer may offer a deal to the retailer that entails promoting (i.e., decreasing prices) some of the items in the set K_1 at a given time t (e.g., during a national holiday). However, if the retailer decides to promote some of these items, the exclusivity constraint prohibits the retailer to promote other competing items, say in the set K_2 , at time t . For each block B_i , we compute the maximum profit $\pi_1(i)$ where no item in K_1 is promoted at time t (by simply deleting the nodes that promote items in K_1), as well as the maximum profit $\pi_2(i)$ where no item in K_2 is promoted at time t . Finally, the maximum achievable profit can be obtained by $\max\{\sum_i \pi_1(i), \sum_i \pi_2(i)\}$. In the worst case, we require solving the profit maximization problem twice for each block.

6. Computational Experiments

In this section, we present computational experiments to validate the efficiency and scalability of our methods. First, we test the exact dynamic program for a single item using demand models calibrated from data. Second, we apply our graphical method to the reference price model from section 4. Third, we evaluate the three proposed approaches for multiple items, as discussed in section 5. The tests presented in this section reflect realistic instances faced by supermarket retailers.

6.1. Supermarket Data

Via a collaboration with Oracle Retail, we received a large data set from several categories of items in supermarkets. Specifically, we use aggregate weekly sales data for several brands of coffee in 2009–2011. We calibrate the demand models and observe that our estimated demand yield a good out-of-sample forecast accuracy. In particular, the out-of-sample R^2 is between 0.85 and 0.96. We use $c_t = 0.4; \forall t, T = 35, q^0 = 1$, and $q^Q = 0.4$. More details on the data and demand estimation can be found in Cohen et al. (2017). All the tests were run using an Intel Xeon @ 3.10GHz CPU with 125 GB RAM, and the dynamic program was solved using Julia and Gurobi 6.0.0.

6.2. Single Item

As shown in Table 1, solving the dynamic program based on our graphical representation scales linearly with T . However, the runtime is exponential with m .

Our goal is to test the runtime with respect to the different problem parameters using realistic instances to understand the limitations of our approach. As discussed in section 3.2, we expect to observe asymptotic runtimes of $O(T|Q_p|^m)$. In this section, we consider the following log-log demand function estimated from data:

$$d_t = a_t \exp \left\{ -3.277 \log(p_t) + 0.518 \log(p_{t-1}) + 0.465 \log(p_{t-2}) + 0.2325 \log(p_{t-3}) + 0.115 \log(p_{t-4}) \right\},$$

where the coefficients $a_t, t = 1, \dots, 35$ represent the multiplicative seasonality effects and were estimated to be between 759.4 and 975.7. In Figure 4, we plot the runtime as a function of the price ladder size (for $T = 35$) and as a function of the number of periods (using 8 price points). We consider three values for the memory parameter ($m = 2, 3, 4$) and impose a timeout of 2,000 seconds. For several item categories in supermarkets, m is between 0 and 4. For instance, for several coffee items, the memory parameter was found to be equal to 2. The planning horizon ranges between 10 and 52 weeks, and the number of price points varies between 2 and 20.

One can see that when $m = 2$, our solution approach solves the problem in less than a second. This allows the retailer to perform several “what-if” scenarios (sensitivity analysis tests) by varying the demand parameters to obtain a robust solution. In addition, for items without cross-item effects, one can solve the problem for thousands of different items in a few seconds. However, when the memory parameter becomes large (e.g., $m = 4$), this is not the case anymore, as our method can take several minutes to solve a single instance. For example, with 12 prices and $m = 4$, it takes more than 20 minutes. In such a

case, one can use our reference price approximation (see section 4) that allows us to compute a near-optimal solution in seconds, as shown in section 6.3.

We next study the effect of incorporating business rules on the runtime of the dynamic program. As discussed in section 2.1, two of the main business rules are limiting the number of price changes (denoted by L) and imposing a separating period between successive price changes (denoted by S). We test these two cases in Figure 5. We fix $|Q_p| = 8, T = 35$, and vary L and S . Adding the L constraint to the formulation increases the number of possible states multiplicatively by L . Figure 5 is consistent with this analytical result (the y-axis is in logarithmic scale). Incorporating the no-touch constraint and varying the value of S have a relatively low effect on the runtime, which is consistent with the asymptotic runtime of $O(T^2|Q_p|^{\frac{m}{S}})$. We conclude that for large memory parameters (i.e., $m \geq 4$), solving the dynamic program is not a viable option. Fortunately, in practice, a significant number of items in supermarkets admit a small memory parameter. However, when m is large, one can use the approximation based on the reference price, as we discuss next.

6.3. Discrete Reference Price Model

While the exact dynamic program provides an efficient method when m is low, instances with $m \geq 4$ can take several minutes (or even hours) to run (see Figures 4 and 5). We address this issue by considering the discrete reference price model introduced in section 4.1. We consider the following log-log reference price-demand function:

$$d_t = a_t \exp \left\{ -3.3p_t + 0.52r_t \right\},$$

Figure 4 Runtimes as a Function of the Price Ladder Size (left) and the Number of Time Periods (right) [Color figure can be viewed at wileyonline library.com]

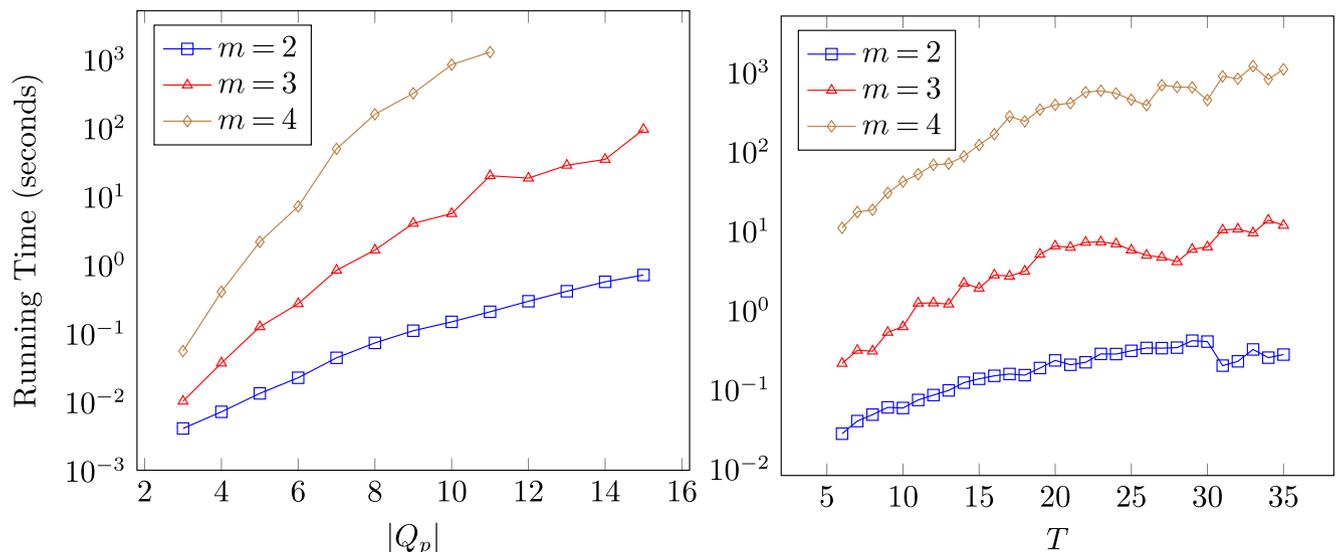
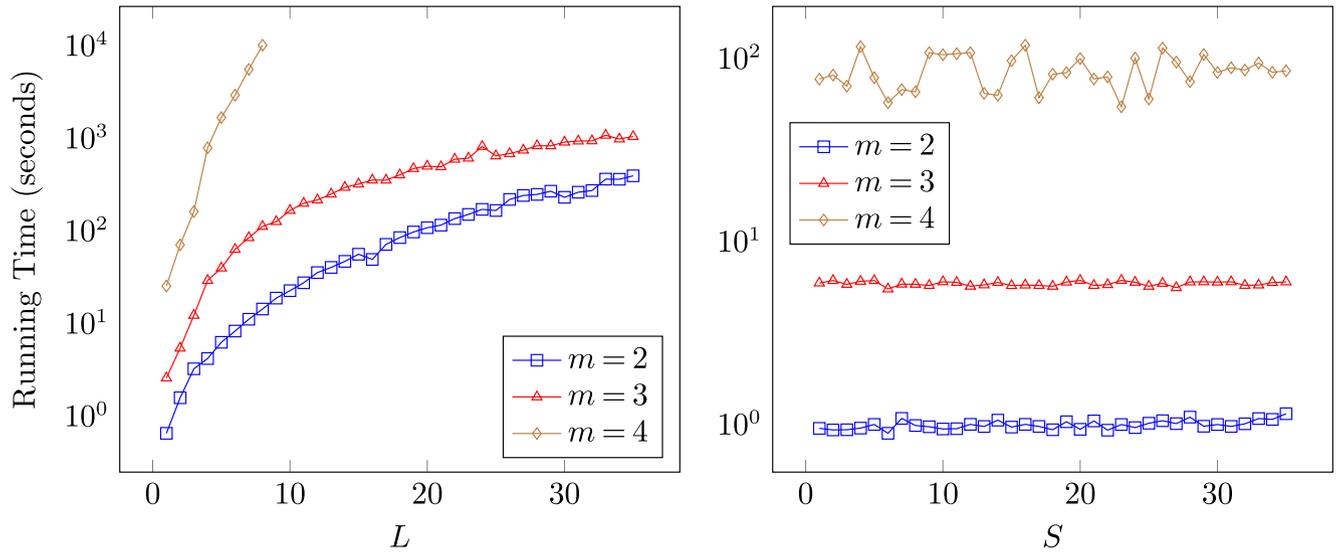


Figure 5 Runtimes as a Function of L (left) and S (right) [Color figure can be viewed at wileyonlinelibrary.com]



where the coefficients $a_t, t = 1, \dots, 35$ represent the multiplicative seasonality effects and are between 777.2 and 930.5. The reference price follows: $r_t = 0.6p_{t-1} + 0.4r_{t-1}$ and $r_0 = q^0 = 1$. In Figure 6, we investigate the runtimes of the dynamic program for the discrete reference price model by varying the reference price ladder precision ϵ , the size of the price ladder, and the number of periods. One can see that the discrete reference price model can be solved efficiently even with a large number of prices and a granular reference price ladder ($\epsilon = 0.025$). More precisely, all the instances we tested could be solved within 0.1 second.

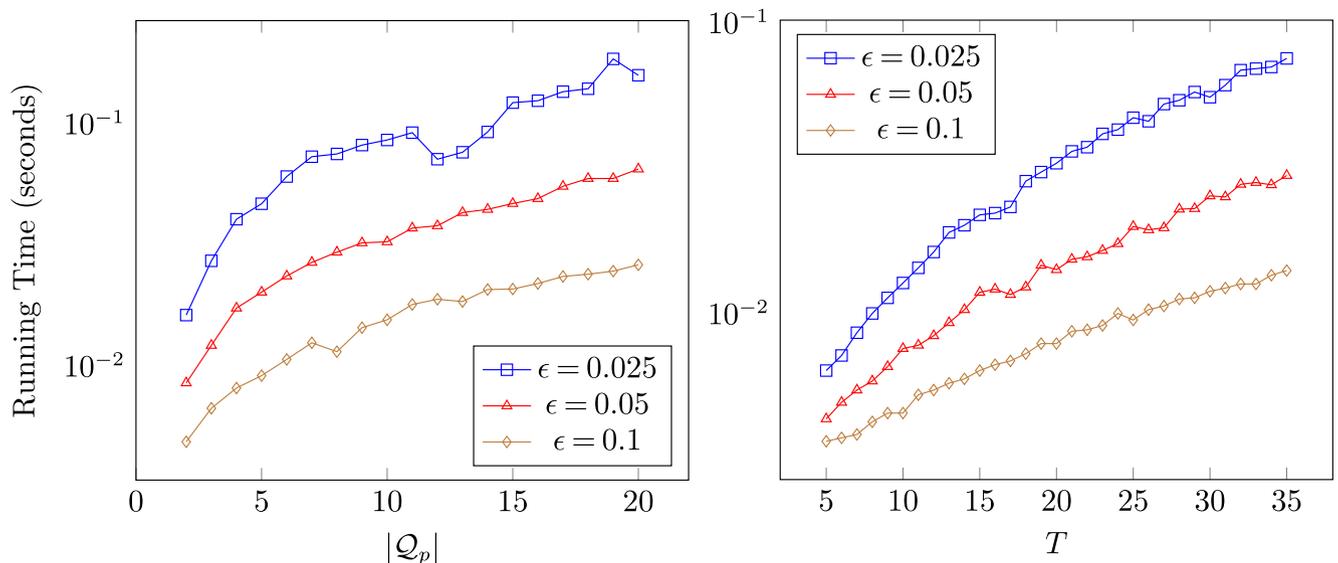
We next test the quality of the approximation presented in section 4.2. Our goal is to approximate a

demand function with a large memory parameter by a discrete reference price model. We consider 100 randomly generated instances of linear demand models as in Equation (13). For each instance, we assume $m = T = 10$ and $|\mathcal{Q}_p| = 2$ (i.e., $q^0 = 1$ and $q^1 = 0.7$) and the following linear demand:

$$d_t = a_t - b_0p_t + b_1p_{t-1} + b_2p_{t-2} + \dots + b_{10}p_{t-10},$$

where a_t and b_0 are randomly drawn from a uniform distribution on [3000,5000] and [2000,4000], respectively. The vector of parameters b_1, \dots, b_{10} is also randomly generated from a uniform distribution on $[0, 200]^{10}$. We then order the random vector such that $b_1 \geq b_2 \geq \dots \geq b_{10}$.

Figure 6 Runtimes of the Dynamic Program with Reference Price When Varying ϵ as a Function of the Price Ladder Size (left) and the Number of Time Periods (right) [Color figure can be viewed at wileyonlinelibrary.com]



For each instance, we obtain a linear demand function with a large memory parameter. We then follow the procedure described in section 4.2 and fit a discrete reference price model to approximate the true demand function by finding the value of $\hat{\theta}_{LS}$ for each instance. We then solve the unconstrained dynamic program and compute the optimal prices for the discrete reference price model with $\varepsilon = 0.001$. For each instance, we compare the total profit generated by these prices relative to the optimal prices obtained by solving the exact (non-tractable) formulation under the true demand. For most of the instances we considered, the approximated model based on the discrete reference price model yields a near-optimal solution. In particular, the minimum is 77.6%, the 25th percentile is 97.7%, the median is 99.4%, and both the 75th percentile and the maximum are 100% relative to the optimal profit. Consequently, for demand models with a large memory parameter, the approximation developed in section 4.2 allows us to solve the problem in milliseconds while finding a near-optimal solution.

6.4. Reference Price Approximation

We next compare models with $m = 2$, $m = 3$, and $m = 4$ when using the reference price approximation. Specifically, we proceed as follows:

1. We generate several demand models with $m = 2$. For consistency, we consider the same functional form as in section 6.2. Specifically, we use the same parameters for the seasonality coefficients a_t and vary only the parameters that multiply the prices (i.e., b_0, b_1, b_2). As before, we use a cost $c_t = 0.4$ for all t and consider 8 price points between 0.4 and 1. We consider three distribution families to generate the price coefficients (Uniform, Beta, and Lognormal). For example, we use a Uniform distribution between 0 and 5—that is, $b_0, b_1, b_2 \sim U[0, 5]$ —and we then sort the parameters so that $b_0 > b_1 > b_2$.
2. We repeat the same procedure for demand models with $m = 3$ and $m = 4$.
3. For each demand model, we approximate it using the reference price model based on $\hat{\theta}_{LS}$.

4. We find the optimal prices using the reference price approximation. We then evaluate the total profit obtained under these prices (using the true demand model).
5. We find the optimal prices and profits for the true model (by solving the dynamic program) and compute the optimal total profit.
6. We compute the relative ratio of the profits in Steps 4 and 5 above. We repeat this procedure across 10,000 independent trials and plot the relative ratio (distribution and summary statistics). We also consider several values of the selling season ($T = 10, 15, 20$).⁶

As mentioned, we consider three distribution families to generate the price coefficients. Each distribution has positive values and a mean equal to 2.5. This corresponds to reasonable parameter values based on our estimation results. We also made sure that our generated demand models give rise to variation in prices (i.e., we avoid the case where it is optimal to have no price changes). The results are reported in Figures A1–A3 in Appendix B. In each plot, the bars represent (from top to bottom): the maximum, 75th percentile, median, 25th percentile, and minimum. The “+” signs underneath the minimum bar are considered as outliers. We also summarize the median values in Figure 7.

As we can see from the results, the performance of the reference price approximation is high across all tested settings (i.e., different distributions to generate the parameters and different values of T). Specifically, the median ratio relative to the optimal profits always exceeds 90%. As expected, the exact performance depends on the specific instance (i.e., the combination of parameters in the demand model). Interestingly, the performance is not highly affected by the value of T , the value of m , or the distribution of the parameters (holding the mean constant). More precisely, we do not observe a clear monotonic pattern with respect to the value of m . Instead, we find that the performance remains high for $m = 2, 3, 4$. We also consider the case with a higher memory parameter and found similar results. For example, the results for $T = 15$ and $m = 8$ when the demand-price coefficients are generated

Figure 7 Performance of the Reference Price Approximation for the Different Settings (median values across 10,000 independent trials)

	T = 10			T = 15			T = 20	
	m = 2	m = 3	m = 4	m = 2	m = 3	m = 4	m = 2	m = 3
Unif(0,5)	93.04%	96.61%	97.65%	96.00%	97.96%	98.48%	93.90%	97.76%
10*Beta(1,3)	90.71%	96.07%	93.24%	92.90%	96.72%	93.58%	90.31%	95.75%
(2.5/exp(5.5))*Lognormal(5,1)	95.74%	98.14%	94.54%	96.23%	97.51%	94.52%	94.72%	97.45%

using the same Lognormal distribution as before are reported in Figure A4 (using three price points in the price ladder and 10,000 iterations). As we can see, the performance remains high (the median of the profit ratio is equal to 95.34%).

Finally, we consider an instance where the demand-price coefficients decrease by a constant factor. Such a structural assumption helps capturing the fact that further past prices have a smaller impact on current demand. Specifically, we randomly generate the coefficients b_0 and b_1 (using a Uniform distribution between 1 and 6). We also generate the parameter θ using a Uniform distribution between 0 and 1. We then impose: $b_2 = \theta b_1$, $b_3 = \theta b_2$, and so on. We call this model the *truncated reference price model*. We consider $T = 10$ and vary m between 2 and 5 (similar results were observed for different values of T and m). The results are presented in Figure A5. As we can see, the reference price approximation yields a good performance throughout (in this case, the performance seems to slightly decrease with m). However, for all memory values we tested, we obtained that the median performance was higher than 91% relative to the optimal profits.

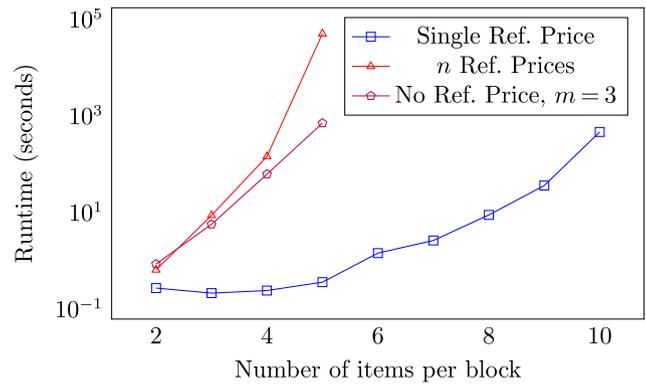
6.5. Multiple Items

We now consider the setting with multiple items and test the three methods developed in section 5. Our goal is to compare the runtimes from Table 2. We consider an instance with $n = 100$ items and $T = 10$ periods. Our objective is to maximize the total profit generated by all items during the selling season. For simplicity, we consider that all items are identical (same demand functions and cost value), impose a flat seasonality $a_t = 50 \forall t$, and that there are two prices ($q^0 = 1$ and $q^1 = 0.7$). We then compare the runtimes of the three solution approaches: (i) the dynamic program via the naive model, (ii) the model with n discrete reference prices from section 5.1, and (iii) the model with a single discrete virtual reference price from section 5.2. Motivated by the discussion in section 5.3, we cluster the 100 items into different blocks and vary the number of items per block between 2 and 10. We run the three approaches for each block and record the corresponding runtime to solve the problem for all 100 items. We set $\varepsilon = 0.1$ and impose a timeout of 12 hours.

In this section, we consider the following linear demand functions for each item $i = 1, \dots, 100$:

- Naive model: $d_t^i = 50 - 15p_t^i + 10p_{t-1}^i + 5p_{t-2}^i + \sum_{j \neq i} 5p_t^j$.
- Model with n reference prices: $d_t^i = 50 - 15p_t^i + 10r_t^i + \sum_{j \neq i} 5p_t^j$.

Figure 8 Runtimes for an Instance with 100 Items and 10 Periods
 [Color figure can be viewed at wileyonlinelibrary.com]



- Model with a single virtual reference price: $d_t^i = 50 - 15p_t^i + 10r_t^i + \sum_{j \neq i} 5p_t^j$.

The reference prices follow: $r_t^i = \text{round}[0.6p_{t-1}^i + 0.4r_{t-1}^i]$ with $r_0^i = q^0 = 1$ and $r_t^i = \text{round}[0.6 \frac{1}{100} \sum_{i=1}^{100} p_{t-1}^i + 0.4r_{t-1}^i]$ with $r_0^i = q^0 = 1$ (we use $\varepsilon = 0.1$). Note that one can update the virtual reference price by either the average or the maximum of the prices p_{t-1}^i , as all items are identical. The results are presented in Figure 8. For practical considerations, our target is to solve the problem for 100 items in a few minutes. Thus, one can see that the only viable method is the model with a single virtual reference price. This approach scales significantly better than the two other methods, and thus allows us to solve the problem for larger blocks of items in reasonable time frames. Note that in practice, one could achieve better runtimes by optimizing each block in parallel, which would speed up the runtime multiplicatively by the number of blocks.

7. Conclusion

In this study, we study the multi-item multi-period pricing problem faced by supermarket retailers. Typically, this problem involves a non-linear demand model that depends on current and past prices and the presence of business rules. We introduce a graphical representation that translates the profit maximization into solving a maximum weighted path problem on a layered graph. We further prove that the problem is NP-hard by showing an approximation preserving reduction from weighted Max-3-SAT problem. We also develop a dynamic programming solution method with a runtime of $O(T|Q_p|^{m+1})$, where T is the number of periods, m the memory parameter (number of past prices that affect current demand), and Q_p is the price ladder.

When m is large, this approach is not practical, as it may take minutes (or hours) to solve a single instance. Several categories of products can have a large m .

Typically, if the product is non-perishable (e.g., toiletries and laundry detergent), m can be between 4 and 8. In addition, even if we consider a memory of 2 with 50 products, solving the problem to optimality can take several hours (or days). This motivates us to develop an approximation solution approach. We first consider the setting under a reference price model. We propose to use the discrete reference price model that restricts reference prices to lie in a discrete set under the premise that customers do not form a reference price with infinite precision. Under this model, we show how to solve the problem efficiently. We then consider several demand functions (linear, log-log, and log-linear) with a large m . We approximate these demand functions using the discrete reference price model—allowing us to solve instances with a large memory in milliseconds, while having a performance guarantee.

We then consider the problem for multiple items and develop two solution approaches inspired by the discrete reference price model. We assume that consumers form a reference price either for each product separately or a joint virtual reference price for the entire category. To increase the tractability of our approach, we introduce the notion of blocks and organize items into small clusters so that cross-item interactions across blocks are negligible. Although this reduces the runtime of the profit maximization problem for each block, it becomes more challenging to impose global pricing constraints across blocks. By borrowing ideas from combinatorial optimization, we limit the number of price changes across blocks by solving a multi-choice knapsack. We finally apply our solution approach using demand models calibrated with supermarket data and show that we can solve realistic instances efficiently.

Acknowledgment

The authors thank the department editor (Kalyan Singhal), the associate editor, and the anonymous referees for their insightful comments which have helped improve this study. We also thank Oracle Corporation for their generous support through an ERO grant that made this work possible. In addition, we acknowledge Kiran Panchamgam who gave us valuable insights on this project. We thank Kiran Gite (from Georgia Tech) who helped us conduct the computations presented in section 6.4. This work was done when the second author was a doctoral student at MIT and acknowledge the support of NSF Grant Nos. CCF-1115849, OR-1029603, ONR grant Nos. N00014-12-1-0999, N00014-15-1-2083, and N00014-14-1-0072. The fourth author would like to acknowledge the generous support of NSF grants CMMI-1162034 and CMMI-1563343.

Appendix A: Log-Log and Log-Linear Demand Models

In this section, we study the log-log and log-linear models. First, we present the analysis for the log-reference price model that will be useful in finding a tractable algorithm for the log-log model in Equation (A6). In this case, the reference prices are maintained and updated in the log space:

$$\log r_t = (1 - \theta) \log p_{t-1} + \theta \log r_{t-1}, \quad (\text{A1})$$

where $0 \leq \theta < 1$ is a parameter that can be used to fit the underlying (true) demand model. Note that this is similar to the traditional reference price model in Equation (4) but the update is in the log space. The log-log reference price and the log-linear reference price-demand models are given by:

$$\log d_t(p_t, r_t) = f_t(p_t) + g(\log p_t - \log r_t), \quad (\text{A2})$$

$$\log d_t(p_t, r_t) = f_t(p_t) + g(p_t - r_t). \quad (\text{A3})$$

We do not impose any assumption on $f_t(\cdot)$ and assume that $g(\cdot)$ is G -Lipschitz.

We next consider the discrete log-reference price model by discretizing the log space of reference prices so that the reference price at time t is given by (the exact way of rounding does not affect any of our results):

$$\log \hat{r}_t = \text{round} \left[(1 - \theta) \log p_{t-1} + \theta \log r_{t-1} \right]. \quad (\text{A4})$$

More precisely, we round to the nearest element in the set $\mathcal{Q}_{\log} = \{r^0 > r^1 > \dots > r^m > \dots > r^N\}$, where $\log r^i = \log r^{i+1} + \epsilon$ for all $i=0, \dots, N-1$. Using this rounding procedure, we show that the propagated difference between the continuous and discrete reference prices at time t varies linearly with ϵ and with the continuous reference price.

PROPOSITION 5. Consider the continuous log-reference price model from Equation (A1) and the discrete log-reference price model with precision $\epsilon > 0$ as in Equation (A4). Then, the difference in the reference prices at time t is bounded by:

$$|\hat{r}_t - r_t| \leq \frac{1 - \theta^{t-1}}{1 - \theta} r_t \epsilon, \quad (\text{A5})$$

where r_t and \hat{r}_t denote the continuous and discrete reference prices at time t , respectively.

PROOF. For the first period, we have: $\log \hat{r}_1 = \text{round} \left[(1 - \theta) \log p_0 + \theta \log r_0 \right]$. Therefore, we obtain: $\log \hat{r}_1 = (1 - \theta) \log p_0 + \theta \log r_0 \pm \epsilon$ which implies $\hat{r}_1 = p_0^{(1-\theta)} r_0^\theta e^{\pm \epsilon} \approx p_0^{(1-\theta)} r_0^\theta (1 \pm \epsilon)$, where we used the approximation $e^x \approx 1 + x$ for $x \in (0, 1), x \ll 1$. We then have: $r_1(1 - \epsilon) \leq \hat{r}_1 \leq r_1(1 + \epsilon)$.

We next proceed by induction on $t > 1$. We assume that for $t \leq k$, $\hat{r}_k = r_k(1 \pm \epsilon)^{\sum_{u=0}^{k-1} \theta^u}$ and show the claim for $t = k + 1$. We have: $\log \hat{r}_{k+1} = \text{round} \left[(1 - \theta) \log p_k + \theta \log \hat{r}_k \right]$. Then, we obtain:

$$\begin{aligned} \hat{r}_{k+1} &= p_k^{1-\theta} \hat{r}_k^\theta e^{\pm \epsilon} = p_k^{1-\theta} r_k^\theta (1 \pm \epsilon)^{\sum_{u=0}^{k-1} \theta^{u+1}} e^{\pm \epsilon} \\ &\approx p_k^{1-\theta} r_k^\theta (1 \pm \epsilon)^{\sum_{u=0}^k \theta^u} = r_{k+1} (1 \pm \epsilon)^{\sum_{u=0}^k \theta^u}. \end{aligned}$$

Therefore, $r_{k+1}(1 - \sum_{u=0}^k \theta^u \epsilon) \leq \hat{r}_{k+1} \leq r_{k+1}(1 + \sum_{u=0}^k \theta^u \epsilon)$ concluding the proof.

We next quantify the difference in demand and profit.

COROLLARY 2. Consider the log-log demand model from Equation (A2) and the discrete log-reference price model with precision $\epsilon > 0$. Then, the demand value at time t and the difference in the total profit are bounded by:

$$\begin{aligned} \hat{d}_t &= d_t \exp(g(\log \frac{p_t}{r_t}) - g(\log \frac{p_t}{\hat{r}_t})) \leq d_t \exp(G \log \frac{\hat{r}_t}{r_t}) \\ &\approx d_t \exp(G \log(1 \pm \sum_{u=0}^t \theta^u \epsilon)) \leq d_t \exp(G \epsilon \sum_{u=0}^t \theta^u), \\ |\hat{\Pi} - \Pi| &\leq (\max_t d_t) T (q^0 - c_{\min}) G \epsilon / (1 - \theta), \end{aligned}$$

where \hat{d}_t and $\hat{\Pi}$ denote the demand and profit at time t , respectively, using the discrete log-reference price model. Here, c_{\min} denotes the minimal cost value, $c_{\min} = \min_t c_t$, and $\epsilon \ll 1$.

Finally, we extend the analysis for the log-linear reference price model in Equation (A3). In this case, we use the traditional reference price model as in section 4.

COROLLARY 3. Consider the log-linear demand model from Equation (A3) and the discrete reference price model with precision $\epsilon > 0$. Then, the demand value at time t and the difference in the total profit are bounded by:

$$\begin{aligned} \hat{d}_t &= d_t \exp(g(p_t - r_t) - g(p_t - \hat{r}_t)) \leq d_t \exp(G(\hat{r}_t - r_t)) \\ &\leq d_t \exp(G \epsilon (1 + \sum_{u=0}^t \theta^u)), \quad |\hat{\Pi} - \Pi| \\ &\leq (\max_t d_t) T (q^0 - c_{\min}) G \epsilon \frac{2 - \theta}{1 - \theta}, \end{aligned}$$

where \hat{d}_t and $\hat{\Pi}$ denote the demand and profit at time t , respectively, using the discrete reference price model.

As a result, we obtain constant gap guarantees for the optimal profit. Observe that the maximum possible demand over all time periods, $\max_t d_t$, can be easily obtained from the context.

Approximating the log-log and log-linear demand models. As in section 4.2, for ease of exposition, we present our analysis on the approximation gap relative to the continuous reference price model (recall that the discrete model approaches the continuous model when ϵ tends to 0). We next extend the results of section 4.2 to the log-log and log-linear demand models, given by:

$$\log d_t(\mathbf{p}_t) = f_t(p_t) + \beta^1 \log p_{t-1} + \dots + \beta^T \log p_{t-T}, \tag{A6}$$

$$\log d_t(\mathbf{p}_t) = f_t(p_t) + \beta^1 p_{t-1} + \dots + \beta^T p_{t-T}. \tag{A7}$$

This type of models are popular in retail applications such as supermarkets. The parameters β^1, \dots, β^T and the functions $f_t(\cdot)$ can be estimated from data. As in section 4, we assume that the gain parameters are non-negative and non-increasing: $\beta^1 \geq \dots \geq \beta^T \geq 0$.

Using the continuous log-linear reference price and the log-log reference price models, we approximate Equations (A6) and (A7) as follows:

$$\begin{aligned} \log \tilde{d}_t(p_t, \tilde{r}_t) &= f_t(p_t) + g(\log p_t - \log \tilde{r}_t) \\ &= f_t(p_t) + \tilde{\phi} \sum_{k=1}^T (1 - \tilde{\theta}) \tilde{\theta}^{k-1} \log p_{t-k}, \end{aligned} \tag{A8}$$

$$\begin{aligned} \log \tilde{d}_t(p_t, \tilde{r}_t) &= f_t(p_t) + g(p_t - \tilde{r}_t) \\ &= f_t(p_t) + \tilde{\phi} \sum_{k=1}^T (1 - \tilde{\theta}) \tilde{\theta}^{k-1} p_{t-k}. \end{aligned} \tag{A9}$$

Following the same procedure as in section 4.2, we impose $\tilde{\phi}(1 - \tilde{\theta}) = \beta^1$ to match the coefficient of p_{t-1} . Then, $\tilde{\theta}$ can be computed as the minimum, maximum, or least-squares fit. Thus, we obtain the same result as in Proposition 4 for the log-log and log-linear demand models.

Appendix B: Testing the Reference Price Approximation

Figure A1 Performance of the Reference Price Approximation (price coefficients are distributed $U(0,5)$, $T = 10,15,20$, and $m = 2,3,4$) [Colour figure can be viewed at wileyonlinelibrary.com]

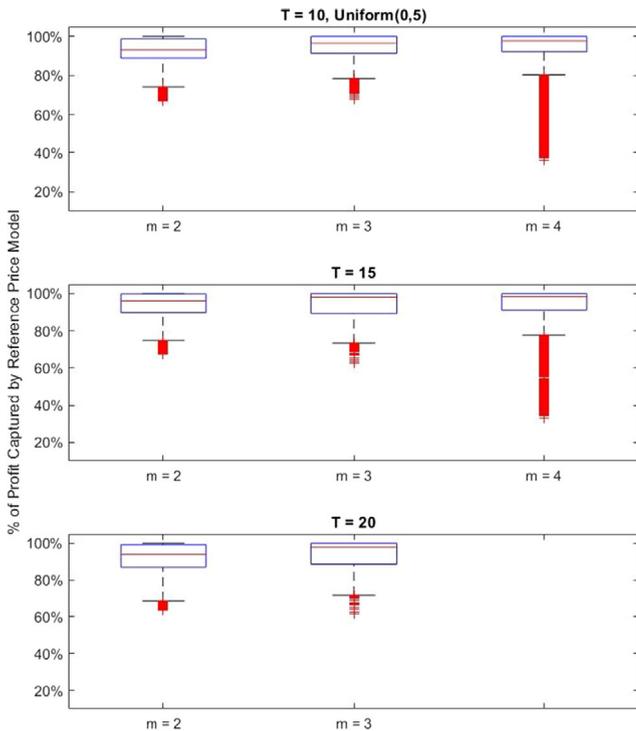


Figure A2 Performance of the Reference Price Approximation (price coefficients are distributed $10\text{Beta}(1,3)$, $T = 10,15,20$, and $m = 2,3,4$) [Colour figure can be viewed at wileyonlinelibrary.com]

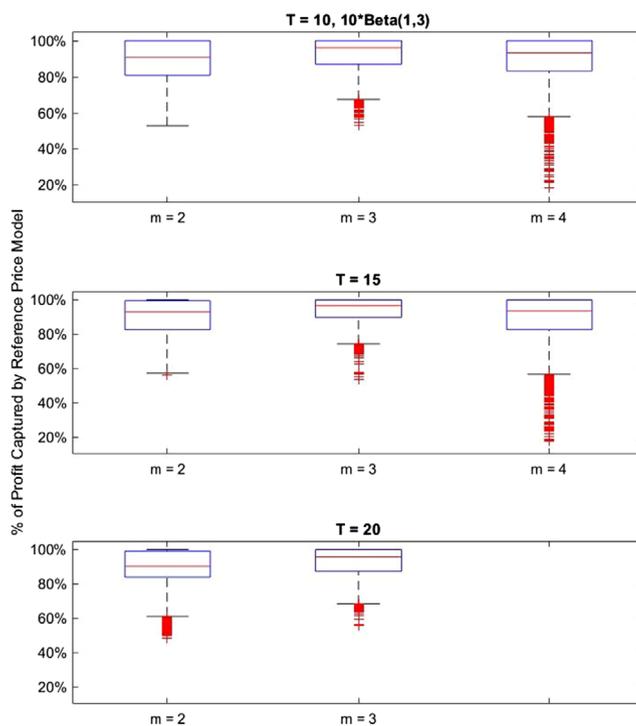


Figure A3 Performance of the Reference Price Approximation (price coefficients are distributed $(2.5/\exp(5.5))\text{Lognormal}(5,1)$, $T = 10,15,20$, and $m = 2,3,4$) [Colour figure can be viewed at wileyonlinelibrary.com]

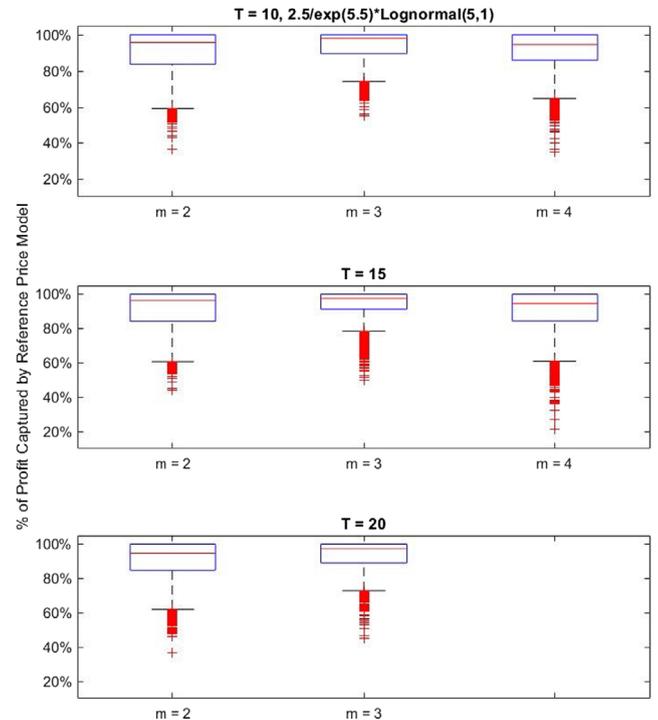


Figure A4 Performance of the Reference Price Approximation for a Truncated Reference Price-Demand Model ($T = 15$ and $m = 8$) [Colour figure can be viewed at wileyonlinelibrary.com]

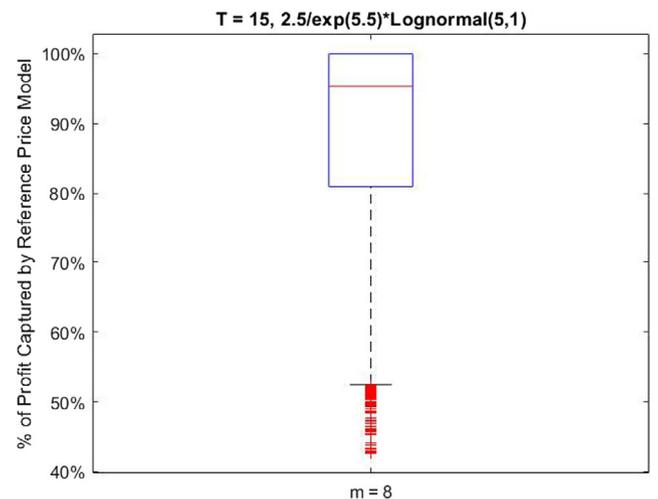
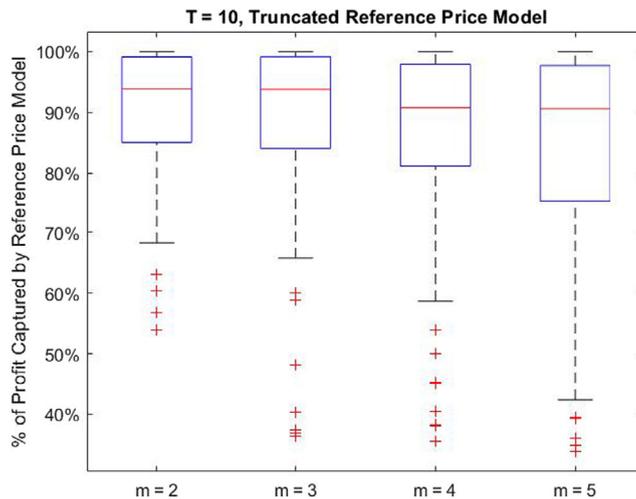


Figure A5 Performance of the Reference Price Approximation for a Truncated Reference Price-Demand Model ($T = 10$ and $m = 2,3,4,5$)
[Colour figure can be viewed at wileyonlinelibrary.com]



Notes

¹This study applies for both price changes and promotions (a price change at time t is defined such that $p_t \neq p_{t-1}$, whereas a promotion is a temporary reduction from the regular price q^0). To avoid confusion, we only present the case of price changes.

²We are ignoring the dependence in the memory parameter (m) in the order notation.

³We say that $f(T) = \Omega(g(T))$, when there exists a constant c such that $f(n) \geq cg(n)$ for all sufficiently large n .

⁴The exact way of rounding does not affect any of our results, so we simply round to the closest element in \mathcal{Q} .

⁵For ease of exposition, we present our analysis of the approximation gap relative to the continuous reference price model (recall from Proposition 2 that the discrete model approaches the continuous model when ε tends to 0). In other words, Equation (14) is obtained using continuous reference prices (and not discretized).

⁶We did not consider the combination $T = 20$ and $m = 4$ as it requires several days to run this case, without adding further insights.

References

- Ahuja, R. K., T. L. Magnanti, J. B. Orlin. 1988. Network flows. Technical report, DTIC Document.
- Balcan, M. F., A. Blum, Y. Mansour. 2008. Item pricing for revenue maximization. Proceedings of the 9th ACM conference on Electronic commerce, pp. 50–59.
- Blattberg, R. C., S. A. Neslin. 1990. *Sales Promotion: Concepts, Methods, and Strategies*. Prentice Hall, Upper Saddle River, NJ.
- Blattberg, R. C., R. Briesch, E. J. Fox. 1995. How promotions work. *Market. Sci.* **14**(3): G122–G132.
- den Boer, A. V. 2015. Dynamic pricing and learning: Historical origins, current research, and new directions. *Surv. Oper. Res. Manag. Sci.* **20**(1): 1–18.
- Chaib-draa, B., J. Müller. 2006. *Multiagent Based Supply Chain Management*, Vol. 28. Springer Science & Business Media, Berlin.
- Chakraborty, T., Z. Huang, S. Khanna. 2013. Dynamic and nonuniform pricing strategies for revenue maximization. *SIAM J. Comput.* **42**(6): 2424–2451.
- Chen, X., P. Hu, Z. Hu. 2016. Efficient algorithms for the dynamic pricing problem with reference price effect. *Management Sci.* **63**(12): 4389–4408.
- Cohen, M. C., N. H. Z. Leung, K. Panchamgam, G. Perakis, A. Smith. 2017. The impact of linear optimization on promotion planning. *Oper. Res.* **65**(2): 446–468.
- Dockner, E. J., S. Jorgensen, N. Van Long, G. Sorger. 2000. *Differential Games in Economics and Management Science*. Cambridge University Press, Cambridge.
- Dudziński, K., S. Walukiewicz. 1987. Exact methods for the knapsack problem and its generalizations. *Eur. J. Oper. Res.* **28**(1): 3–21.
- Ellickson, P. B., S. Misra. 2008. Supermarket pricing strategies. *Market. Sci.* **27**(5): 811–828.
- Fibich, G., A. Gavious, O. Lowengart. 2003. Explicit solutions of optimization models and differential games with nonsmooth (asymmetric) reference-price effects. *Oper. Res.* **51**(5): 721–734.
- Foekens, E. W., P. S. H. Leeflang, D. R. Wittink. 1998. Varying parameter models to accommodate dynamic promotion effects. *J. Econ.* **89**(1): 249–268.
- Gedenk, K., S. A. Neslin, K. L. Ailawadi. 2006. Sales promotion. M. Krafft, M. K. Mantrala, eds. *Retailing in the 21st Century*. Springer, Berlin, 345–359.
- Håstad, J. 2001. Some optimal inapproximability results. *J. ACM (JACM)*. **48**(4): 798–859.
- Hartline, J. D., V. Koltun. 2005. Near-optimal pricing in near-linear time. F. Dehne, J. R. Sack, A. López-Ortiz, eds. *Workshop on Algorithms and Data Structures*. Springer, Berlin, 422–431.
- Heerde, H. J. V., P. S. H. Leeflang, D. R. Wittink. 2000. The estimation of pre- and post-promotion dips with store-level scanner data. *J. Mark. Res.* **37**: 383–395.
- Herer, Y.T., M. Tzur. 2001. The dynamic trans-shipment problem. *Nav. Res. Logist.* **48**(5): 386–408.
- Karp, R. M. 1972. *Reducibility Among Combinatorial Problems*. Springer, Berlin.
- Kopalle, P. K., A. G. Rao, J. L. Assuncao. 1996. Asymmetric reference price effects and dynamic pricing policies. *Market. Sci.* **15**(1): 60–85.
- Lal, R., R. Rao. 1997. Supermarket competition: The case of every day low pricing. *Market. Sci.* **16**(1): 60–80.
- Macé, S., S. A. Neslin. 2004. The determinants of pre- and post-promotion dips in sales of frequently purchased goods. *J. Mark. Res.* **41**: 339–350.
- Morávek, J. 1970. A note upon minimal path problem. *J. Math. Anal. Appl.* **30**(3): 702–717.
- Nijs, V. R., M. G. Dekimpe, J. B. E. Steenkamps, D. M. Hanssens. 2001. The category-demand effects of price promotions. *Market. Sci.* **20**(1): 1–22.
- Özer, Ö., O. Ozer, R. Phillips. 2012. *The Oxford Handbook of Pricing Management*. Oxford University Press, Oxford.
- Pisinger, D. 1995. A minimal algorithm for the multiple-choice knapsack problem. *Eur. J. Oper. Res.* **83**(2): 394–410.
- Popescu, I., Y. Wu. 2007. Dynamic pricing strategies with reference effects. *Oper. Res.* **55**(1): 98–112.
- Porteus, E. L. 1990. Stochastic inventory theory. *Handbooks in OR & MS*: **2**: 605–652.
- Schrijver, A. 2003. *Combinatorial Optimization: Polyhedra and Efficiency*, Vol. 24. Springer Science & Business Media, Berlin.

- Subramanian, S., H. D. Sherali. 2010. A fractional programming approach for retail category price optimization. *J. Global Optim.* **48**(2): 263–277.
- Talluri, K. T., G. J. van Ryzin. 2005. *The Theory and Practice of Revenue Management*. Springer, Berlin.
- Vazirani, V. V. 2013. *Approximation Algorithms*, Springer Science & Business Media, Berlin.
- Zhao, W., Y. Zheng. 2000. Optimal dynamic pricing for perishable assets with nonhomogeneous demand. *Management Sci.* **46**(3): 375–388.