An Efficient Algorithm for Dynamic Pricing using a Graphical Representation

Maxime C. Cohen
NYU Stern School of Business, New York, NY 10012, maxime.cohen@stern.nyu.edu

Swati Gupta
Massachusetts Institute of Technology, Cambridge, MA 02139, swatig@mit.edu,

Jeremy J. Kalas
Massachusetts Institute of Technology, Cambridge, MA 02139, jkalas@mit.edu

Georgia Perakis
Massachusetts Institute of Technology, Cambridge, MA 02139, 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 optimal prices, while satisfying several important practical business rules. The strength of our work lies in a graphical model reformulation we introduce, which allows us to use ideas from combinatorial optimization. Contrasting to previous literature, we do not make any assumptions on the structure of the demand functions. 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 the current demand) and in the number of items. Consequently for problems with large memory, we show that the profit maximization problem is NP-hard by presenting a reduction from the Traveling Salesman Problem. We introduce the *discrete reference price model* which is a discretized version of the commonly used reference price model, accounting for an exponentially smoothed contribution of all the past prices. This discrete model allows us to capture the fact that customers do not form reference prices with infinite precision. For this model, we show that the problem can be solved efficiently with low runtimes. We then approximate several common demand functions by using the discrete reference price model. Next, we extend the reference price model to handle cross-item effects among multiple items using the notion of a *virtual* reference price. To allow for scalability of our approach, we cluster the items into blocks, and show how to adapt our methods to incorporate global business constraints which are important and challenging in practice. Finally, we apply our solution approaches using demand models calibrated by supermarket data, and show that we can solve realistic size instances in a few minutes.

Key words: Dynamic Pricing, Retail Pricing, Layered Graph, Reference Price Model

1. Introduction

Dynamic pricing plays a significant role in determining the profit of any commodity-based industry. In supermarkets, an important part of dynamic pricing translates to promoting the right product at the right time using the appropriate promotion depth. In particular, price promotions can dramatically increase product visibility, store traffic, and sometimes cause customers to switch brands or bolster customer loyalty. A study by A.C. Nielsen during January-July 2004 estimated that 12–25% of supermarket sales in five European countries (Great Britain, Spain, Italy, Germany and France) were made during promotion (Gedenk et al. (2006)).

Price variations not only help retailers meet their sales targets, but also contribute to a significant percentage of the total profits. One of the supermarket industry characteristics lies in the small profit margins that retailer perceives for most items. A report published by the Community Development Financial Institutions (CDFI) Fund states that the average profit margin for 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 for 2012 was also close to 2010's 1.9% average. These studies provide ample evidence that manipulating prices gives the retailers a good handle on increasing their sales and profits. However, the current process of planning prices for items in supermarkets is still fairly manual and does not account for the effect of a sale-item on other items in the same category. The natural question is, *can we develop efficient optimization models that cater to various business requirements and can be used to solve realistic size intances?*

In this paper, 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., a quarter of 13 weeks). In practice, retailers need to obey several business rules, for example to satisfy a limited number of price changes (a full discussion is presented in Section 2.1). Our approach can handle general non-linear demand models that capture behavioral effects that are 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 between similar items). We consider demand models that are general non-linear and time dependent functions of the current and past prices, and we seek to solve the profit maximization problem faced by the retailer. We develop efficient methods to solve this problem, allowing the retailer to test several *what-if* scenarios in order to understand better the impact of dynamic 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 provide efficient and practical methods for solving the problem.

1.1. Contributions

Maximizing profits through dynamic pricing is an important business problem that has captured the attention of both retailers and researchers. The contributions of the paper can be summarized as follows.

- *Formulation of the problem as a layered graph.*

We present a way of solving 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 business rules on the pricing policy by modifying the structure of the graph.

- *NP-hardness and complexity results.*

Using the graphical representation, we provide an NP-hardness result by reducing our problem from the Traveling Salesman Problem. We then derive complexity results implying that our method scales linearly with the number of time periods but is exponential in the memory of the model (number of past prices that affect the current demand) as well as the number of items.

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

We propose the *discrete reference price model* which is a discretized version of the commonly used reference price model. This allows us to capture the fact that customers do not form reference prices with infinite precision. We first 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 with low runtimes for the profit maximization problem. Third, we propose a procedure to approximate several common demand functions by using the discrete reference price model. This allows us to solve instances with a large memory parameter in less than a millisecond, while having a guarantee on the quality of the approximation.

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

We extend the results and analysis for 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 product category. To increase the tractability of our approach, we introduce the notion of blocks and organize items into smaller clusters such that the cross-item interactions across blocks is negligible. However, it now becomes more challenging to impose global constraints on the promotions across the different blocks. Using ideas from combinatorial optimization, we limit the total number of promotions across blocks by solving a multi-choice knapsack problem. We also propose methods to handle price-ordering and exclusivity constraints that are often important in retail.

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

Using data we obtained from Oracle Retail, we evaluate the methods proposed in this paper using actual coffee data from a large supermarket. We show that our solution approaches can solve realistic size instances in a few minutes.

1.2. Literature Review

Dynamic pricing and sales promotions are extensively studied in the literature. A recent related work on scheduling price promotions can be found in Cohen et al. (2017), where the authors provide an optimization formulation. They propose an efficient approximation method (with tight parametric bounds on the performance guarantee for different classes of demand models) for solving the problem based on discretely linearizing the objective function and solving a linear program. Our paper has a similar motivation but bears at least four 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 show that the problem is NP-hard. Second, the algorithms developed in this paper yield exact solutions. Third, we extend our treatment to address the case of multiple items. Fourth, the problem considered in this paper is not restricted to promotions as we study a general dynamic pricing problem.

Sales promotions are well-studied in the field of marketing (see Blattberg and Neslin (1990) and the references therein). Some retailers such as Walmart employ an everyday-low-price-strategy (see, e.g., Lal and Rao (1997)), whereas many others use temporary price reductions (aka promotions) on a selected subset of items (examples of such works include Blattberg et al. (1995), Greenleaf (1995) and Nijs et al. (2001)). For a comparison between the two pricing strategies, see Ellickson and Misra (2008) and the references therein. In this paper, we consider the latter pricing strategy. It has been observed that temporary price reductions may lead to a reduction in future sales, a phenomenon which is referred to as the *post-promotion dip* effect. Researchers in the marketing community typically focus on developing and estimating demand models, e.g., linear regression or choice models, in order to draw managerial insights about promotions. For example, Foekens et al. (1998) study econometrics models based on scanner data to examine the dynamic effects of sales promotions. One of the methods used in the marketing literature to model the post-promotion dip effect is to model demand as a function of not only the price in the current period, but also the prices in the most recent periods (Mela et al. 1998, Heerde et al. 2000, Macé and Neslin 2004, Ailawadi et al. 2007).

Our work is naturally related to the field of dynamic pricing (see e.g., Talluri and van Ryzin (2005) and the references therein). More specifically, in the operations management literature, researchers study sales promotions from the angle of how to optimize the dynamic pricing policy given that consumer behavior leads to post-promotion dips in demand. A common approach used in the dynamic pricing literature is to model consumers using a reference price model (Kopalle et al. 1996, Fibich et al. 2003, Popescu and Wu 2007, Greenleaf 1995). The reference price model assumes that consumers form a reference price based on past prices. Then, the consumers compare the current price of the product to the reference price as a benchmark. Prices above the reference price reduce demand, whereas prices below the reference price lead to an increase in demand. Kopalle et al. (1996), Fibich et al. (2003) and Popescu and Wu (2007) study an infinite-horizon dynamic pricing problem with a reference price demand 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 the supermarket industry and include important practical 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, and solve the promotion problems for several items simultaneously in a retail context. The work by Chen et al. (2016) considers 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. Note that their formulation cannot easily handle price-dependent business constraints. On the other hand, our paper can handle several practical business rules, provides NP-hardness and complexity results, and studies the setting with multiple items. A recent work by Wang (2016) studies the dynamic pricing problem with the reference price model, and shows that inter-temporal price discrimination can be achieved by a cyclic pricing policy.

Finally, our work is related to the field of retail operations and more specifically pricing problems under business rules. One of the constraints considered in this paper 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 a backward recursion. The computational complexity of their approach grows linearly with the number of discrete time intervals. Our approach is different in nature as it considers a general demand function form that depends on current and past prices. Our method is based on solving a dynamic program that also yields a complexity polynomial in the number of time periods. 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 paper, we also consider a model for multiple items that includes several global inter-item constraints.

On the technical side, we use concepts related to graph theory and combinatorial algorithms, in particular algorithms for maximum weighted path in a directed graph, dynamic programming and approximation 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), Vazirani (2013), Garey and Johnson (1979) are excellent references to review these concepts.

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

NP-hardness and the complexity results. We then introduce and study the discrete reference price model in Section 4. Section 5 considers the problem for multiple items and extend the results for this setting. Computational experiments using supermarket data and conclusions are presented in Sections 6 and 7 respectively. Most of the proofs of the Theorems and Propositions are relegated to the Appendix.

2. Model and Assumptions

Given a set of n items in a category of products and a finite planning horizon of T time periods, the profit maximization problem aims to decide the price of each item at each period so as to maximize the total profit. The prices for the items are assumed to come from a discrete set, for example the prices 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 denote the number of time periods in the planning horizon by T, the unit cost of the item at time $t \in \{1, \dots, T\}$ as c_t and the discrete set of admissible prices, i.e., the *price ladder* by $\mathcal{Q}_p = \{q^0 > q^1 > \cdots > q^k > \dots > q^Q\}$. The regular price (i.e., the maximum price element) 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 lead to post-promotion dip effect (see, e.g., Macé and Neslin (2004)) due to the stockpiling behavior of consumers. In other words, for particular items, customers will purchase larger quantities for future consumption (e.g., toiletries and non-perishable goods). Therefore, 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 $p_{t-1}, p_{t-2}, \dots, p_{t-m}$. In addition, our models allow to have the flexibility of assigning 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, the parameter m can be 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, as well as on demand seasonality and trend. Mathematically, the demand at time t is given by:

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

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

$$\begin{aligned} \max_{p_t} \quad & \sum_{t=1}^{T} (p_t - c_t) d_t(\mathbf{p_t}) \\ \text{s.t} \quad & \text{Various known business rules} \end{aligned} \tag{2}$$

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. In addition, the price ladder for an item can be time-dependent. For simplicity, we assume that the elements of the price ladder are time independent but all the results of this paper still hold when this assumption is relaxed.

2. *Limited number of price changes.* The retailer may want to limit the frequency of the 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 to train customers to be deal seekers. For example, it may be required to vary the price of a particular product at most $L = 3$ times during the quarter.

3. *Separating periods between successive price changes.* A common additional 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 one another, this may hurt the store image and incentivize consumers to behave more as deal-seekers. In addition, this type of business requirement may be dictated directly by the manufacturer that sometimes restricts the frequency of promotions in order to preserve the image of the brand.

4. *Inter price constraints.* Very often, the retailer wants to impose constraints on the prices at the different time 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 at the same value.

5. *Inter-item constraints.* The category manager may need to satisfy some constraints that links the prices of the different items within the category. A very common business rule is to limit the total number of price changes for all the items during the selling season. We provide a more detailed discussion on this type of constraints in Section 5.4.

2.2. Assumptions

We first consider the problem for a single item and we explore the extension to multiple items in Section 5. We assume that at each time period t, the retailer orders the item from the supplier at a linear ordering cost that can vary over time, i.e., each unit sold in period t costs c_t. This assumption holds under the conventional wholesale price contract which is frequently used in practice as well as in the academic literature (see for example, Porteus (1990)).

We also consider *the demand to be specified by 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

[1] This paper 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 in p_t from the regular price q^0). To avoid confusion, we only present the case of price changes.

affect demand: seasonality, current and past prices. As a result, the estimated demand models provide an accurate prediction with a low forecast error. In particular, the out-of-sample R^2 and $MAPE$ are between 0.85 and 0.96, and 0.1 and 0.3 respectively (see, estimation results with actual data in Cohen et al. (2017)). A possible typical process in practice is to estimate a demand model from historical data and then to compute the optimal prices based on the estimated demand model. In addition, one can resolve the optimization problem under various sensitivity analysis scenarios in order to obtain a robust pricing policy. For the multiple items setting studied in Section 5, we consider cross-item effects in the demand function (i.e., the price of item i can affect the demand of item $j \neq i$).

Finally, we assume that *the retailer always carries enough inventory to meet demand* in each period, i.e., sales are equal to demand. This assumption does not apply to all products and retail settings. For example, it is common practice in the fashion industry (e.g., Talbots or Zara) to intentionally produce limited amounts of inventory. Nevertheless, the assumption that the retailer carries enough inventory is reasonable in settings such as supermarket items like coffee and soft drinks. These products are called fast-moving consumer goods and are typically available all year round. Since their shelf lives is usually greater than six months, customers have been conditioned to expect that these products would always be in stock at retail stores. In the data we have, we actually observed that the inventory was not issue and saw very few events of stock-outs over a two-year period. This can be justified by the fact that supermarkets have a long experience with inventory decisions and accumulated large data sets allowing them to develop sophisticated forecasting demand models to support capacity and ordering decisions. Many such models were developed in last two decades such as the work in Cooper et al. (1999). In addition, retailers are aware of the negative effects of stocking out of promoted products (see, e.g., Corsten and Gruen (2004)) and use accurate demand estimation models in order to forecast demand and plan inventory accordingly.

We first consider and study the case of a single item. This allows us to present the main techniques in a more concise way. In addition, several categories of items 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. The setting with multiple items is studied in Section 5, where we extend our results to capture cross-item effects in demand.

3. Graphical Representation

In this section, we present the construction of our graphical representation to model the profit maximization problem as a maximum weighted path problem on a layered graph. Then, we report the complexity results for the various cases and we conclude by showing that the problem is NP-hard.

3.1. Constructing the graph

Recall that we denote our planning horizon by T and the discrete price ladder by $\mathcal{Q}_p = \{q^0, q^1, \dots, q^Q\}$. For each time period $t \in \{m, \dots, T\}$, we construct the nodes $(\mathbf{x}, t)$ where $\mathbf{x} \in \mathcal{Q}_p^m$, i.e., all possible $m-$tuples of prices in the ladder $\mathcal{Q}_p$. When the price ladder is time-varying, i.e., $p_t \in \mathcal{Q}_{t,p}$ for different time periods, we construct nodes $(\mathbf{x}, t)$ such that $\mathbf{x} \in \mathcal{Q}_{t,p}^m$. As discussed before, for ease of exposition, we will consider a static price ladder $\mathcal{Q}_p$. We illustrate the layered graph constructed for $\mathcal{Q}_p = \{5, 3, 1\}$, $m = 2$ and $T = 4$ in Figure 1. 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 $(x_2, x_3, \dots, x_m) = (y_1, y_2, \dots, y_{m-1})$, i.e., the prices are consistent at the overlapping time periods. The edges of the graph are given by $A = \big\{(\mathbf{x}, t), (\mathbf{y}, t+1) |\ \mathbf{x}, \mathbf{y} \in \mathcal{Q}_p^m, 1 \le t \le T-1\big\}$ 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, this weight is equal to the marginal profit obtained by introducing the price y_m at time $t+1$. Finally, we add arcs from the *source* node to all the nodes in time period m with the weight equal to the profit obtained due to the first m prices, i.e., $w((\text{source}, (\mathbf{x}, m)))$ is equal to $\sum_{t=1}^m (x_t - c_t) d_t(x_t, \dots, x_0)$. In addition, we connect all the nodes in the last time period, $t = T$, to the *sink* with a zero weight. Note that the graph we constructed forms a directed layered graph since the edges exist only between the 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 graphical representation.

PROPOSITION 1. *Consider the layered graph construction, as explained above. Any (simple) path P from the* source *node to the* sink *node 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 simple path from the *source* to the *sink* must use exactly one node from each layer for time periods $\{m, \dots, T\}$. As the edges are between price compatible nodes, there is exactly one price used at each period $\{1, \dots, T\}$. As a result, this leads to a price-assignment for each time period. In addition, summing the weights of the edges in the path yields the total profit induced by these prices. $\square$

Consequently, 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, i.e., $O(T|\mathcal{Q}_p|^{m+1})$ for the unconstrained version of

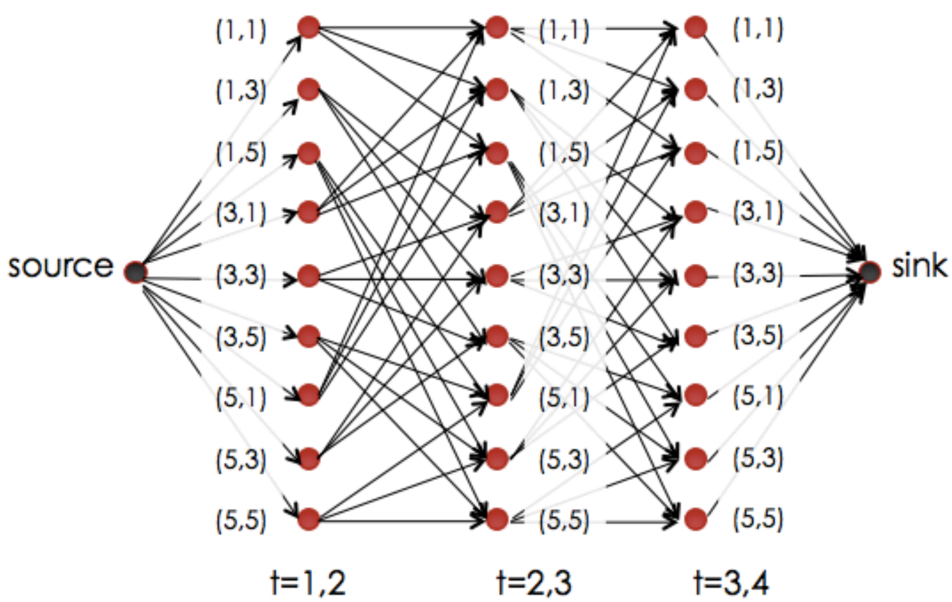

Figure 1 **Layered graph construction for the case where** $\mathcal{Q}_p = \{q^0 = 5, q^1 = 3, q^2 = 1\}$**,** $m = 2$ **and** $T = 4$**.**

the problem. Note that in order 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 (as a unit flow polytope, see for 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 we previously discussed, the runtime complexity for the unconstrained version of the problem is $O(T|\mathcal{Q}_p|^{m+1})$. We next present appropriate modifications to the graphical representation that allow to handle the practical business rules from 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 time periods to be always non-increasing. An additional example is to restrict the price of a product to be below some value or the price at the end of the horizon to be equal to the regular price. In such cases, one can simply delete the set of nodes and arcs that violate the rules of interest. In the markdown policy, all the arcs connecting a lower price to a higher price in the subsequent time period are simply deleted. Note that these reductions decrease the size of the graph relative to the unconstrained case and therefore, may improve the runtime.

Case 2. Limiting the number of price changes. As we previously discussed, an important business rule is to restrict the number of price changes for the item to L. In this case, for each node in the graph, one can maintain a table of the maximum achievable profit with some $k \in \{1, \dots, L\}$ price changes used so far. As a result, for each time period $t \in \{m, \dots, T\}$, we construct the nodes $(\mathbf{x}, l, t)$ where $\mathbf{x} \in \mathcal{Q}_p^m$, $l \in \{0, 1, \dots, L\}$ and $t \in \{1, \dots, T\}$. In this case, an edge between nodes $(\mathbf{x}, l_1, t)$ and $(\mathbf{y}, l_2, t+1)$ exists if and only if:

(i) $(\mathbf{x}, \mathbf{y})$ are price-compatible, and

(ii) $l_2 = l_1 + 1$ when $y_m \neq x_m$ and $l_2 = l_1$ otherwise.

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

Case 3. No-touch constraints. In this case, our goal is to restrict the minimal duration S between two successive price changes. To this end, one can maintain a parameter at each node that denotes the number of periods before which a price variation can occur. For each period $t \in \{m, \dots, T\}$, we construct the nodes (x, s, t) where $\mathbf{x} \in \mathcal{Q}_p^m$, $s \in \{0, 1, \dots, S\}$ and $t \in \{1, \dots, T\}$. An edge between nodes $(\mathbf{x}, s_1, t)$ and $(\mathbf{y}, s_2, t+1)$ exists if and only if *all* of the following are satisfied:

(i) $(\mathbf{x}, \mathbf{y})$ are price-compatible,

(ii) $s_2 = \max(s_1 - 1, 0)$ if $s_1 \geq 0$ and $y_m = x_m$ (i.e., no price change at $t+1$), and

(iii) $s_2 = S$ if $s_1 = 0$ and $y_m \neq x_m$ (i.e., a price change at $t+1$).

The edges of the graph ensure that we correctly count the number of time periods before a price variation can occur. The edges weights are defined as before, i.e., $w((\mathbf{x}, s_1, t), (\mathbf{y}, s_2, t+1)) = (y_m - c_{t+1}) d_{t+1}(y_m, \dots, y_1, x_1)$. In addition, we delete the nodes with price vectors that violate the no-touch constraint. Since we maintain a tuple of m prices in each node, we can have at most $\lceil \frac{m}{S} \rceil$ price changes in any given node. Therefore, the total number of nodes in the graph can be at most $|\mathcal{Q}_p|^{\lceil \frac{m}{S} \rceil}$. Note that the nodes in which the s parameter is positive have exactly one outgoing edge, whereas nodes with $s = 0$ have up to $T(S+1)|\mathcal{Q}_p|$ outgoing edges. The total number of edges in the graph is thus of the order $O(TS|\mathcal{Q}_p|^{\lceil \frac{m}{S} \rceil} + T|\mathcal{Q}_p|^{\lceil \frac{m}{S} \rceil + 1})$, corresponding to the running time required to solve the maximum profit price assignment.

Recall that our demand model assumes that the demand at time t depends on the current price p_t, 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 towards the end of the horizon, as the price effect on future demand is ignored. A possible way to address this issue is to consider a rolling horizon such that the price at time T affects the demand at times 1 to m (similarly, the price at time $T-1$ affects the 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-effects anymore. One can solve this version of the problem by using our graphical representation together with an increased length of planning peril of $2T$. This only doubles the graph size and the run time complexity remains the same as the unconstrained case.

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

Table 1 **Summary of runtime complexity results for various single item promotion optimization problems.**

We summarize the runtimes of the aforementioned cases in Table 1. Note that incorporating business rules such as markdown prices or no-touch constraints improves the optimization problem by reducing the size of the instance. As we previously mentioned, this approach is independent of the demand structure so that all the 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 (for more details, see the computational experiments in Section 6). Next, we discuss the hardness of the profit maximization problem for the case with a large memory parameter m. Note that the model we study in this section including the different business rules can be written and solved as a dynamic program. We include the relevant discussion in Appendix A.

3.3. Hardness Result

We show that when the memory parameter m is large, the profit maximization problem is NP-hard. We present a reduction of the Traveling Salesman Problem (TSP) (see e.g., Karp (1972) and Schrijver (2003)) to our problem. Given a set of n cities and the traveling distances between each pair of cities i, j, $\{c_{i,j}\}$, the TSP seeks to find the shortest tour that visits each city exactly once.

THEOREM 1. *The profit maximization problem is NP-hard when the memory parameter $m = \Omega(T)$[1], where T is the length of the horizon, unless $P = NP$. Moreover, it is NP-hard to approximate the profit maximization problem to any approximation factor polynomial in T.*

[1] We 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.

Proof. To prove the NP-hardness of the profit maximization problem, we consider an arbitrary instance I_{tsp} of the TSP, 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_{tsp}, thereby proving that a polynomial time algorithm for profit maximization problem in case of memory of the order of $\Omega(T)$ is not possible, unless $P = NP$. Moreover, since it is NP-hard to approximate the TSP to any approximation factor polynomial in n (see, e.g., Vazirani (2013)), the same inapproximability result applies to profit maximization problem due to the reduction we present next.

Let the TSP instance, I_{tsp}, be defined over n cities, such that the traveling distance between cities i and j is denoted by $\{c_{i,j}\}$. Let the set of all permutations of $[n]$ be $S(n)$ (i.e., the set of all possible tours), and let the cost of each tour in $S(n)$ be given by a function $C : S(n) \to \mathbb{R}$.

We construct an instance of the profit maximization problem, I_Π, as follows: let the time horizon be $T = n$ and the memory parameter be $m = n-1$. We set the unit cost of the item to be zero at all times, i.e., $c_t = 0$ for all $t \in [T]$. We define the discrete price ladder as $\mathcal{Q}_p = \{q^0 > q^1 > \ldots > q^{n-1}\}$, such that $q^i = n - i$ for $i \in \{0, \ldots, n-1\}$. Let the demand function for the profit maximization problem to be as follows:

$$d_t(\mathbf{p}_t) = \begin{cases} Z - C(\mathbf{p}_t) & \text{if } t = n,\ \mathbf{p}_t \in S(n),\ p_T = n \\ 0 & \text{if } t < n \\ 0 & \text{if } t = n,\ \mathbf{p}_t \notin S(n). \end{cases}$$

Here, $\mathbf{p}_t \in \mathcal{Q}_p^n$ and $Z = \sum_{i \le j, i,j \in [n]} c_{i,j}$, i.e., Z represents the sum of all the distances between the cities. In other words, the demand is non-zero only at the last time period $t = T$, if $p_T = n$. Note that since we did not impose any assumption on the demand function, one can encapsulate the TSP cost function into the above demand function. As a result, our finite horizon profit maximization problem reduces to:

$$\max_{p_t} \sum_{t=1}^{T} \big(p_t - c_t\big) \cdot d_t(\mathbf{p_t}) = \max_{\mathbf{p}_t \in S(n)} p_T \cdot d_T(\mathbf{p_t}) = \max_{\mathbf{p}_t \in S(n)} n \cdot \big[Z - C(\mathbf{p}_t)\big]. \tag{3}$$

Note that the maximum profit is obtained by a permutation of prices such that the price at the last period is n, and the permutation corresponds to the shortest TSP tour with respect to the given cost between the cities. Therefore, solving the profit maximization problem in polynomial time would in fact contradict the NP-hardness of the TSP and this concludes the proof. □

Having shown that the profit maximization problem is NP-hard to approximate (to any approximation factor polynomial in T) for large memory, a natural question to consider at this point is the following. *What assumptions on the demand function would still render the problem in polynomial space even with large memory?* In the next section, we study the reference price model, that is a special case of a demand with a large memory, and develop an efficient approximation method to solve the problem.

4. Reference Price Model

In this section, we introduce and study a discrete version of the commonly used reference price model (see, e.g., Kopalle et al. (1996), 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 on 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, we believe that customers do not form a reference price of arbitrary real numbers. Consequently, we propose to only allow reference prices that belong to a discrete price 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 (1), where the memory 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 ladders. Subsequently, we approximate a general demand with linear past price effects using the discrete reference price model, and provide bounds on the profit performance.

4.1. Discrete Reference Price Model

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

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

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

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

The parameter ϕ denotes the price sensitivity with respect to the reference price, whereas $\beta^0 + \phi$ corresponds to the price sensitivity with respect to p_t. Note that the reference price at time t can be rewritten in terms of the past prices as follows:

$$r_t = (1-\theta)p_{t-1} + \theta(1-\theta)p_{t-2} + \theta^2(1-\theta)p_{t-3} + \cdots = (1-\theta)\sum_{k=1}^{T}\theta^{k-1}p_{t-k}.$$

As a result, the demand at time t from equation (5) can also be written in terms of the current and past prices:

$$d_t(p_t, p_{t-1}, \ldots, p_{t-T}) = a_t - (\beta^0 + \phi)p_t + \phi\sum_{k=1}^{T}(1-\theta)\theta^{k-1}p_{t-k}. \tag{6}$$

One can see that equation (6) depicts a model that depends on the current and m past prices, when the memory parameter m is equal to the length of the horizon 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 $g(\cdot): \mathbb{R} \to \mathbb{R}$ that depends explicitly on the difference between the price and the reference price. We assume that the function $g(\cdot)$ is G-Lipschitz (i.e., $||\nabla g(x)|| \leq G$ for all $x \in \mathbb{R}$). Note that the demand model in equation (7) includes cases where the reference price effect can be asymmetric.

Recall that the reference price represents the price that consumers are willing to pay for the item based on past prices. It seems then reasonable to assume that r_t lies in a discrete set where the values are rounded to some finite level (e.g., 5 cents for an item that costs 1 dollar). In particular, consumers will not form a reference price with a very high accuracy. We denote the discrete reference price by $\hat{r}_t$, and its set of values by $\mathcal{Q}_r = \{r^0 > r^1 > \cdots > r^n > \ldots > r^N\}$. Note that $r^0 = q^0$ (the regular price) and $r^N = q^Q$ (the smallest element of the price ladder). Thus, we introduce the discrete reference price at time t which is now obtained by (the exact way of rounding does not affect any of our results, so we assume simply rounding to the closest element in the set $\mathcal{Q}_r$ defined below):

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

We next show that the discrete reference price model tends to the continuous reference price model in the limit of the discretization of the price ladder, by bounding the difference in the discrete and continuous reference prices. We choose a discrete reference price ladder $\mathcal{Q}_r$ such that $r^k - r^{k-1} = \epsilon;\ \forall k = 1, 2, \ldots N$, for some given $\epsilon > 0$.

Proposition 2. *Consider the continuous reference price model in* (4) *and the proposed discrete reference price model obtained by rounding the reference price to the nearest value in the set* $\mathcal{Q}_r$ *constructed with parametrization* $\epsilon > 0$ *as in* (8). *Then, the difference in the continuous and discrete reference prices at time* t *can be 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 time period, we have: $\hat{r}_1 = \text{round}\Big[(1-\theta)p_0 + \theta r_0\Big] = r_1 \pm \epsilon$. Then, for $t=2$, one can write: $\hat{r}_2 = \text{round}\Big[(1-\theta)p_1 + \theta \hat{r}_1\Big] = \text{round}\Big[(1-\theta)p_1 + \theta(r_1 \pm \epsilon)\Big]$. 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: $\hat{r}_2 = \text{round}\Big[r_2 \pm \theta\epsilon\Big] = r_2 \pm (\theta+1)\epsilon$. We next proceed by induction on t. We assume that the claim is true for $t=k$, i.e., $\hat{r}_k = r_k \pm (\sum_{u=0}^{k-1}\theta^u)\epsilon$. We next show for time $t = k+1$. We have:

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

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

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

concluding our proof. □

Note that since $0 \le \theta < 1$, the difference is guaranteed to be within a constant factor of ϵ. We next show that in the limit of discretization of the reference price ladder, the demand and the total profit also tend to their continuous model counterparts. As a result, the discrete reference price model is not far from 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 the customer preferences can only be granular up-to 1 cent.

Corollary 1. *Consider the demand model in* (7) *and the discrete reference price model obtained by rounding the reference price to the nearest value in the set $\mathcal{Q}_r$ with precision $\epsilon > 0$. Then, the difference in the demand value and the total profits predicted by the two models at time t can be bounded by:*

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

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

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

This follows from Proposition 2 and from the fact that the function $g(\cdot)$ is G-Lipschitz. For the special case with a linear demand model (see equation (5)), we have:

$$|\hat{d}_t - d_t| = \phi|\hat{r}_t - r_t| \le \phi\frac{1-\theta^{t-1}}{1-\theta}\epsilon. \tag{12}$$

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

We next present a polynomial algorithm that solves the promotion optimization problem under the discrete reference price model efficiently. More precisely, we propose a dynamic program algorithm that is polynomial in both $|\mathcal{Q}_p|$ and $|\mathcal{Q}_r|$. This result holds independently of the way the reference price is rounded.

Graphical Representation. Suppose we are given a discrete reference price ladder $\mathcal{Q}_r = \{r^0 > r^1 > r^2 > \ldots > r^N\}$, where $r^0 = q^0$, $r^N = q^Q$ such that $r^k - r^{k-1} = \epsilon$; $\forall k = 1, 2, \ldots N$, for some $\epsilon > 0$. We can construct a layered graph such that computing a maximum weighted path yields the optimal prices. For each time period $t \in [1, \ldots, T]$, 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 $2 + T|\mathcal{Q}_p||\mathcal{Q}_r|$). We call a pair of nodes, $(\mathbf{x}, t), (\mathbf{y}, t+1)$, *price-compatible* if $(1-\theta)x_p + \theta x_r - \epsilon \le y_r \le (1-\theta)x_p + \theta x_r + \epsilon$ for $1 \le t \le T-1$, i.e., the reference prices are updated in a consistent manner in consecutive periods. We then add a set of arcs to the graph given by $A = \big\{(\mathbf{x}, t), (\mathbf{y}, t+1) |\ \mathbf{x}, \mathbf{y} \in \mathcal{Q}_p \times \mathcal{Q}_r, 1 \le t \le T-1\big\}$ 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 marginal profit obtained by introducing the price y_p at time $t+1$. Finally, we add arcs from the *source* node to all the nodes in time 1 with weight $(x_p - c_1) d_1(x_p, x_r)$, where we assume that $x_r = 1$. In addition, we connect all the nodes in the last time period, $t = T$, to the *sink* node with zero weight. Note that the graph we constructed forms a directed layered graph since the edges exist only between the 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 to the nearest price in $\mathcal{Q}_r = \{1, 1.25, 1.5, 1.75, 2\}$.

In the same spirit as in Proposition 1, there exists an equivalence between optimizing prices for the discrete reference price model and finding the maximum weighted path in the graphical representation explained above.

Proposition 3. *Consider the layered graph constructed for the discrete reference price model* (8)*, as explained above. Any (simple) path P from the* source *to the* sink *corresponds to a price assignment $(p_1, p_2, \ldots, p_T)$. Moreover, the sum of the weights of the edges on P corresponds to the total profit in the discrete reference price model. As a result, 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. Note that the time complexity of finding the maximum profit path in

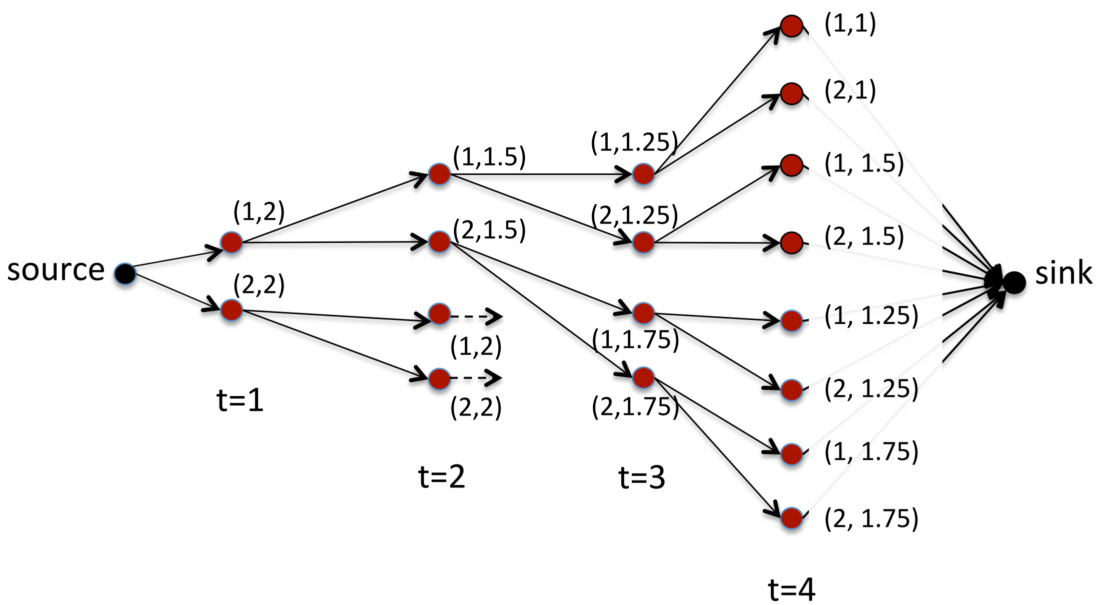

Figure 2 **Layered graph construction with $\mathcal{Q}_p = \{q^0 = 2, q^1 = 1\}$, $x_r = 2$ for $t = 1$, $\theta = 0.5$, $\epsilon = 0.25$ and $T = 4$, with reference prices rounded down to the nearest multiple of 0.25.**

the graph 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 within the price ladder), we obtain a time complexity of $O(T|\mathcal{Q}_p|^3)$. Recall that we are concerned with optimizing the total profits for a demand model with a large memory parameter (i.e., $m = T$). We then have proposed an alternate discrete reference price model which effectively behaves as if the memory parameter is equal to 2 by collecting all the required information in a single quantity (the discrete reference price at each time). Finally, one can easily incorporate business constraints in a similar manner as in the general demand model from Section 3.2 (we include a discussion for completeness in Appendix A.2). In addition, as we discuss in Section 6, the methods we propose run in less than 0.1s for realistic size instances.

4.2. Reference Price Model Approximation

As discussed in Section 3.3, for a general demand function with a large memory parameter m, our method may be not 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 demand model can be solved efficiently. In this section, we present a way to approximate several general demand functions by using the discrete reference price model, and derive bounds on the quality of the approximation. We consider three commonly used demand models in the retail industry that can be easily estimated from data: linear past prices, log-log past prices and log-linear past prices. We report the treatment for demand models with linear past prices effects such as in (13), and include the analysis for the log-log and log-linear demand models in Appendix B. In all three cases, we develop a procedure to approximate the true demand function with a large memory parameter and provide an efficient algorithm to solve the problem.

We consider a model where the demand at time t depends explicitly on the current and past m prices, where m is equal to T (i.e., a setting with a large memory parameter). More precisely, the demand at time t can be written as:

$$d_t(\mathbf{p_t}) = f_t(p_t) + \beta^1 p_{t-1} + \cdots + \beta^T p_{t-T}. \tag{13}$$

The parameters $\beta^1, \ldots, \beta^T$ as well as the functions $f_t(\cdot)$ are usually estimated from data. A common assumption, confirmed by the data we have, requires the gain parameters to be non-negative and non-increasing: $\beta^1 \geq \cdots \geq \beta^T \geq 0$. In addition, very often the first coefficient β^1 is much larger relative to the other ones. Intuitively, the last price in time has a more significant 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 in the discrete reference price model and approximate the demand function in (13) to depend only on the current price p_t and on a single additional variable $\widetilde{r}_t$, called the modified reference price. In addition, we require $\widetilde{r}_{t+1}$ to depend only on $\widetilde{r}_t$ and p_t. We naturally impose the following relation[2]:

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

where $\widetilde{\theta}$ is a design parameter that aims to approximate the true demand model (13) with respect to the following approximated demand model:

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

In other words, equations (14) and (15) aim to approximate the true demand model in (13) by carefully choosing the parameters $\widetilde{\theta}$ and $\widetilde{\phi}$. More precisely, we need to approximate a linear function with T coefficients by using an approximation with only two parameters. Similarly as in (6), one can write:

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

Observe that if the coefficients β_i are decreasing by a constant factor, i.e., β^{i+1}/β^i is constant for all i, then we 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 will obtain an approximation.

[2] Note that for 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 tends to the continuous model when the discretization parameter ϵ approaches 0). In other words, equation (14) is obtained using continuous reference prices and not discretized reference prices.

Since we observed very often in our data that β^1 is larger relative to the other coefficients (i.e., the last price has a more significant effect relative to further past prices), we want to make sure that we match the effect of the past price p_{t-1}. In other words, we require that: $\widetilde{\phi}(1-\widetilde{\theta}) = \beta^1$, or equivalently the parameter $\widetilde{\phi}$ is always set such that $\widetilde{\phi} = \beta^1/(1-\widetilde{\theta})$. This leaves us with a single degree of freedom (the parameter $\widetilde{\theta}$) so as to approximate the $T-1$ remaining factors. Note that equation (16) becomes:

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

We next propose three different approximations that depend on the value of $\widetilde{\theta}$. In particular, we define:

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

$$\sum_{k=2}^{T} \left[\beta^k - \beta^1 \widetilde{\theta}^{k-1}\right]^2. \tag{18}$$

Note that one can compute $\widetilde{\theta}_{LS}$ numerically by searching, as minimizing the function in (18) is a single dimensional optimization problem.

The corresponding demand approximations are labeled as $d_t^{\min}$, $d_t^{\max}$ and d_t^{LS} respectively. Similarly, the total optimal profits using the different demand approximations are denoted by Π^{min}, Π^{max} and Π^{LS}. More precisely, Π^{min} corresponds to solving the approximated problem with $\widetilde{\theta} = \widetilde{\theta}_{min}$ (recall that the problem becomes tractable, as one can apply the efficient algorithm developed in Section 4.1).

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

$$\Pi^{\min} \leq \Pi^{LS} \leq \Pi^{\max},$$
$$\Pi^{\min} \leq \Pi^{True} \leq \Pi^{\max},$$

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

Proof. We show that using the approximation with $\widetilde{\theta}_{min}$ (resp. $\widetilde{\theta}_{max}$) provides a lower (resp. upper) bound relative to any other approximation (in particular, using $\widetilde{\theta}_{LS}$), as well as to the true demand function. For any given price vector $\mathbf{p_t}$, we have:

$$\Pi^{\min} = \sum_{t=1}^{T} (p_t - c_t) d_t^{\min} \leq \sum_{t=1}^{T} (p_t - c_t) d_t^{LS} \leq \sum_{t=1}^{T} (p_t - c_t) d_t^{\max} = \Pi^{\max},$$

$$\Pi^{\min} = \sum_{t=1}^{T}(p_t - c_t)d_t^{\min} \leq \sum_{t=1}^{T}(p_t - c_t)d_t^{\text{True}} \leq \sum_{t=1}^{T}(p_t - c_t)d_t^{\max} = \Pi^{\max}.$$

The four inequalities follow from the facts that β^i ($\forall i = 1, \ldots, T$) are all non-negative and that for the same given price vector, using a smaller value of $\widetilde{\theta}$ can only decrease the demand and hence the total profit (it is easy to observe that for 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). □

Note that the results from Propositions 2 and 4 imply that one can approximate the problem with the demand function in (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 the three parameters $\widetilde{\theta}_{min}$, $\widetilde{\theta}_{LS}$ and $\widetilde{\theta}_{max}$. The values in the y-axis correspond to the true and approximated (using the three approximations) values of β_t for $t = 1, 2, \ldots, 10$. One can see that both the true values and the ones obtained using $\widetilde{\theta}_{LS}$ are lower bounded by the $\widetilde{\theta}_{min}$ approximation and upper bounded by the $\widetilde{\theta}_{max}$ approximation. In addition, the approximation based on $\widetilde{\theta}_{LS}$ yields a good approximation very often. In particular, as we show in Section 6, for randomly generated instances, the 25 percentile and the median are 97.7% and 99.4% relative to the optimal profit respectively.

To summarize, one can solve the problem with the value $\widetilde{\theta}_{LS}$ and obtain an approximation solution efficiently. In addition, we have a lower and an upper bounds on the profit performance by solving the problem with $\widetilde{\theta}_{\min}$ and $\widetilde{\theta}_{\max}$. As discussed before, observe that for the reference price model, we have $\Pi^{\min} = \Pi^{\max}$ so that the approximation is exact. For any other demand function of the form (13), the approximation is not exact but provides good computational results in the tests we conducted (see Section 6).

5. Multiple Items

In this section, we extend the graphical models for the profit maximization problem of multiple items within a category. In particular, we consider a setting with n items for which the retailer needs to decide the prices of each item at each time of the planning horizon. 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 $p_{t-1}^i, \ldots, p_{t-m_i}^i$. However, the demand of item i also depends on the vector of current other prices: p_t^j for $j \neq i$. This aims to capture the cross item effects, i.e., a price variation on one item can affect the sales of the 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 substitute) or increase the sales of item k (in this case, items j and k are complements).

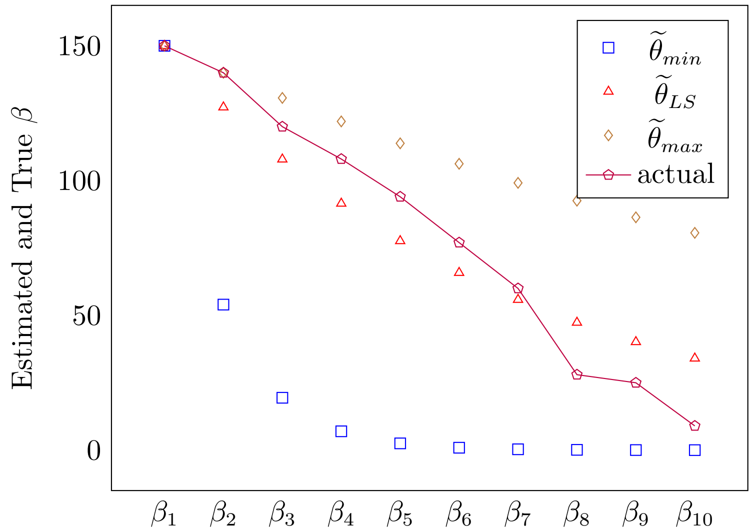

Figure 3 **An example of how the estimated parameters compare to the true parameters (the true parameter values are highlighted with the line-plot).**

We first observe that one can extend the graphical representation presented in Section 3 for the case of n items by expanding the size of each node in the graph. Namely, in each node we maintain a tuple of mn prices, i.e., m prices of each of the n items. The total number of nodes at each time period is then $|\mathcal{Q}_p|^{mn}$. Note that in order to simplify the exposition, we assume that each item has the same memory parameter with respect to past prices, i.e., $m_i = m \ \forall i = 1, 2, \ldots, n$ and the same price ladder $\mathcal{Q}_p$. Nevertheless, one can easily adapt the method for different values of m_i and price ladders. Consequently, we obtain a naive extension of the previous results. For example, the time complexity of the unconstrained problem for n items is given by: $T|\mathcal{Q}_p|^{n(m+1)}$. Unfortunately, this approach becomes 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 computational experiments in Section 6). In order to tackle this issue, we propose two alternative approaches that allow us to solve the problem in a more tractable fashion. Both methods borrow the intuition and results we developed for the reference price model in Section 4. The first method is based on having a reference price for each item, whereas the second approach considers a single *virtual* reference price for all the items. Next, we introduce and analyze both methods.

5.1. Model with n reference prices

In this section, we consider that consumers form a reference price based on past prices for each item separately. In particular, we assume that the demand of item i at time t is given by:

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

where the first term represents the effect of all the current prices and the second term captures the effect of the reference price of item i at time t, r_t^i. As in Section 4, r_t^i represents the price consumers are willing to pay for item i at time t based on past prices and is given by:

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

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

Using this representation, we first extend the result that the discrete reference price model for n items has time complexity equal to $T|\mathcal{Q}_p|^{n+1}\mathcal{Q}_r^n$, where $\mathcal{Q}_r$ is the reference price ladder (see Section 4.1). Therefore, this is a significant improvement in time complexity relative to the naive extension discussed before with a time complexity of $T|\mathcal{Q}_p|^{n(m+1)}$, for items with large values of m.

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, but instead consider a reference price for the coffee category. We call this notion of aggregate baseline the *virtual reference price*. In this section, we consider the model where 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 the current prices and the second term captures the effect of the virtual reference price at time t. As before, the virtual reference price 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 the prices at the previous time period. For example, the function $h(\cdot)$ can be a weighted average of the n past prices, the minimum, the maximum etc. The parameter $0 < \theta_V \leq 1$ represents the memory of past prices with respect to the virtual reference price and can be estimated from data.

Using this representation, one can consider discretized virtual reference prices as well. Then, the time complexity of the model reduces to $T|\mathcal{Q}_p|^{n+1}\mathcal{Q}_r$, where $\mathcal{Q}_r$ is the number of elements in the discrete ladder for the virtual reference price. Note that if the number of items n is large or if $\mathcal{Q}_r$

includes many elements (i.e., the parametrization ϵ is small), then this model is the only one (out of three approaches we considered) that is tractable and will solve the problem in a reasonable time frame. We compare the time complexity of the three models in Table 2 for the general case as well as for the case where $\mathcal{Q}_r = \mathcal{Q}_p$. We further compare the three models computationally in Section 6.4.

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

Table 2 Summary of time complexity results for multiple items.

Interestingly, the three methods we presented for multiple items can easily incorporate several business rules. First, one can satisfy business constraints for each item in a similar fashion 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 variations during the selling season for the entire category. Alternative examples include enforcing a relationship between prices of the different items (e.g., for a particular brand, the smaller size should be always cheaper relative to the larger format) and exclusivity deals. We will discuss in more detail the implementation of this type of global constraints in Section 5.4. Next, motivated by retail data we had access to, we describe a practical model that makes the approach presented in this section even more efficient.

5.3. Practical model with blocks

Very often, the demand of a particular product is 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 the price variation of certain products will have no impact on the sales of other products. For instance, two items may be in the same category but have a very weak connection in terms of substitution/complementarity (e.g., dark roast coffee and decaf products). Using actual data from supermarkets, we observed that the demand of a product was affected only by a few prices with some statistical significance. We next use this observation in order to refine the time complexity of the three previous algorithms and obtain a faster practical solution approach. In particular, out of the n different items in the category, 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) in order to partition the n products into K clusters (the value of K depends on the category) 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. In what follows, we assume that the n items can be clustered in K blocks, where the number of items in each block is quite smaller than n. We then consider that only the items in the same block affect each other, i.e., the demand of item i depends only on the prices of the items in the same block. Using this assumption, one can take advantage of this structure to improve the time complexity of the three algorithms mentioned above 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 illustration), the time complexities are given by:

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

A more concrete comparison of the runtimes for the three methods is presented in Section 6.4 on 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, when using the model with blocks. In the next section, we discuss adapting our methods to handle global business rules.

5.4. Business Constraints with blocks

Motivated from practical retail settings, we consider global business rules that aim to restrict the price changes for different items. For example, the retailer may have a restriction on the total number of price changes in a category. This rule might be imposed by a store manager who is concerned about preserving the image of the store. By restricting the total number of price changes, it also allows not to train consumers to be deal seekers and as a result, reach a better customer segmentation. A second very common restriction is called price-ordering constraints that dictate the price ordering of two items. Finally, we also consider exclusive deals offered by manufacturers in the supermarket industry.

We first discuss a scenario to illustrate the combinatorial blow up that occurs while trying to satisfy these global business constraints. Consider two blocks of items with no significant cross-item effects between them. For instance, one could consider two sets of items: one with dark roast coffee products (block 1), and the other with decaffeinated products (block 2). Suppose that by optimizing the prices independently for each block, the profits are maximized in block 1 by setting 6 promotions, whereas block 2 requires to offer 4 promotions. However, consider the global constraint in which the store enforces a limit of 8 promotions for all the coffee products. We then need to decide which promotions to remove from each block so that the overall profits are maximized while satisfying the global limit of 8 promotions. Note that the number of options increases exponentially with the number of blocks. We next show how to adapt our methods in order to avoid this combinatorial blow up, and to efficiently handle global constraints across multiple blocks.

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}, over the K blocks. To do so, we first compute the function $Y(\cdot) : \{1, \dots, K\} \times \{0, \dots, L_{total}\} \to \mathbb{R}$ 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 developed in 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 with at most j price changes is selected for block B_i (and $x_{(i,j)} = 0$, otherwise). In order to find the optimal price assignment for all the blocks that satisfies the global constraint of allowing at most L_{total} price changes, the following optimization problem can be solved:

$$
\begin{aligned}
&\max \sum_{i=1}^{K} \sum_{j=0}^{L_{total}} Y_i(j) x_{(i,j)} \\
&\sum_{j=0}^{L_{total}} x_{(i,j)} = 1 \ \forall i \in \{1, \dots, K\} && (19) \\
&\sum_{i=1}^{K} \sum_{j=0}^{L_{total}} j x_{(i,j)} \le L_{total} && (20) \\
&x_{(i,j)} \in \{0,1\} \ \forall i \in \{1, \dots, K\}, j \in \{1, \dots, L_{total}\}.
\end{aligned}
$$

Here, the decision variables are $x_{(i,j)}$ while $Y_i(j)$ are computed beforehand. Note that the constraints (19) ensure that for any block B_i, a specific number of price changes is selected. Constraints (20) ensure that the total number of price changes across all blocks does not exceed L_{total}, as requested. This is an instance of the well-studied 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)), yielding a running time of $O(KL_{total}^2)$. Therefore, this allows us to handle the restriction on the total number of price changes in a tractable fashion.

Case 2. Price-ordering constraints. Retailers may need to satisfy price orderings on some sets of items. For example, there exist price restrictions with respect to large versus small quantity of the same brand, premium quality versus private label, organic versus non-organic etc. A concrete example is the same brand with two different formats. In this case, one needs to impose that the price of the smaller format is always cheaper relative to the price of the larger format throughout the selling season. Consider items i and j (possibly in different blocks) such that the price of item i is required to be lower relative to the price of j at all times, i.e., we want to ensure that:

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

Let $\mathcal{Q}_i$ and $\mathcal{Q}_j$ 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 $t = 1, \ldots, T$, for each value $v \in \mathcal{Q}_i \cup \mathcal{Q}_j$. One can do so by simply deleting the nodes that violate these inequalities. Then, one can obtain the optimal prices that satisfies equation (21) by selecting the value of v that maximizes the total profit. In the worst-case, we require solving the profit maximization problem for $2|\mathcal{Q}_i \cup \mathcal{Q}_j|$.

Case 3. Exclusivity constraints. Trade funds dictated by manufacturers may impose an "exclusivity deal" that prohibits retailers from decreasing the prices of other competing products at the same time. For example, a manufacturer may offer a deal to the retailer that entails promoting some of the items in the set K_1 at a given time period 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 period 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 for each block twice.

6. Computational Experiments

In this section, we perform computational experiments to validate the efficiency and scalability of our solution approach. First, we test the exact dynamic program for the single item case using demand models calibrated from real data. Second, we apply our graphical methods to the reference price model presented in Section 4 and observe very low runtimes. Third, we evaluate the three proposed models for solving the multiple items setting, as outlined in Section 5. All the tests in this section reflect realistic scenarios faced by retailers in the supermarket industry.

6.1. Supermarket data

Through a collaboration with Oracle Retail, we had access to actual data from supermarkets. We received large data sets from several categories of products in supermarkets. In this paper, we use aggregate weekly sales data for several brands of coffee over the period 2009-2011. We calibrate the demand models from the data, and observe that our estimated demand models yield a good out-of-sample forecast accuracy. In particular, the out-of-sample R^2 and $MAPE$ are between 0.85 and 0.96, and 0.1 and 0.3 respectively. In our tests, we use $c_t = 0.4; \forall t$, $T = 35$, $q^0 = 1$ and $q^Q = 0.4$.

More details on the data and on demand estimation can be found in Cohen et al. (2017). All the experiments were run using a server with 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 we have shown in Table 1, solving the dynamic program based on the graphical representation from Section 3 scales linearly with the time horizon T. However, the runtime is exponential with respect to the memory parameter m. Our goal is to test the running time with respect to the different problem parameters on practical instances in order to understand the limitations of our approach. As we discussed in Section 3.2, we expect to observe asymptotic runtimes of $O(T|\mathcal{Q}_p|^m)$. In this section, we consider the following log-log demand function estimated from data:

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

where the coefficients $a_t,\ t=1,\dots,35$ represent the multiplicative seasonality effects and are between 759.4 and 975.7. In Figure 4, we plot the runtime as a function of the size of the price ladder (for $T=35$) and as a function of the number of time periods (for 8 prices). We consider three different values of the memory parameter ($m=2,3,4$), and impose a timeout of 2000 seconds. Typically, for most of the item categories in supermarkets, the memory parameter is between 0 and 4. For instance, for several coffee items, the memory parameter was found to be equal to 2. The planning horizon can be between 10 and 52 weeks, and the number of elements in the price ladder vary between 2 and 20.

One can see that for a demand model with $m=2$, 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 some demand parameters in order 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. When $m=2$, for a single item, we can solve the problem in less than a second. 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 the the reference price approximation introduced in Section 4 that allows us to compute a near-optimal solution within seconds, as we show in the 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 $|\mathcal{Q}_p|=8$, $T=35$, and vary the parameters L and

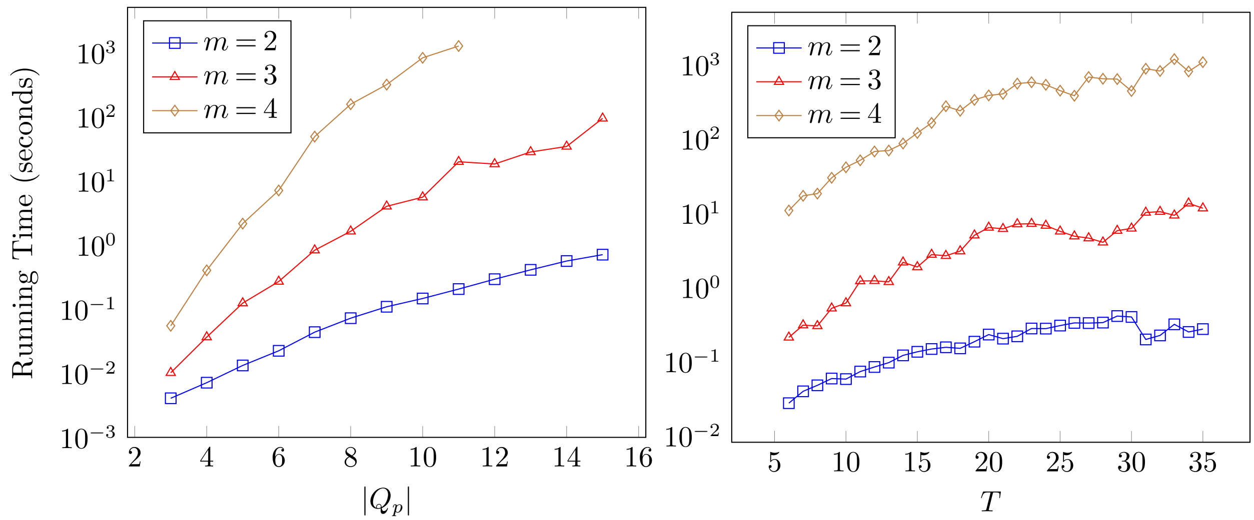

Figure 4 Runtimes of the dynamic program with different memory parameters as a function of the size of the price ladder (left) and the number of time periods (right).

S. Note that adding the L constraint to the formulation increases the number of possible states multiplicatively by a factor of L. The plot in Figure 5 is consistent with this analytical result (recall that the y axis is in logarithmic scale). In addition, incorporating the no-touch constraint and varying the value of S has a relatively flat effect on the runtime, which is consistent with the asymptotic runtime of $O(T^2|\mathcal{Q}_p|^{\lceil \frac{m}{S} \rceil})$. We conclude that for large memories (i.e., $m \geq 4$), solving the dynamic program is not a viable option in this context. Fortunately, in practice, a significant number of items in supermarkets admit a small memory parameter. However, when m is large, one can consider the approximation based on the reference price model, discussed next.

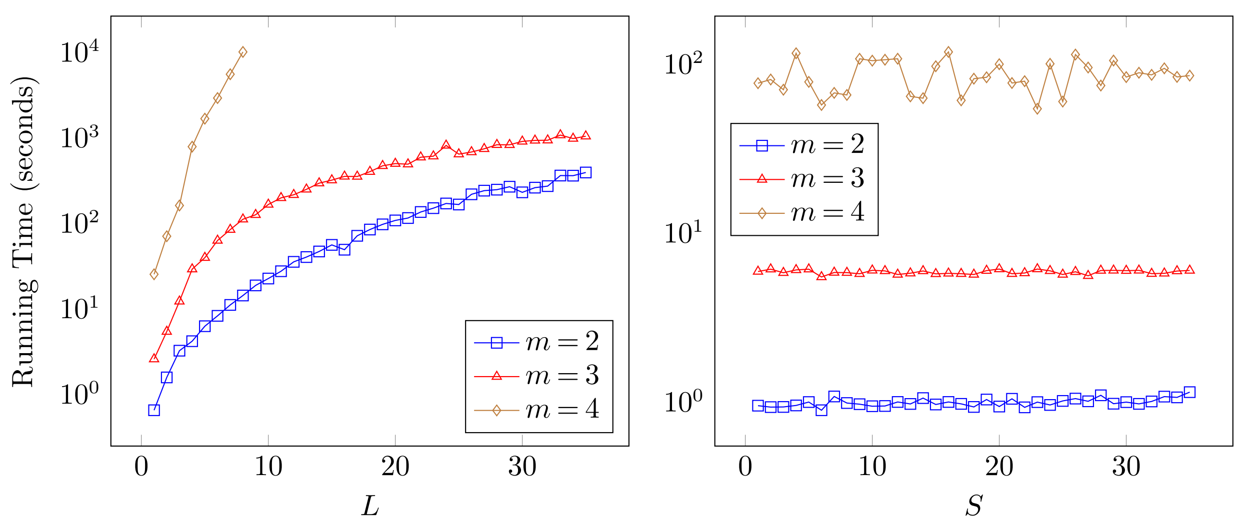

Figure 5 Runtimes of the dynamic program over 35 time periods varying the memory parameter from 2 to 4 as a function of L (left) and S (right). The size of the price ladder is 6.

6.3. Discrete reference price model

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

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

where the coefficients a_t, $t = 1, \ldots, 35$ 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 granularity ϵ, the size of the price ladder and the number of time periods. One can see that the discrete reference price model can be solved efficiently even with a large set of prices and a granular reference price ladder ($\epsilon = 0.025$). More precisely, all the instances we tested solve in less than 0.1 second.

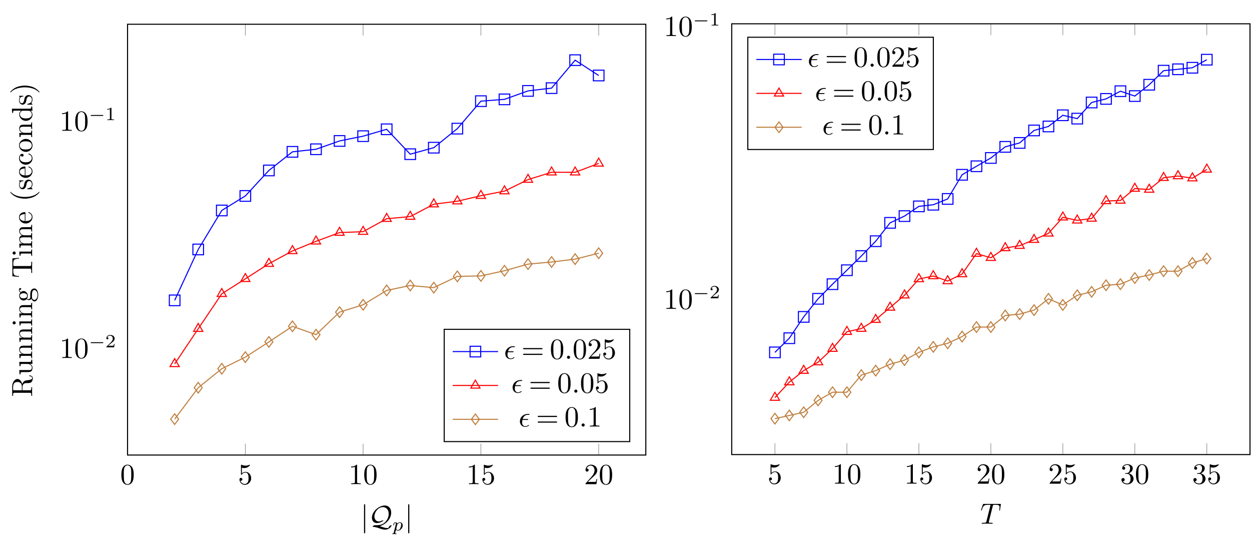

Figure 6 Runtimes of the dynamic program with reference price when varying the accuracy level ϵ as a function of the size of the price ladder (left) and the number of time periods (right). The size of the price ladder is 8 and $\theta = 0.4$.

We next test the quality of the approximation described in Section 4.2. Our goal is to approximate a given demand function with a large memory parameter (see equation (13)) by a discrete reference price demand model. To this end, we consider 100 randomly generated instances of linear demand models in the form of (13). For each instance, we assume $m = T = 10$, and $|\mathcal{Q}_p| = 2$ (i.e., the regular price $q^0 = 1$, and the promotional price $q^1 = 0.7$).

We consider the following linear demand model:

$$d_t = a_t - b_0 p_t + b_1 p_{t-1} + b_2 p_{t-2} + \ldots b_{10} p_{t-10},$$

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

Note that for each instance, we obtain a linear demand function with a large memory parameter. We then follow the procedure described in Section 4.2 so as to fit a discrete reference price model in order to approximate the true demand function by finding the value of $\widetilde{\theta}_{LS}$ for each instance. We next solve the unconstrained dynamic program and compute the optimal prices for the discrete reference price model using reference price ladder with discretization $\epsilon = 0.001$. For each instance, we compare the total profit induced by these prices relative to the optimal prices obtained by solving the exact (non-tractable) formulation with the true demand. We obtained that for most of the instances, the approximated model based on the discrete reference price model yields a near optimal solution. In particular, in this example, the minimum is 77.6%, the 25 percentile is 97.7%, the median is 99.4% and the 75 percentile and the maximum are 100% relative to the optimal profit. Consequently, for demand models with a large memory parameter, the approximation method developed in Section 4.2 allows to solve the problem in milliseconds while finding a near-optimal solution.

6.4. Multiple items

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

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

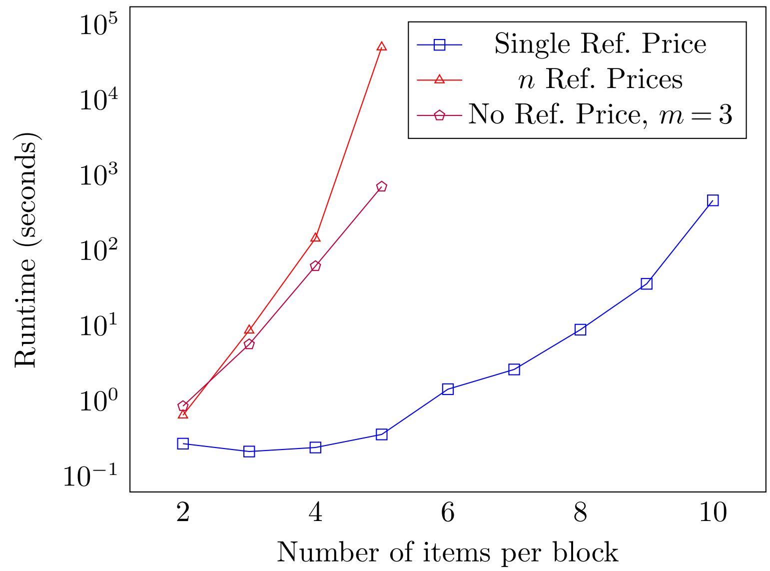

Figure 7 **Runtimes for an instance with 100 items and 10 time periods using** $\epsilon = 0.1$**.**

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

The reference prices follow: $r_t^i = \text{round}[0.6p_{t-1}^i + 0.4r_{t-1}^i]$ and $r_0^i = q^0 = 1$, and $r_t^V = \text{round}[0.6\frac{1}{100}\sum_{i=1}^{100} p_{t-1}^i + 0.4r_{t-1}^V]$ and $r_0^V = q^0 = 1$ (with a discretization parameter of $\epsilon = 0.1$). Note that one can update the virtual reference price by either the average, the minimum or the maximum of the prices p_{t-1}^i, as all the items are identical. The results are presented in Figure 7. For practical considerations, our target is to solve the problem for 100 items in a few minutes. Therefore, one can see that the only viable method is the model with a single virtual reference price from Section 5.2. This solution approach scales significantly better relative to the other two methods and as a result, allows to solve the problem for larger blocks of items in reasonable time frames. Note that in practice, one could achieve significantly better runtimes by optimizing each block in parallel, which would speed up the runtime multiplicatively by the number of blocks.

7. Conclusion

In this paper, we study the dynamic 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 on the pricing policy. We introduce a graphical representation of the problem that translates the profit maximization into solving a maximum weighted path problem on a layered graph. We further show that the problem is NP-hard by reducing from the Traveling Salesman Problem. In addition, we develop a dynamic programming solution method with a runtime of $O(T|\mathcal{Q}_p|^{m+1})$, where T is the number of time periods, m the memory parameter that captures the number of past prices that affect the current demand, and $\mathcal{Q}_p$ is the price ladder.

For large values of m, this solution approach is not practical, as it may take several minutes (or hours) to solve a single instance. This motivates us to propose a good approximation method that can solve the problem in reasonable time frames. We first consider the case where the demand follows the commonly used reference price model, i.e., the demand depends on the current price and on a reference price based on historical prices. In this case, we introduce the discrete reference price model that restricts the reference prices to lie in a discrete price ladder under the premise that customers usually do not form a reference price with high precision. Under this model, we show how to solve the problem efficiently. Second, we consider several general demand functions (linear, log-log and log-linear) with a potentially large memory parameter. We propose a procedure to approximate these general demand functions by using the discrete reference price model. This allows us to solve instances with a large memory parameter in less than a millisecond, while having a guarantee on the quality of the approximation.

Finally, we explore the setting where the retailer solves the dynamic pricing problem for multiple items. In this case, we propose two solution approaches that are inspired by the discrete reference price model. In particular, we assume that consumers form a reference price either for each product separately, or a joint virtual reference price for the entire product category. To increase the tractability of our approach, we introduce the notion of blocks and organize items into smaller clusters such that the cross-item interactions across blocks is negligible. Although this reduces the running time of the promotion optimization problem significantly for each block, it becomes more challenging to impose global constraints on the promotions across the different blocks. While drawing ideas from the combinatorial optimization literature, we limit the total number of promotions across blocks by solving a multi-choice knapsack problem. We also propose methods to handle price-ordering and exclusivity constraints that are often important in retail. We finally apply our solution approaches using demand models calibrated by supermarket real data, and show that we can solve realistic size instances in a few minutes.

Acknowledgments

We would like to thank Oracle Corporation for their generous support through an ERO grant that made this work possible. In addition, we would like to acknowledge Kiran Panchamgam who gave us invaluable insights on this project. The second author would like to 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.

References

Ahuja RK, Magnanti TL, Orlin JB (1988) Network flows. Technical report, DTIC Document.

Ailawadi KL, Gedenk K, Lutzky C, Neslin SA (2007) Decomposition of the sales impact of promotion-induced stockpiling. *Journal of Marketing Research* 450–467.

Blattberg RC, Briesch R, Fox EJ (1995) How promotions work. *Marketing science* 14(3_supplement):G122–G132.

Blattberg RC, Neslin SA (1990) *Sales promotion: Concepts, Methods, and Strategies* (Prentice Hall).

Chen X, Hu P, Hu Z (2016) Efficient algorithms for dynamic pricing problem with reference price effect Submitted to Management Science.

Cohen MC, Leung NZ, Panchamgam K, Perakis G, Smith A (2017) The impact of linear optimization on promotion planning Forthcoming in Operations Research.

Cooper LG, Baron P, Levy W, Swisher M, Gogos P (1999) Promocast: A new forecasting method for promotion planning. *Marketing Science* 18(3):301–316.

Corsten D, Gruen T (2004) Stock-outs cause walkouts. *Harvard Business Review* 82(5):26–28.

Dudziński K, Walukiewicz S (1987) Exact methods for the knapsack problem and its generalizations. *European Journal of Operational Research* 28(1):3–21.

Ellickson PB, Misra S (2008) Supermarket pricing strategies. *Marketing science* 27(5):811–828.

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

Foekens EW, Leeflang PSH, Wittink DR (1998) Varying parameter models to accommodate dynamic promotion effects. *Journal of Econometrics* 89(1):249–268.

Garey MR, Johnson DS (1979) *Computers and Intractability: A Guide to the Theory of NP-completeness* (W.H. Freeman and Company).

Gedenk K, Neslin SA, Ailawadi KL (2006) Sales promotion. *Retailing in the 21st Century*, 345–359 (Springer).

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

Heerde HJV, Leeflang PSH, Wittink DR (2000) The estimation of pre- and post- promotion dips with store-level scanner data. *Journal of Marketing Research* 37:383–395.

Herer YT, Tzur M (2001) The dynamic trans-shipment problem. *Naval Research Logistics* 48(5):386–408.

Karp RM (1972) *Reducibility Among Combinatorial Problems* (Springer).

Kopalle PK, Rao AG, Assuncao JL (1996) Asymmetric reference price effects and dynamic pricing policies. *Marketing Science* 15(1):60–85.

Lal R, Rao R (1997) Supermarket competition: The case of every day low pricing. *Marketing science* 16(1):60–80.

Macé S, Neslin SA (2004) The determinants of pre- and postpromotion dips in sales of frequently purchased goods. *Journal of Marketing Research* 41:339–350.

Mela CF, Jedidi K, Bowman D (1998) The long-term impact of promotions on consumer stockpiling behavior. *Journal of Marketing Research* 250–262.

Morávek J (1970) A note upon minimal path problem. *Journal of Mathematical Analysis and Applications* 30(3):702–717.

Nijs VR, Dekimpe MG, Steenkamps JBE, Hanssens DM (2001) The category-demand effects of price promotions. *Marketing science* 20(1):1–22.

Pisinger D (1995) A minimal algorithm for the multiple-choice knapsack problem. *European Journal of Operational Research* 83(2):394–410.

Popescu I, Wu Y (2007) Dynamic pricing strategies with reference effects. *Operations Research* 55(1):98–112.

Porteus EL (1990) Stochastic inventory theory. *Handbooks in OR & MS* 2:605–652.

Schrijver A (2003) *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24 (Springer Science & Business Media).

Subramanian S, Sherali HD (2010) A fractional programming approach for retail category price optimization. *Journal of Global Optimization* 48(2):263–277.

Talluri KT, van Ryzin GJ (2005) *The Theory and Practice of Revenue Management* (Springer).

Vazirani VV (2013) *Approximation algorithms* (Springer Science & Business Media).

Wang Z (2016) Technical note – intertemporal price discrimination via reference price effects. *Operations Research* 64(2):290–296.

Zhao W, Zheng Y (2000) Optimal dynamic pricing for perishable assets with nonhomogeneous demand. *Management science* 46(3):375–388.

A. Equivalent view using Dynamic Programming

A.1. General model for single item.

For the single item profit maximization problem under a general demand model, we present equivalent dynamic programming formulations to solve the problem.

1. **The unconstrained problem.**

We compute a recursive function $Y(\mathbf{x},t)$ for the best possible profit up to time t such that the price during time periods $\{t-m+1,\dots,t\}$ is $\mathbf{x}\in\mathcal{Q}_p^m$:

$$Y(\mathbf{x},t)=\max_{\mathbf{q}\in\mathcal{Q}_p^m:q_{[2,m]}=x_{[1,m-1]}} Y(\mathbf{q},t-1)+(x_t-c_t)\cdot d_t(q_1,\dots,q_m,x_t).$$

The optimal profit is then $\max\limits_{\mathbf{x}\in\mathcal{Q}_p^m} Y(x,T)$. The state space of this dynamic program is of size $T|\mathcal{Q}_p|^m$. Since we perform $|\mathcal{Q}_p|$ comparisons at each state, the time complexity is $O(T|\mathcal{Q}_p|^{m+1})$.

2. **Limited number of price changes (say $\leq L$) in the planning horizon.**

We compute a recursive function $Y(\mathbf{x},l,t)$ for the best possible profit using l price changes up to time t such that the price during time periods $\{t-m+1,\dots,t\}$ is $\mathbf{x}\in\mathcal{Q}_p^m$. Let the set of price vectors with exactly l price changes for the time period $[1,t]$ be denoted as $\mathcal{P}_l^t$. The recurrence for the dynamic program is given by:

$$\begin{aligned}Y(\mathbf{x},l,t)&=\max_{\mathbf{p}\in\mathcal{P}_l^t:\mathbf{p_t}=\mathbf{x}}\sum_{k=1}^{t}\big(p_k-c_k\big)\cdot d_k(p_k,\dots,p_{k-m}),\\&=\begin{cases}\max\limits_{\mathbf{q}\in\mathcal{Q}_p^m:q_{[2,m]}=x_{[1,t-1]}} Y(\mathbf{q},l,t-1)+(x_t-c_t)\cdot d_t(q_1,\dots,q_m,x_t), & \text{if } x_t=q_m,\\ \max\limits_{\mathbf{q}\in\mathcal{Q}_p^m:q_{[2,m]}=x_{[1,t-1]}} Y(\mathbf{q},l-1,t-1)+(x_t-c_t)\cdot d_t(q_1,\dots,q_m,x_t), & \text{if } x_t\neq q_m.\end{cases}\end{aligned}$$

The optimal profit under limited number of price changes is then $\max\limits_{\mathbf{x}\in\mathcal{Q}_p^m,\,l\leq L} Y(\mathbf{x},l,T)$. The state space of this dynamic program is of size $(L+1)|\mathcal{Q}_p|^mT$, and the running time complexity is $O(LT|\mathcal{Q}_p|^{m+1})$.

3. ***No-touch*** **constraints, i.e., at least S time periods between two price changes.**

We compute a recursive function $Y(\mathbf{x},s,t)$ for the best possible profit with a price change allowed after at least s time periods (i.e., the last price variation must have occurred before $t-s$ time periods), such that the price during time periods $\{t-m+1,\dots,t\}$ is $\mathbf{x}\in\mathcal{Q}_p^m$. Let the set of price vectors with at least S time periods between any consecutive price changes for the time period $[1,t]$ be denoted as $\mathcal{P}_s^t$. The recurrence for the dynamic program is given by:

$$\begin{aligned}Y(\mathbf{x},s,t)&=\max_{\mathbf{p}\in\mathcal{P}_s^t:\mathbf{p_t}=\mathbf{x}}\sum_{k=1}^{t}\big(p_k-c_k\big)\cdot d_k(p_k,\dots,p_{k-m}),\\&=\begin{cases}\max\limits_{\mathbf{q}\in\mathcal{Q}_p^m:q_{[2,m]}=x_{[1,m-1]}} Y(\mathbf{q},s+1,t-1)+(x_m-c_t)\cdot d_t(q_1,\dots,q_m,x_m), & \text{if } x_m=q_m, 0<s<S,\\ \max\limits_{\mathbf{q}\in\mathcal{Q}_p^m:q_{[2,m]}=x_{[1,m-1]}} Y(\mathbf{q},0,t-1)+(x_m-c_t)\cdot d_t(q_1,\dots,q_m,x_m), & \text{if } x_m\neq q_m, s=S,\\ \max\Big\{\max\limits_{\mathbf{q}\in\mathcal{Q}_p^m:q_{[2,m]}=x_{[1,m-1]}} Y(\mathbf{q},0,t-1)+(x_m-c_t)\cdot d_t(q_1,\dots,q_m,x_m), & \\ \max\limits_{\mathbf{q}\in\mathcal{Q}_p^m:q_{[2,m]}=x_{[1,m-1]}} Y(\mathbf{q},1,t-1)+(x_m-c_t)\cdot d_t(q_1,\dots,q_m,x_m)\Big\}, & \text{if } x_m=q_m, s=0,\\ -\infty & \text{otherwise.}\end{cases}\end{aligned}$$

The optimal profit under no-touch price constraints is then $\max_{\mathbf{x}\in\mathcal{Q}_p^m,\, s\leq S} Y(\mathbf{x},s,T)$. The state space of this dynamic program is of size $O(T|\mathcal{Q}_p|^{\lceil \frac{m}{s}\rceil})$, and the running time complexity is $O(T|\mathcal{Q}_p|^{\lceil \frac{m}{s}\rceil}\max(|\mathcal{Q}_p|,S))$.

A.2. Discrete reference price model

For the profit maximization problem with a single item under the discrete reference price model, we show equivalent dynamic programming formulations to solve the problem when the reference price is discretized up to ϵ.

1. **The unconstrained problem.**

Recall that we denote the nodes in the graph by $(\mathbf{x},t)$, where $\mathbf{x}=(x_p,x_r)\in\mathcal{Q}_p\times\mathcal{Q}_r$, i.e., all possible pairs of prices and reference prices. We compute a function $F(\mathbf{x},t)$ for the best possible profit up to time t such that the price at time t is x_p and the corresponding reference price is x_r:

$$F(\mathbf{x},t)=\max_{\mathbf{y}\in\mathcal{Q}_p\times\mathcal{Q}_r:|y_p(1-\theta)+\theta y_r-x_r|\leq\epsilon} F(\mathbf{y},t-1)+(x_p-c_t)\cdot(a-\beta^0 x_p-\phi(x_p-x_r)).$$

The optimal profit under the discrete reference price model is then $\max_{\mathbf{x}\in\mathcal{Q}_p\times\mathcal{Q}_r} F(\mathbf{x},T)$. The state space of this dynamic program is of size $T|\mathcal{Q}_p||\mathcal{Q}_r|$, and the running time complexity is $O(T|\mathcal{Q}_p|^2\mathcal{Q}_r)$.

2. **Limited number of price changes.**

We compute a function $F(\mathbf{x},l,t)$ for the best possible profit up to time t such that the price at time t is x_p and the corresponding reference price is x_r with l price changes up to time t:

$$F(\mathbf{x},l,t)=\begin{cases}\max\limits_{\mathbf{y}\in\mathcal{Q}_p\times\mathcal{Q}_r:|y_p(1-\theta)+\theta y_r-x_r|\leq\epsilon} F(\mathbf{y},l,t-1)+(x_p-c_t)\cdot(a-\beta^0 x_p-\phi(x_p-x_r)), & \text{if } x_p=y_p,\\ \max\limits_{\mathbf{y}\in\mathcal{Q}_p\times\mathcal{Q}_r:|y_p(1-\theta)+\theta y_r-x_r|\leq\epsilon} F(\mathbf{y},l-1,t-1)+(x_p-c_t)\cdot(a-\beta^0 x_p-\phi(x_p-x_r)), & \text{if } x_p\neq q_{p-1}.\end{cases}$$

The optimal profit under limited number of price changes is then $\max_{\mathbf{x}\in\mathcal{Q}_p^m,\, l\leq L} Y(\mathbf{x},l,T)$. The state space of this dynamic program is of size $(L+1)T|\mathcal{Q}_p||\mathcal{Q}_r|$, and the running time complexity is $O((L+1)T|\mathcal{Q}_p|^2|\mathcal{Q}_r|)$.

3. ***No-touch* constraints, i.e., at least S time periods between two price changes.**

We compute a function $F(\mathbf{x},s,t)$ for the best possible profit up to time t such that the price at time t is x_p and the corresponding reference price is x_r with the next price change allowed after $s\in[0,S]$ time periods:

$$F(\mathbf{x},s,t)=\begin{cases}\max\limits_{\mathbf{y}\in\mathcal{Q}_p\times\mathcal{Q}_r:|y_p(1-\theta)+\theta y_r-x_r|\leq\epsilon} F(\mathbf{y},t-1,s+1)+(x_p-c_t)\cdot(a-\beta^0 x_p-\phi(x_p-x_r)), & \text{if } x_p=y_p, 0<s<S,\\ \max\limits_{\mathbf{y}\in\mathcal{Q}_p\times\mathcal{Q}_r:|y_p(1-\theta)+\theta y_r-x_r|\leq\epsilon} F(\mathbf{y},0,t-1)+(x_p-c_t)\cdot(a-\beta^0 x_p-\phi(x_p-x_r)), & \text{if } x_p\neq y_p, s=S,\\ \max\big\{\max\limits_{\mathbf{y}\in\mathcal{Q}_p\times\mathcal{Q}_r:|y_p(1-\theta)+\theta y_r-x_r|\leq\epsilon} F(\mathbf{y},t-1,1)+(x_p-c_t)\cdot(a-\beta^0 x_p-\phi(x_p-x_r)), & \\ \max\limits_{\mathbf{y}\in\mathcal{Q}_p\times\mathcal{Q}_r:|y_p(1-\theta)+\theta y_r-x_r|\leq\epsilon} F(\mathbf{y},t-1,0)+(x_p-c_t)\cdot(a-\beta^0 x_p-\phi(x_p-x_r))\big\}, & \text{if } x_p=y_p, s=0,\\ -\infty & \text{otherwise.}\end{cases}$$

The optimal profit under limited number of price changes is then $\max_{\mathbf{x}\in\mathcal{Q}_p^m,\, l\leq L} Y(\mathbf{x},l,T)$. The state space of this dynamic program is of size $O(TS|\mathcal{Q}_p||\mathcal{Q}_r|)$, and the running time complexity is $O(T|\mathcal{Q}_p||\mathcal{Q}_r|\max(|\mathcal{Q}_p|,S))$.

B. Log-log and log-linear demand models

In this section, we study the log-log and log-linear demand 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 demand model in (27). In this case, the reference prices are maintained and updated in the log-space as follows:

$$\log r_t = (1-\theta)\log p_{t-1} + \theta \log r_{t-1}, \tag{22}$$

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 (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:

$$\log d_t(p_t, r_t) = f_t(p_t) + g(\log p_t - \log r_t), \tag{23}$$

$$\log d_t(p_t, r_t) = f_t(p_t) + g(p_t - r_t). \tag{24}$$

We do not impose any assumption on the function $f_t(\cdot)$, 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):

$$\log \hat{r}_t = \text{round}\Big[(1-\theta)\log p_{t-1} + \theta \log r_{t-1}\Big]. \tag{25}$$

More precisely, we round to the nearest element in the set $\mathcal{Q}_{log} = \{r^0 > r^1 > \cdots > r^n > \ldots > r^N\}$, where $\log r^i = \log r^{i+1} + \epsilon$ for all $i = 0, \ldots, N-1$. 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 the continuous reference price.

Proposition 5. *Consider the continuous log-reference price model in* (22) *and the proposed discrete log-reference price model by rounding the log-reference price to the nearest value in the set* $\mathcal{Q}_{log}$ *with precision* $\epsilon > 0$ *as in* (25)*. Then, the difference in the reference prices at time* t *can be bounded by:*

$$|\hat{r}_t - r_t| \leq \frac{1-\theta^{t-1}}{1-\theta} r_t \epsilon, \tag{26}$$

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

Proof. For the first time period, we have: $\log \hat{r}_1 = \text{round}\Big[(1-\theta)\log p_0 + \theta \log r_0\Big]$. 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: $\log \hat{r}_{k+1} = \text{round}\Big[(1-\theta)\log p_k + \theta \log \hat{r}_k\Big]$. Then, we obtain:

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

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 terms of the demand function and the total profit.

Corollary 2. *Consider the log-log demand model in* (23) *and the discrete log-reference price model by rounding to the nearest value in the set $\mathcal{Q}_{log}$ with precision $\epsilon > 0$. Then, the demand value at time t, and the difference in the total profit can be bounded by:*

$$\hat{d}_t = d_t \exp(g(\log \frac{p_t}{r_t}) - g(\log \frac{p_t}{\hat{r}_t})) \leq d_t \exp(G \log \frac{\hat{r}_t}{r_t}) \approx d_t \exp(G \log(1 \pm \sum_{u=0}^{t} \theta^u \epsilon)) \leq d_t \exp(G\epsilon \sum_{u=0}^{t} \theta^u)),$$

$$\left|\hat{\Pi} - \Pi\right| \leq (\max_t d_t)\ T(q^0 - c_{min}) G\epsilon/(1-\theta),$$

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

Finally, we extend the treatment for the log-linear reference price model in (24). Note that in this case, we use the traditional reference price model as explained in Section 4.

Corollary 3. *Consider the log-linear demand model in* (24) *and the discrete reference price model by rounding to the nearest value in the set $\mathcal{Q}_r$ with precision $\epsilon > 0$. Then, the demand value at time t, and the difference in the total profit can be bounded by:*

$$\hat{d}_t = d_t \exp(g(p_t - r_t) - g(p_t - \hat{r}_t)) \leq d_t \exp(G(\hat{r}_t - r_t)) \leq d_t \exp(G\epsilon(1 + \sum_{u=0}^{t} \theta^u)),$$

$$\left|\hat{\Pi} - \Pi\right| \leq (\max_t d_t)\ T(q^0 - c_{min}) G\epsilon \frac{2-\theta}{1-\theta},$$

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

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

Approximating the log-log and log-linear demand models. Similarly as in Section 4.2, for 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 tends to the continuous model when the discretization parameter ϵ approaches 0). We next extend the results of Section 4.2 for the commonly used log-log and log-linear demand models, given by:

$$\log d_t(\mathbf{p_t}) = f_t(p_t) + \beta^1 \log p_{t-1} + \cdots + \beta^T \log p_{t-T}, \tag{27}$$

$$\log d_t(\mathbf{p_t}) = f_t(p_t) + \beta^1 p_{t-1} + \cdots + \beta^T p_{t-T}. \tag{28}$$

This type of models is popular in retail applications such as supermarkets. The parameters $\beta^1, \ldots, \beta^T$ as well as 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: $\beta^1 \geq \ldots \beta^T \geq 0$.

Using the continuous log-linear reference price and the log-log reference price models, we approximate equations (27) and (28) as follows:

$$\log \widetilde{d}_t(p_t, \widetilde{r}_t) = f_t(p_t) + g(\log p_t - \log \widetilde{r}_t) = f_t(p_t) + \widetilde{\phi} \sum_{k=1}^{T} (1 - \widetilde{\theta}) \widetilde{\theta}^{k-1} \log p_{t-k}, \tag{29}$$

$$\log \widetilde{d}_t(p_t, \widetilde{r}_t) = f_t(p_t) + g(p_t - \widetilde{r}_t) = f_t(p_t) + \widetilde{\phi} \sum_{k=1}^{T} (1 - \widetilde{\theta}) \widetilde{\theta}^{k-1} p_{t-k}. \tag{30}$$

Following the exact same procedure as in Section 4.2, we impose that $\widetilde{\phi}(1 - \widetilde{\theta}) = \beta^1$ in order to match perfectly the coefficient of p_{t-1}. Then, the design parameter $\widetilde{\theta}$ can be computed as the minimum, the maximum or the least-squares fit. Consequently, we obtain the same result as in Proposition 4 for both the log-log and log-linear demand models.