informs
http://pubsonline.informs.org/journal/mnsc/

MANAGEMENT SCIENCE
Vol. 65, No. 1, January 2019, pp. 50–70
ISSN 0025-1909 (print), ISSN 1526-5501 (online)

Scheduling Promotion Vehicles to Boost Profits

Lennart Baardman,[a] Maxime C. Cohen,[b] Kiran Panchamgam,[c] Georgia Perakis,[d] Danny Segev[e]

[a] Operations Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139; [b] Stern School of Business, New York University, New York, New York 10012; [c] Oracle Retail Global Business Unit, Burlington, Massachusetts 01803; [d] Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139; [e] Department of Statistics, University of Haifa, Haifa 31905, Israel

Contact: baardman@mit.edu, http://orcid.org/0000-0001-7602-6571 (LB); maxime.cohen@stern.nyu.edu (MCC); kiran.panchamgam@oracle.com (KP); georgiap@mit.edu, http://orcid.org/0000-0002-0888-9030 (GP); segevd@stat.haifa.ac.il (DS)

Received: November 5, 2016
Revised: May 1, 2017; August 8, 2017
Accepted: August 16, 2017
Published Online in Articles in Advance: April 6, 2018

https://doi.org/10.1287/mnsc.2017.2926

Abstract. In addition to setting price discounts, retailers need to decide how to schedule promotion vehicles, such as flyers and TV commercials. Unlike the promotion pricing problem that received great attention from both academics and practitioners, the promotion vehicle scheduling problem was largely overlooked, and our goal is to study this problem both theoretically and in practice. We model the problem of scheduling promotion vehicles to maximize profits as a nonlinear bipartite matching-type problem, where promotion vehicles should be assigned to time periods, subject to capacity constraints. Our modeling approach is motivated and calibrated using actual data in collaboration with Oracle Retail, leading us to introduce and study a class of models for which the boost effects of promotion vehicles on demand are multiplicative. From a technical perspective, we prove that the general setting considered is computationally intractable. Nevertheless, we develop approximation algorithms and propose a compact integer programming formulation. In particular, we show how to obtain a $(1-\epsilon)$-approximation using an integer program of polynomial size, and investigate the performance of a greedy procedure, both analytically and computationally. We also discuss an extension that includes cross-term effects to capture the cannibalization aspect of using several vehicles simultaneously. From a practical perspective, we test our methods on actual data through a case study, and quantify the impact of our models. Under our model assumptions and for a particular item considered in our case study, we show that a rigorous optimization approach to the promotion vehicle scheduling problem allows the retailer to increase its profit by 2% to 9%.

History: Accepted by Yinyu Ye, optimization.
Funding: This work was supported by the Oracle Corporation [ERO grant] and the National Science Foundation [Grants CMMI-1162034 and CMMI-1563343]. D. Segev's research on this project was supported by the Israel Science Foundation [Grant 148/16].
Supplemental Material: The online appendix is available at https://doi.org/10.1287/mnsc.2017.2926.

Keywords: retail operations • promotion optimization • integer programming • approximation algorithms

1. Introduction

Retailers use sales promotions to attract new customers, increase sales, and encourage existing customers to switch brands, among other reasons. Focusing attention on the supermarket industry, sales promotions are often used to generate higher profits, typically through price reductions, placing products at the end of an aisle, dedicating an in-aisle display to advertise some products, sending out flyers, and broadcasting commercials. The first sales promotion in this list is a price discount, which is a temporary reduction in the product's price. The additional examples listed above are called promotion vehicles—i.e., various methods of communicating to customers that certain products are worth purchasing. Given that the inherent purpose of these vehicles is to boost profits, one needs to sensibly determine which vehicles to assign in which periods throughout the selling season.

Deciding the right time for sales promotions, which price reductions to offer, and which promotion vehicles to use is a fundamental problem of interest to supermarket managers. Illustrating the potential impact, more than 50% of many brands' sales occur during sales promotions (see chap. 12 of Blattberg and Neslin 1990). However, the effective scheduling of sales promotions is a complex and challenging problem, both theoretically and in practice. Currently, many supermarket chains are still making decisions based on intuition, past experience, and heuristic arguments. Consequently, this provides great opportunities to apply advanced data-driven optimization techniques to improve the planning process and to gain managerial insights. One of the key practical questions that motivates this paper is: How much money does the retailer leave on the table by using current promotion vehicle assignment policies (based mainly on intuition and heuristics) relative to what can be attained by developing data-driven optimization tools?

The analysis conducted in this paper provides concrete evidence that promotions can be a key driver for increasing profits. In particular, scheduling sales promotions effectively by using the right promotion vehicles at the right times can lead to a significant profit improvement. As explained later on, we validate the impact of our model by using sales data from a large supermarket retailer. Based on our modeling approach and algorithmic methods, calibrated with real data, we observe that optimizing vehicle assignments throughout the selling season yields a profit improvement of between 2% and 9% for the retailer. To better understand the significance of this finding, it is worth mentioning that a report published by the Community Development Financial Institutions (CDFI) Fund indicates that the average profit margin for the supermarket industry was only 1.9% in 2010 (see The Reinvestment Fund 2011).

1.1. Informal Modeling Approach

We consider the problem faced by a supermarket manager, who seeks to assign promotion vehicles over a finite planning horizon so as to maximize profit. To the best of our knowledge, modeling and formulating the promotion vehicle scheduling problem was not considered in the literature before. In this paper, motivated by supermarket data, we propose an analytical model for this problem as well as efficient approximation algorithms that yield provably good scheduling policies. Our model allows retailers to improve their decisions on the scheduling of promotion vehicles by relying on a support decision tool calibrated with historical data. More precisely, we formulate the problem as a bipartite matching-type problem, where promotion vehicles should be assigned to time periods, subject to capacity constraints. What makes our setting significantly different from existing optimization problems in this context relates to the form of the objective function. Here, we are not maximizing a linear function, but instead, the contribution of using several vehicles at any given time period has a multiplicative effect, introducing a host of computational obstacles in optimizing this function. In fact, as explained in Section 4, we try to fit our data to both additive (linear) and multiplicative (nonlinear) models, and consistently observe that the multiplicative model provides a better fit to the data. In this model, each time period is associated with some nominal profit (i.e., without accounting for promotion vehicles), which can be boosted based on the subset of promotion vehicles assigned to that period. The boosting factor may also account for cannibalization effects, from using multiple promotion vehicles simultaneously. In practice, retailers often refer to this cannibalization effect as overlapping promotion vehicles (e.g., the boost in demand from simultaneously using two vehicles is lower than using these two vehicles independently). Finally, our formulation admits two types of business rules as constraints: (i) imposing a limit on the number of times each promotion vehicle can be assigned throughout the planning horizon; and (ii) imposing a limit on the number of promotion vehicles that can be assigned to each time period. In Sections 3 and 4.3, we present a formal description of this model, its main business rules, and the relevant mathematical notation.

1.2. Contributions

From a theoretical perspective, the main contribution of this paper lies in introducing and studying a new matching-type optimization problem. To the best of our knowledge, this problem has not been studied before. As explained in greater detail below, we provide complexity results, devise efficient approximation algorithms, and propose a compact integer programming formulation. In addition, since this research was conducted in collaboration with Oracle Retail, a particular emphasis has been put on the real-world applicability of our methods. With this goal in mind, we pay special attention to testing the validity of our models and to measuring their impact using actual data. We next briefly summarize our main contributions.

- *Modeling the promotion vehicle scheduling problem.* Motivated by real-world retail environments, we introduce a new class of models for scheduling promotion vehicles, where the boost effects of vehicles on demand are multiplicative. This class of models is easy to estimate from data and yields a good forecast accuracy. Our modeling approach and its empirical motivation are discussed in Sections 3 and 4, whereas the resulting optimization problem is formally described in Section 4.3.
- *Complexity results.* We show that, unlike standard (linear) matching problems, the introduction of multiplicative boost terms into our formulation renders the problem NP-hard. Moreover, we prove that our problem cannot be efficiently approximated within some given constant by relating it to the task of detecting large independent sets in regular graphs. This hardness result is presented in Section 5.1.
- *Approximation algorithms.* We develop three different approaches for computing provably good solutions. The first approach consists of an efficient greedy algorithm, attaining an approximation ratio of $\Delta + 1$, where Δ stands for the maximum number of vehicles that can be assigned to any time period (see Section 5.2). Our second approach shows that by losing an ϵ-factor in optimality, our problem can be formulated as a polynomial-size integer program (see Section 5.3). Finally, we also show that the special case when the vehicle boosts are uniform admits a polynomial-time approximation scheme (PTAS) (see Online Appendix D).

• *Extension to cross terms.* We study an extension of our model by considering cross-term interactions between pairs of vehicles. In particular, we prove a stronger hardness result for this extension, and show that our integer programming approach is flexible enough to incorporate these terms into the formulation, while still guaranteeing ϵ-optimality (see Appendix B). We also test this model computationally in Section 6.2, by comparing it to the model that ignores cross terms. In practice, we observe that only very heavy cannibalization effects have a significant impact on the optimal solution, suggesting that retailers may overlook complementary effects that are common in retail.

• *Computational experiments and case study.* Our industry partners provided us with sales data from multiple stores, allowing us to test our model and algorithms on real-world instances. We first run computational experiments to test our algorithms in terms of running times and performance accuracy. We show that both the IP-based approach and the greedy algorithm perform well, as they compute near-optimal solutions within seconds (see Section 6). We then present a case study by applying our method on actual retail data. By comparing the predicted profit under our proposed algorithms to current practice, we quantify the added value of our model. Under our model assumptions and for a particular item considered in our case study, these tests suggest a profit increase of between 2% and 9% relative to current practice (see Section 7).

2. Literature Review

Our work is related to retail operations and promotion optimization. In particular, when a retailer needs to design and schedule promotions, it often consists of choosing the right price discounts as well as the appropriate promotion vehicles for each time period of the selling season. Typically, retailers make these two decisions independently. Namely, they decide on the promotion depth first, and only then choose which/when promotion vehicles to use. Price promotion decisions are discussed below, whereas a recent work on this topic can be found in Cohen et al. (2017). In this paper, we focus on tackling the latter task of scheduling promotion vehicles.

Sales promotions have been studied extensively in the literature, mostly in marketing and economics. We refer the reader to Blattberg and Neslin (1990) and the references therein for a comprehensive review. However, with regard to sales promotions, the marketing community is mainly focused on modeling and estimating dynamic sales models that can be used to derive managerial insights (see, e.g., Cooper et al. 1999 and Foekens et al. 1998), typically in the form of econometric or choice models. For example, Foekens et al. (1998) study econometric models based on scanner data to examine the dynamic effects of sales promotions. In this paper, however, we formulate the underlying problem using an optimization approach and compute near-optimal solutions for scheduling promotion vehicles. Note that the existing literature has considered additive and multiplicative demand in terms of the noise dependence (see, e.g., Chen and Simchi-Levi 2004) or the price dependence (see, e.g., Cohen et al. 2017). To our knowledge, the structural dependence of the demand on the promotion vehicles was not considered before in the operations management community (as mentioned, econometric models were proposed, such as Foekens et al. 1998). Inspired by existing demand models for price and noise dependence, we consider the additive and multiplicative forms of promotion vehicle dependence.

Optimizing of sales promotions is also closely related to the field of dynamic pricing. An extensive survey on this topic is provided by Talluri and van Ryzin (2004). Recent advances in scheduling price promotions can be found in Cohen et al. (2017), where the authors provide an optimization formulation with a demand model estimated from data as input. They propose an efficient algorithm based on discretely linearizing the objective, and show that their approximation yields near-optimal solutions (in the vast majority of practical instances), runs in milliseconds, and can easily be implemented by retailers. It is important to point out that Cohen et al. (2017) focus on the price promotion problem and do not consider the question of how to effectively schedule promotion vehicles. In this paper, our efforts are concentrated on questions surrounding the scheduling of promotion vehicles. To the best of our knowledge, we are the first to propose provably good promotion vehicle scheduling policies using an optimization approach.

As previously mentioned, our work is also related to retail operations, which has received a great deal of attention from both academics and practitioners. Nowadays, it has become very common for retailers (e.g., fashion, supermarkets, electronics, etc.) to hire business analysts or consultants to develop data-driven decision-making tools. Such retailers need to make a very large number of decisions at any point in time. These decisions typically include inventory, capacity, assortment, pricing, and promotions. Several works consider the problem of inventory management in a retail environment, and many tools were developed for demand forecasting and inventory planning (see, e.g., Cooper et al. 1999, Caro and Gallien 2010). The same statement can be made for both assortment planning (see the survey of Kök et al. 2008 and the references therein) and pricing decisions (see, e.g., Phillips 2005, Cohen et al. 2017). It is also worth noting that several prescriptive works in the marketing community study the impact of retail coupons (see, e.g., Heilman et al. 2002). However, to the best of our knowledge, the problem of optimally scheduling promotion vehicles in a retail environment has not been considered before. This paper is the first to address this retail operational problem by using a rigorous analytical model and developing efficient data-driven optimization approaches.

From a methodological perspective, the theoretical contributions of our work are obtained by synthesizing techniques related to computational complexity, approximation algorithms, and integer programming. Even though the technical part of this paper is self-contained, we assume that the reader is equipped with basic working knowledge in the above-mentioned topics. For this reason, to better understand some of our results, nonspecialists could still consult a number of excellent surveys and books related to the computation of independent sets in graphs (Pardalos and Xue 1994, Bomze et al. 1999, Gutin 2013), greedy methods in exact and approximation algorithms (see, e.g., Cormen et al. 2009, chap. 16; Kleinberg and Tardos 2005, chap. 4; Williamson and Shmoys 2010, chaps. 2 and 9), and integer programming (Schrijver 1998, Wolsey and Nemhauser 1999, Bertsimas and Weismantel 2005).

As previously mentioned, the concrete optimization problem considered in this paper (see Section 4.3) can be viewed as a bipartite matching-type problem in disguise, where promotion vehicles should be assigned to time periods. However, rather than maximizing a linear function, the concurrent utilization of several vehicles at any given time period has a multiplicative effect, leading to a nonlinear formulation. From this perspective, the problem of optimizing an arbitrary nonlinear function over the bipartite matching polytope is known to be NP-hard (see, e.g., Chandrasekaran et al. 1982, Berstein and Onn 2008). To our knowledge, Section 5.1 and Appendix B.2, where we connect between the promotion vehicle scheduling problem and detecting large independent sets in certain graph classes, provide new inapproximability bounds for nonlinear bipartite matching. From an algorithmic point of view, exact polynomial-time solution methods have been proposed over the years for computing bipartite matchings that optimize specific (nonlinear) objective functions subject to various structural assumptions. We refer the reader to selected papers in this context (Papadimitriou and Yannakakis 1982, Papadimitriou 1984, Mulmuley et al. 1987, Yi et al. 2002, Berstein and Onn 2008), and to the references therein, for a detailed literature review, as well as to additional related work on nonlinear integer programming and matroid optimization (Hassin and Tamir 1989, Onn 2003, Hochbaum 2007, Berstein et al. 2008, Lee et al. 2009, Hemmecke et al. 2010, Köppe 2012). We are not aware of straightforward ways to make use of these algorithms for the purpose of deriving our main results.

3. General Modeling Approach

In this paper, we consider the problem formulation (P), formally defined in Section 4.3, that was developed in collaboration with Oracle Retail and calibrated with actual retail data. This section is devoted to introducing the context and general formulation of the promotion vehicle scheduling problem.

We consider a single-item setting in which a retailer needs to decide how to schedule promotion vehicles for this particular item (see Section 8 for an extension to the multi-item setting). The retail manager's objective is to maximize the total profits during a finite time horizon, where the underlying decision is which promotion vehicles to use in each time period. Typically, a retailer chooses among five to 40 distinct promotion vehicles; examples include product placement at the end of an aisle (endcap display), dedication of an in-aisle display, flyer mailings, broadcast TV commercials, radio advertisements, tasting stands, and in-store flyers. In our model, the retailer does not decide on prices—the reason being that the promotion price optimization and promotion vehicle scheduling are generally solved by different departments of the retailer. Hence, the focus of this paper is on the promotion vehicle scheduling problem, which has not been studied rigorously before. To arrive at a concrete model formulation, we first introduce some useful notation:

- T—Number of time periods (e.g., weeks) in the planning horizon.
- V—Set of different vehicles available to the retailer.
- L_t—Limitation on the number of vehicles available at time t; i.e., an upper bound on the number of vehicles that can be assigned to time t.
- C_v—Upper bound on the number of times the retailer can use vehicle v throughout the planning season.
- x_{vt}—Binary decision variable that indicates whether vehicle v is assigned to time t.

Note that we have a total of $|V| \cdot T$ binary variables to be determined by the retailer. To maximize total profits, we need to understand how promotion vehicles affect demand. In practice, the demand of an item depends on various observable features such as current and past prices, prices of other products in the same category, shelf space, seasonality, trend effects, and promotion vehicles. In particular, using a promotion vehicle may enhance cumulative sales by generating additional traffic, increasing the visibility of the product, or making the customer aware of the product. One key challenge is to propose a demand model that captures this effect. For example, one can consider a general time-dependent demand function $d_t(\cdot,\cdot)$ that explicitly depends on a vector of prices denoted by p_t (that could include current and past prices as well as prices of other products), and also on the promotion vehicles being used—i.e., d_t is a function of p_t and $\{x_{vt}\}_{v\in V}$. In Section 4, motivated by real data, we propose a class of models that aim to capture the effects of promotion vehicles on demand.

We assume that this deterministic demand function provides an accurate estimate of expected demand. The good out-of-sample forecasting metrics obtained in Section 4 serve as justification for this assumption. In addition, we assume that the retailer has sufficient inventory to meet demand, so that predictions of sales and demand are equivalent. This assumption is reasonable for grocery retailers, and in particular for nonperishable packaged goods such as coffee and cereals. Literature has shown that grocery retailers recognize the negative effects of stocking out of promoted products (see e.g., Corsten and Gruen 2004, Campo et al. 2000) and use accurate demand forecasting models (e.g., Cooper et al. 1999, Van Donselaar et al. 2006).

In practice, there are several business rules that constrain the promotion vehicle schedule. These rules are usually dictated by the brand's manufacturer or related to certain financial/spatial constraints of the retailer. Below, we discuss the different business rules that our formulation incorporates.

1. *Limited number of times a particular vehicle can be used.* For example, during the next quarter, a total of four in-store flyers and two TV advertisements are available. This rule may come from a contract between the manufacturer and the retailer, where the manufacturer covers the cost of using a particular promotion vehicle in exchange for an increased visibility. One can encode this requirement as

$$\sum_{t=1}^{T} x_{vt} \le C_v \quad \forall\, v \in V. \tag{1}$$

2. *Limited number of vehicles per time period.* For instance, during a given week, the retailer can use at most four vehicles. During another week, in which a holiday event occurs, at most seven vehicles can be used. These rules are usually known up front for the entire selling season. One can impose the following constraint in the formulation:

$$\sum_{v\in V} x_{vt} \le L_t \quad \forall\, t \in [T], \tag{2}$$

where $[T]$ denotes $1,\ldots,T$.

3. *A particular promotion vehicle has to be used* (*or cannot be used*) *at a specific time period.* In many cases, the retailer anticipates promotional events and knows in advance that a particular vehicle has to be used during a specific week (e.g., a tasting stand for a particular brand is scheduled during a given week). This translates into $x_{vt} = 1$. Alternatively, the retailer may not be allowed to use a particular vehicle at a certain time period—i.e., $x_{vt} = 0$.

In this context, some retailers might want to impose global constraints. For example, certain coupons might need be mailed out for several stores simultaneously. These global requirements can be incorporated into our computational framework. In the case that several stores are required to implement the same promotion schedule, we can pool the separate store demand functions into a single aggregate demand function, virtually treating the different stores as a single aggregate store. With this new aggregate demand, we can solve the promotion vehicle scheduling problem and implement the resulting promotion schedule in all stores. That being said, the retailer considered in our case study implements a decentralized promotion schedule, where each store is responsible for running its own promotion campaigns. This policy is frequently used when the stores are located in different states, with different management teams. As a result, we do not incur such global constraints in this paper.

In the problem formulation below, we capture the above business rules as linear constraints. The retailer maximizes the total profit during the planning horizon, while satisfying the business rules:

$$\begin{aligned}
\max \quad & \sum_{t=1}^{T} (p_t - c_t)\, d_t(p_t, \{x_{vt}\}_{v\in V}) \\
\text{s.t.} \quad & \sum_{t=1}^{T} x_{vt} \le C_v \quad \forall\, v \in V, \\
& \sum_{v\in V} x_{vt} \le L_t \quad \forall\, t \in [T], \\
& x_{vt} \in \{0, 1\} \quad \forall\, v \in V,\ t \in [T].
\end{aligned}$$

Here, p_t and c_t are the price and cost of the item at time t, respectively. Based on the assumption that the prices have been determined in advance, when solving the above optimization problem, the vector of prices p_t (that can include past and cross-item prices) is assumed to be known. In Section 4.3, we explain how to easily incorporate constraints of the form $x_{vt} = 1$ or $x_{vt} = 0$ using basic preprocessing steps.

4. Empirical Motivation and the Multiplicative Model

In what follows, our goal is to use real data to motivate the promotion vehicle scheduling problem from a business perspective. Concretely, we present an empirical motivation for considering different demand models $d_t(p_t, \{x_{vt}\}_{v\in V})$. In particular, we discuss how demand depends on the use of promotion vehicles $\{x_{vt}\}_{v\in V}$.

4.1. Data Description

Our data set contains data collected from 18 stores of a large supermarket client of the Oracle Retail Global Business Unit. This data set spans a period of roughly two years, from the beginning of 2009 to mid 2011, which we split into a training set consisting of the first 80 weeks and a testing set consisting of the last 33 weeks. The product category we focus on is the coffee category, in which 21 different promotion vehicles were used. This category contains products differing by brand, size, and coffee roast, just to name a few examples. As we consider the single-item problem, we select one particular item, representative of the entire category. For confidentiality reasons, the precise name of this item cannot be specified.

To estimate the boosts in demand due to promotion vehicles, the dependent variable in each observation is the weekly sales of our item in one of the supermarkets. It is worth noting that we assume forecasting sales and demand to be equivalent; this assumption was justified in Section 3, especially for a nonperishable item such as coffee. The independent variables used to describe weekly coffee sales are the store at which sales are made, the trend (increasing or decreasing sales over time), the seasonality (time of the year), the normalized price, the normalized prices of the past four weeks, and the 21 different promotion vehicles. Some of the most efficient promotion vehicles found in our data set include the following: sending a coupon to customers (*Mailing Coupon*); promoting the product in a flyer, particularly in the first few pages (*Flyer Front*); the middle pages (*Flyer Mid*), and the final pages (*Flyer End*); promoting the product by placing an in-store display (*Display*); offering a bonus snack with soft drinks (*Bonus Snack*); and featuring the product in a TV commercial (*TV Commercial*).

4.2. Demand Model Estimation and Selection

As previously discussed, we assume that the demand $d_t(p_t, \{x_{vt}\}_{v\in V})$ depends on the prices but also explicitly on the promotion vehicles. In what follows, we consider two potential models for the function $d_t(p_t, \{x_{vt}\}_{v\in V})$ and analyze their performance on real data. In the first model, demand has an additive linear dependence on prices and promotion vehicles:

$$d_t^{s,i} = \beta_0^{A,s,i} + \beta_1^A t + \beta_2^{A,i} p_t^{s,i} + \sum_{j=1}^{4} \beta_{2+j}^{A,i} p_{t-j}^{s,i} + \sum_{v=1}^{21} \gamma_{vt}^A x_{vt}^{s,i} + \epsilon_t^{s,i}.$$

The second model assumes a multiplicative log-linear dependence:

$$\log(d_t^{s,i}) = \beta_0^{M,s,i} + \beta_1^M t + \beta_2^{M,i} \log(p_t^{s,i}) + \sum_{j=1}^{4} \beta_{2+j}^{M,i} \log(p_{t-j}^{s,i}) + \sum_{v=1}^{21} \gamma_{vt}^M x_{vt}^{s,i} + \epsilon_t^{s,i},$$

where $d_t^{s,i}$ and $p_t^{s,i}$ represent the demand and price for item i in store s at time t. The variable $p_{t-j}^{s,i}$ is the price of item i in store s at time $t-j$, $x_{vt}^{s,i}$ indicates whether vehicle v is used for item i in store s at time t, and $\epsilon_t^{s,i}$ is an i.i.d. normally distributed noise for all observations.

The parameters capture the following effects. First, $\beta_0^{s,i}$ captures the baseline sales of item i in store s at any time, while β_1 incorporates the trend in demand. Second, β_2^i captures the price elasticity of item i, whereas $\beta_3^i, \ldots, \beta_6^i$ incorporate item i's effect of past prices on current demand. Including these variables in the demand model allows us to capture the well-known stockpiling effect in groceries (see chap. 12 of Blattberg and Neslin 1990). This effect occurs often when consumers purchase larger quantities during price promotions. As a result, consumers stockpile the item for future consumption, especially for nonperishable items. From a modeling perspective, past promotions decrease current demand, which can be captured by using past prices as independent variables in our demand model. Finally, γ_{1t} through γ_{21t} are the parameters of interest, as they capture the boost in demand generated by each of the 21 vehicles at each time period.

The models described above are both commonly used in practice (for example, by Oracle Retail) and in the academic literature (see, e.g., Van Heerde et al. 2000, Macé and Neslin 2004). Nevertheless, most of the models studied so far do not explicitly include promotion vehicle effects. Few works, such as that of Wittink et al. (1988), propose demand models that incorporate a limited number of promotion vehicles. In this paper, we generalize existing demand models to explicitly incorporate the effects of promotion vehicles on demand.

To select the demand model that provides the best description of how demand depends on promotion vehicles, we use ordinary least squares regression and apply stepwise selection based on the Akaike information criterion (AIC) and the Bayesian information criterion (BIC)[1] to obtain the additive demand model:

$$d_t^{s,i} = \beta_0^{A,s,i} + \beta_1^A t + \beta_2^{A,i} p_t^{s,i} + \beta_3^{A,i} p_{t-1}^{s,i} + \sum_{v=1}^{19} \gamma_v^A x_{vt}^{s,i} + \epsilon_t^{s,i}, \tag{3}$$

and the multiplicative demand model:

$$\log(d_t^{s,i}) = \beta_0^{M,s,i} + \beta_1^M t + \beta_2^{M,i} \log(p_t^{s,i}) + \beta_3^{M,i} \log(p_{t-1}^{s,i}) + \sum_{v=1}^{21} \gamma_v^M x_{vt}^{s,i} + \epsilon_t^{s,i}. \tag{4}$$

Because of the sparsity of our data set, we are unable to estimate cross-term effects, and require the boost of each vehicle v to be time independent—i.e., $\gamma_{vt} = \gamma_v$ for all t. According to Oracle researchers and the client's managers, both assumptions are justified for the coffee category. Nonetheless, the analytical results derived in this paper hold for the more general setting, where γ_{vt} can be time dependent.

After estimating these two regression models, we are interested in deciding which of the two provides a

Table 1. Out-of-Sample Forecasting Metrics of the Additive and Multiplicative Models for Different Items

Models	Item	R^2	MAPE	MAE
Additive regression	1	−0.0815	1.3187	29.4980
Multiplicative regression	1	0.7798	0.6725	14.5809
Additive regression	2	0.6431	1.9788	8.6900
Multiplicative regression	2	0.9453	0.5164	2.8520
Additive regression	3	0.6260	0.8703	19.0530
Multiplicative regression	3	0.6823	0.4327	17.0312

better fit to the data. To compare the additive and multiplicative demand models, we compute out-of-sample forecasting metrics for both models. Table 1 presents the out-of-sample R^2, mean absolute percentage error (MAPE), and mean absolute error (MAE) of the two stepwise selected demand models when applied to three representative items in the coffee category. These three items were selected such that a significant use of promotion vehicles was observed in the data. To ensure some diversity, we decided to select one item from the major house brand (private label) and two items from major national brands (one low-selling and one high-selling).

Table 1 demonstrates that the multiplicative model (4) outperforms the additive model (3) in all forecasting metrics. In fact, the differences in these metrics are substantial, which leads us to conclude that the multiplicative model has a significantly better predictive power. One possible reason could be that the additive model suffers from a scale independence, as it assumes an absolute boost independent of the number of sales. For example, sales can differ considerably between stores, making it difficult to estimate a uniform additive parameter for each promotion vehicle. Thus, a relative term, such as in the multiplicative model, seems to be more suitable. Overall, the forecasting accuracy of the multiplicative model (4) is good when compared to standard forecasting models in retail operations, especially when predicting sales for individual items in specific stores.

Based on the preceding discussion, in the remainder of this paper, we consider a broader class of models that subsume the multiplicative log-linear model (4) as a special case. The specifics of this class, which is referred to as the *multiplicative model*, are presented in Section 4.3. The alternative way of modeling the promotion vehicle scheduling problem, by considering an additive model, is reported in Online Appendix C. Even though the computational problem resulting from the latter class can be optimized efficiently, our analysis shows a worse fit to actual retail data.

4.3. The Multiplicative Model

We consider a general class of demand models where the effect of promotion vehicles is multiplicative. From a practical perspective, these models are easy to estimate from data, and provide a meaningful interpretation to each estimated parameter. The multiplicative demand model assumes that the price and vehicle effects are multiplicative:

$$d_t(p_t, \{x_{vt}\}_{v\in V}) = h_t^M(p_t) \cdot \prod_{v\in V} B_{vt}^{x_{vt}}. \tag{5}$$

The function $h_t^M(p_t)$ represents the effect of the price vector p_t on demand, which can include current and past prices as well as cross prices from other products. Each boost parameter $B_{vt} \geq 1$ corresponds to the relative increase in demand when vehicle v is used at time t. For example, if $B_{vt} = 1.03$, then assigning vehicle v at time t yields a 3% increase in demand, relative to the case where this vehicle is not used. Note that, when vehicle v is not used at time t, we have $x_{vt} = 0$, meaning that the nominal demand is unaffected. We consider an extension of demand model (5) that includes cross-term effects (i.e., interactions between pairs of vehicles) in Appendix B.

Altogether, the promotion vehicle scheduling problem can be stated as follows:

$$\begin{aligned}
&(\mathrm{P}) \quad \max \sum_{t=1}^{T} \alpha_t \prod_{v\in V} B_{vt}^{x_{vt}}\\
&(\mathrm{C}_1) \quad \sum_{t=1}^{T} x_{vt} \leq C_v \qquad \forall\, v \in V,\\
&(\mathrm{C}_2) \quad \sum_{v\in V} x_{vt} \leq L_t \qquad \forall\, t \in [T],\\
&(\mathrm{C}_3) \quad x_{vt} \in \{0,1\} \qquad \forall\, v \in V,\ t \in [T].
\end{aligned}$$

Here, the decision variables are x_{vt}, indicating whether vehicle v is scheduled at time t. Typically, in retail applications, the number of vehicles $|V|$ ranges between five and 40. Without loss of generality, we assume that $B_{vt} \geq 1$ for every $v \in V$ and $t \in [T]$. Furthermore, we assume that $\max_v B_{vt} > 1$ for every $t \in [T]$, as otherwise, there is no reason to assign any vehicle to this time period. The latter assumption will be particularly useful for simplifying the analysis of our integer programming formulation in Section 5.3. For convenience, we use α_t to represent the effect of price on profits at time t. More precisely, α_t is equal to the profit margin multiplied by the part of the demand affected by prices—i.e., $\alpha_t = (p_t - c_t) \cdot h_t^M(p_t)$. Since all prices are assumed to be given a priori, α_t is a given quantity as well.

Without loss of generality, we consider only business rules (1) and (2). It is not difficult to verify that one can perform simple preprocessing modifications to capture constraints of the form $x_{vt} = 1$ or $x_{vt} = 0$, mentioned in Section 3. Indeed, $x_{vt} = 0$ can be taken care of by setting $B_{vt} = 1$. In addition, $x_{vt} = 1$ can be handled by modifying α_t to $\alpha_t B_{vt}$, setting $B_{vt} = 1$, and decreasing the values of C_v and L_t by one unit.

It is worth mentioning that a straightforward approach to obtaining a linear formulation is to make use of subset-type variables, $y_{U,t}$, indicating whether the set of vehicles $U \subseteq V$ is assigned to time period t. However, even though one can easily express the objective function and constraints in terms of the $y_{U,t}$ variables, unfortunately, there are $O(2^{|V|} \cdot T)$ such variables—i.e., exponentially many in the number of vehicles. From a practical perspective, utilizing this formulation becomes impractical as the number of promotion vehicles increases beyond 18–20. In such cases, commercial solvers take several hours or even days to compute an optimal solution for a single instance of the problem. In practical retail settings, this scenario is encountered frequently as shown by the 21 promotion vehicles from the real-world data described in this paper.

5. Hardness and Approximability

5.1. Hardness of Approximation

In what follows, we prove that it is NP-hard to approximate the promotion vehicle scheduling problem (P) in polynomial time within some constant factor. To this end, we relate the approximability of this model to that of computing maximum independent sets in Δ-regular graphs (henceforth, Max-IS$_\Delta$). We begin by recalling how the latter problem is defined, and state some known hardness results in this context.

An instance of Max-IS$_\Delta$ is specified by an undirected Δ-regular graph $G = (N, E)$, meaning that the degree of each vertex is precisely Δ. A subset of vertices $U \subseteq N$ is said to be independent if for every pair of vertices in U, there is no edge connecting the two. The objective is to compute an independent set of maximal cardinality. The most useful hardness result for our purposes states that even Max-IS$_3$ is APX-hard (Halldórsson and Yoshihara 1995, Berman and Fujito 1999, Alimonti and Kann 2000), meaning that it cannot be approximated better than some given constant, unless P = NP. In fact, the problem of computing maximum independent sets in Δ-regular graphs has not been shown at present time to admit a better approximation than in Δ-bounded-degree graphs (where the degree of each vertex is at most Δ), a more general case known to be inapproximable within factor $O(\Delta^{1-\epsilon})$, for any fixed $\epsilon > 0$ (Håstad 1996).

Theorem 1. *There exists some constant $\beta < 1$, such that the promotion vehicle scheduling problem cannot be approximated within factor greater than β, unless* P = NP.

Proof. To establish the claim, we describe an approximation-preserving reduction from Max-IS$_\Delta$ to the promotion vehicle scheduling problem. For this purpose, given an instance $G = (N, E)$ of Max-IS$_\Delta$, we create a corresponding instance of the promotion vehicle scheduling problem as follows:

- The set of vehicles is E, while the set of time periods is N. That is, the edges and vertices of G serve as vehicles and time periods, respectively.
- Each vehicle $e \in E$ has a unit capacity—i.e., $C_e = 1$. Each time period $v \in N$ has a capacity of $L_v = \Delta$.
- Now, let $S_v \subseteq E$ be the star centered at the vertex v, that is, the collection of edges adjacent to v, noting that $|S_v| = \Delta$, as G is a Δ-regular graph. Then, for each time period v, we set $\alpha_v = 1$, while its related boosts are given by

$$B_{ev} = \begin{cases} |N|^2 & \text{if } e \in S_v, \\ 1 & \text{if } e \notin S_v. \end{cases}$$

In other words, we have just created the following instance:

$$\begin{aligned} &(\mathrm{P}) && \max \sum_{v\in N} |N|^{2\sum_{e\in S_v} x_{ev}} && \\ &(\mathrm{C}_1) && \sum_{v\in N} x_{ev} \le 1 && \forall\, e \in E, \\ &(\mathrm{C}_2) && \sum_{e\in E} x_{ev} \le \Delta && \forall\, v \in N, \\ &(\mathrm{C}_3) && x_{ev} \in \{0,1\} && \forall\, v \in N,\, e \in E. \end{aligned}$$

Claim 1. *Let U^* be a maximum-cardinality independent set in G. Then,* $\mathrm{OPT}(\mathrm{P}) \ge |N|^{2\Delta} \cdot |U^*|$.

Proof. We construct a feasible solution x, by setting $x_{ev} = 1$ whenever e appears in the star S_v and, at the same time, v is picked by the independent set U^* (namely, $e \in S_v$ and $v \in U^*$); otherwise, $x_{ev} = 0$. To see why x is indeed feasible for (P), note that since U^* is an independent set, any edge e has at most one endpoint in U^*, meaning that this edge is assigned at most once. Moreover, since the set of edges assigned to each vertex $v \in U^*$ is exactly S_v, and recalling that $|S_v| = \Delta$, the objective value of x is

$$\begin{aligned} \sum_{v\in N} |N|^{2\sum_{e\in S_v} x_{ev}} &\ge \sum_{v\in U^*} |N|^{2\sum_{e\in S_v} x_{ev}} \\ &= \sum_{v\in U^*} |N|^{2|S_v|} = |N|^{2\Delta} \cdot |U^*|. \quad \square \end{aligned}$$

Claim 2. *Let x be a feasible solution to* (P)*, with objective value $\mathcal{V}(x)$. Given this solution, we can efficiently compute an independent set in G of size at least $\mathcal{V}(x)/|N|^{2\Delta} - 1/|N|$.*

Proof. The important observation to make notice of is that, for every vertex $v \in N$, since $|S_v| = \Delta$, we have $|N|^{2\sum_{e\in S_v} x_{ev}} \le |N|^{2\Delta}$. Moreover, this holds as an equality if and only if $x_{ev} = 1$ for every $e \in S_v$. Therefore, when the latter condition is not satisfied, we actually have $|N|^{2\sum_{e\in S_v} x_{ev}} \le |N|^{2(\Delta-1)}$. Consequently, letting U_x denote the set of vertices for which equality holds,

$$\begin{aligned} |U_x| \cdot |N|^{2\Delta} &\ge \sum_{v\in N} |N|^{2\sum_{e\in S_v} x_{ev}} - |N \backslash U_x| \cdot |N|^{2(\Delta-1)} \\ &\ge \mathcal{V}(x) - |N|^{2\Delta-1}, \end{aligned}$$

meaning that $|U_x| \ge \mathcal{V}(x)/|N|^{2\Delta} - 1/|N|$. Also, U_x is necessarily an independent set in G. Otherwise, there is an edge $e = (u, v)$ between two vertices in U_x, implying that the solution x cannot assign $\Delta = |S_v| = |S_u|$ edges to both, since $e \in S_v \cap S_u$ can be assigned to at most one vertex. This contradicts the definition of U_x. $\square$

With the above claims in place, the existence of a polynomial-time β-approximation for the vehicle scheduling problem implies that Max-IS$_\Delta$ can be efficiently approximated within factor $\beta - 1/|N|$, as one is able to compute an independent set of cardinality at least $(\beta \cdot |N|^{2\Delta} \cdot |U^*|)/|N|^{2\Delta} - 1/|N| \geq (\beta - 1/|N|) \cdot |U^*|$. Since Max-IS$_\Delta$ is known to be APX-hard (Halldórsson and Yoshihara 1995, Berman and Fujito 1999, Alimonti and Kann 2000), this concludes our proof. □

5.2. The Greedy Algorithm

In this section, we propose an efficient greedy method for approximating the promotion vehicle scheduling problem. Our algorithm is guaranteed to compute an assignment whose objective value is within factor $\Delta + 1$ of optimal. Here, Δ stands for the maximal number of vehicles that can be assigned to any time period—i.e., $\Delta = \max_t L_t$. It is worth pointing out that obtaining a performance guarantee that is sublinear in Δ (e.g., $\log \Delta$ or even $\sqrt{\Delta}$) would immediately translate into a sublinear approximation for computing maximum independent sets in Δ-regular graphs, via the reduction given in Section 5.1. A result of this nature is not known at the present time.

The algorithm. To simplify the presentation, we begin by introducing some helpful notation. Let C_s be a $|V|$-dimensional vector, indicating the remaining capacity of each vehicle when step s of the algorithm begins. That is, for any vehicle v, the value of $C_{s,v}$ is equal to the initial capacity C_v of this vehicle minus the number of times it has already been assigned in steps $1, \ldots, s-1$. Also, let A_s stand for the set of active time periods at the beginning of step s, which are time periods that have not been assigned any vehicle in steps $1, \ldots, s-1$. Our algorithm proceeds as follows:

1. We initialize $C_{1,v} = C_v$ for any vehicle v, with all time periods active—i.e., $A_1 = \{1, \ldots, T\}$.
2. For $s = 1, \ldots, T$:

(a) For every currently active time period $t \in A_s$, we compute the subset of vehicles U_s^t that maximizes the boost $\prod_{v \in U_s^t} B_{vt}$, allowing to pick up to L_t vehicles v out of those with positive remaining capacity $C_{s,v}$. This subset is obtained by picking vehicles in nonincreasing order of B_{vt}-value (breaking ties arbitrarily), until either L_t vehicles are picked or all vehicles with positive remaining capacities have been picked.

(b) Let t^* be the time period $t \in A_s$ for which $\alpha_t \cdot \prod_{v \in U_s^t} B_{vt}$ is maximized.

(c) We assign the vehicles in $U_s^{t^*}$ to time period t^*, make this period inactive (i.e., update $A_{s+1} \leftarrow A_s \backslash \{t^*\}$), and decrement the remaining capacity of each vehicle $v \in U_s^{t^*}$ (i.e., set $C_{s+1,v} \leftarrow C_{s,v} - 1$).

Example. Prior to analyzing the algorithm, we present a small illustrative example in Figure 1. This setting consists of three vehicles and four time periods with boost parameters B_{vt}, limitations C_v and L_t, and underlying profits α_t, as given in each table. Figure 1(a) presents the first step of the greedy algorithm. In particular, for each time period, we compute a feasible set of vehicles that yields the largest boost. In this example, we obtain $\{v_1, v_2\}$ for t_1, $\{v_1, v_3\}$ for t_2, $\{v_3\}$ for t_3, and $\{v_3\}$ for t_4. Next, the profit gains are calculated for each time period and shown in the bottom of Figure 1(a) (e.g., for time t_1, the gain is $\alpha_{t_1} B_{v_1 t_1} B_{v_2 t_1} = 1.872$). The profit gain of time period t_4 happens to be the largest and, therefore, vehicle v_3 is assigned to t_4. In Figure 1(b), the limitation parameters C_v and L_t are updated to account for the assignment of v_3 to t_4 (i.e., C_{v_3} is decreased by one unit and period t_4 is made inactive, with $L_{t_4} = 0$). The next step of the greedy algorithm repeats the same procedure by computing the sets yielding the largest boosts, to obtain $\{v_1, v_2\}$ for t_1, $\{v_1, v_2\}$ for t_2, and $\{v_1\}$ for t_3 (note that vehicle v_3 is depleted). The profit gains are recalculated, and vehicles v_1 and v_2 are assigned to t_2. The algorithm

Figure 1. (Color online) Example Steps of the Greedy Algorithm

(a) Step 1

B_{vt}	t_1	t_2	t_3	t_4	C_V
v_1	1.3	1.4	1.6	1.7	3
v_2	1.2	1.3	1.4	1.5	2
v_3	1.1	1.4	1.7	2.0	1
L_t	2	2	1	1	
α_t	1.2	1.6	1.2	1.6	
Gain	1.872	3.136	2.04	3.2	

(b) Step 2

B_{vt}	t_1	t_2	t_3	t_4	C_V
v_1	1.3	1.4	1.6	1.7	3
v_2	1.2	1.3	1.4	1.5	2
v_3	1.1	1.4	1.7	2.0	0
L_t	2	2	1	0	
α_t	1.2	1.6	1.2	1.6	
Gain	1.872	2.912	1.92	3.2	

continues this procedure until either all time periods become inactive or all vehicles are depleted.

Analysis. The next theorem establishes the performance guarantee of the greedy algorithm.

Theorem 2. *The greedy algorithm approximates the promotions vehicle scheduling problem within factor* $\Delta+1$.

Proof. Consider some fixed optimal solution, and let $R_1^*, \ldots, R_T^*$ be the subsets of vehicles assigned to periods $1, \ldots, T$, respectively. We will establish the performance guarantee stated in Theorem 2 by considering an auxiliary procedure that runs in parallel to the greedy algorithm, only for purposes of analysis. Here, we start with $A_1^* = \{1, \ldots, T\}$. In each step s, whenever the greedy algorithm assigns $U_s^{t^*}$ to time period t^*, we define A_{s+1}^* by deleting several time periods from A_s^*:

1. If $t^* \in A_s^*$, then the time period t^* is removed.
2. For each vehicle $v \in U_s^{t^*} \backslash R_{t^*}^*$, if one or more of the sets in $\{R_t^*: t \in A_s^*\}$ contains v, we pick one of these sets arbitrarily and remove it.

Based on this procedure, we have the following properties at any step s:

- $A_s^* \subseteq A_s$. In other words, the set of active time periods in the greedy algorithm contains all time periods that have not been deleted yet from the optimal solution (by the auxiliary procedure). This property is an immediate consequence of the auxiliary deletion in item 1 above.
- $C_{s,v} \geq C_{s,v}^*$ for any vehicle $v \in V$, where $C_{s,v}^* = |\{t \in A_s^*: v \in R_t^*\}|$. That is, for any vehicle, its remaining capacity at any step of the greedy algorithm is at least as large as the number of times it appears in optimal subsets corresponding to time periods in A_s^*. This property is shown in Lemma 1.

Lemma 1. *At any step* s, *we have* $C_{s,v} \geq C_{s,v}^*$ *for any vehicle* $v \in V$.

Proof. We prove this claim by induction on the step number s. For $s=1$, the claim obviously holds since $C_{1,v} = C_v$, by definition, whereas $C_{1,v}^* \leq C_v$. For $s \geq 2$, we begin by observing that both $C_{s,v}$ and $C_{s,v}^*$ are nonincreasing in s for any vehicle $v \in V$, by definition of the greedy algorithm and our auxiliary procedure. Then, during step s, exactly one of the following cases occurs for any $v \in V$.

Case 1: $v \notin U_s^{t^*}$. In this case, $C_{s+1,v} = C_{s,v}$, and by the induction hypothesis,

$$C_{s+1,v} = C_{s,v} \geq C_{s,v}^* \geq C_{s+1,v}^*.$$

Case 2: $v \in U_s^{t^*}$ and $v \in R_t^*$ for some $t \in A_s^*$. Here, $C_{s+1,v} = C_{s,v} - 1$, while the auxiliary procedure guarantees that $C_{s+1,v}^* \leq C_{s,v}^* - 1$. Therefore, by the induction hypothesis,

$$C_{s+1,v} = C_{s,v} - 1 \geq C_{s,v}^* - 1 \geq C_{s+1,v}^*.$$

Case 3: $v \in U_s^{t^*}$ and $v \notin R_t^*$ for all $t \in A_s^*$. In this case, $C_{s,v}^* = 0$ by definition, and

$$C_{s+1,v} \geq 0 = C_{s,v}^* \geq C_{s+1,v}^*. \quad \square$$

Based on these properties, we know that at any step s, the profit obtained by the greedy algorithm, $\alpha_{t^*} \cdot \prod_{v \in U_s^{t^*}} B_{vt^*}$, is at least as large as the profit $\alpha_t \cdot \prod_{v \in R_t^*} B_{vt}$, for any time period $t \in A_s^*$. This follows by observing that every time period in A_s^* is still active in the greedy algorithm (as $A_s^* \subseteq A_s$), which also has the remaining vehicles to consider the subset R_t^* as the one to pick in the current step (since $C_{s,v} \geq C_{s,v}^*$). In particular, $\alpha_{t^*} \cdot \prod_{v \in U_s^{t^*}} B_{vt^*}$ is at least as large as the profit for each time period deleted in step s from A_s^*, noting that since $|U_s^{t^*}| \leq L_{t^*} \leq \Delta$, at most $\Delta+1$ such periods were deleted (more specifically, at most one period in item 1 of the auxiliary procedure, and at most $|U_s^{t^*} \backslash R_{t^*}^*| \leq |U_s^{t^*}| \leq \Delta$ in item 2). Therefore, summing the profits obtained by the greedy algorithm over all steps, we obtain a combined profit of at least $1/(\Delta+1)$ times the total profit of the optimal solution. $\square$

Tight example. Even though the greedy algorithm approximates the promotion vehicle scheduling problem within factor $\Delta+1$, it is still unclear whether our analysis is tight with respect to the parameter Δ. In Appendix A.1, we show that this is indeed the case (up to an additive factor of 1) by presenting a carefully constructed example, proving the next claim.

Lemma 2. *There is a sequence of instances for which the ratio between the profit of an optimal solution and that of the greedy algorithm approaches* Δ.

Remark. Our algorithm can be viewed as a variant of the greedy approach for approximating an extension of the k-set packing problem (Arkin and Hassin 1998, Chandra and Halldorsson 2001, Berman 2000, Hazan et al. 2006) with exponentially many subsets. In this context, each possible combination of a time period t and at most L_t vehicles defines a subset, whose weight is given by the profit obtained using the corresponding assignment. Under capacity constraints for each vehicle, the objective can alternatively be thought of as picking a maximum weight collection of subsets that satisfies these capacities. Due to having $O(|V|^{O(\Delta)})$ subsets, this collection needs to be handled in an implicit way, along the same lines as how our algorithm operates.

Remark. It is worth mentioning that, in Online Appendix D, we develop a PTAS for the special case of uniform vehicle boosts (i.e., $B_{vt} = B$ for any vehicle v and time period t), and uniform base profits of time periods (i.e., $\alpha_1 = \cdots = \alpha_T$).

5.3. Approximate Integer Program

As the tight example in Lemma 2 demonstrates, one can carefully design problem instances for which the greedy algorithm constructs suboptimal vehicle assignments. While these instances are very different from those considered in practice, we are still motivated to devise a provably good method that computes near-optimal solutions, possibly at the expense of being less efficient in terms of running time. For this purpose, we prove that, in spite of having a nonlinear objective function, the promotion vehicle scheduling problem can be approximated within any degree of accuracy as an integer (linear) program with polynomially many variables and constraints. Specifically, the remainder of this section is devoted to establishing the next theorem.

Theorem 3. *Given an accuracy parameter $\epsilon > 0$, we can efficiently construct an integer program* (IP_ϵ) *that satisfies the following properties*:

1. *The combined number of variables and constraints in* (IP_ϵ) *is polynomial in the input size of* (P) *and in* $1/\epsilon$.
2. (IP_ϵ) *provides a* $(1-\epsilon)$*-approximation to* (P). *That is,* $\mathrm{OPT}(\mathrm{IP}_\epsilon) \geq (1-\epsilon)\cdot \mathrm{OPT}(\mathrm{P})$, *and moreover, any solution to* (IP_ϵ) *can be efficiently translated to* (P) *without any loss in optimality.*

Ingredient 1: The integer program. Let $\mathcal{D} \subseteq \mathbb{R}_+$ be a finite discretization set, as defined in ingredient 2 below, consisting of nonnegative real numbers, with $0 \in \mathcal{D}$. With respect to this set, our integer program (IP_ϵ) is defined as follows:

$$(\mathrm{IP}_\epsilon) \quad \max \sum_{t=1}^{T} \alpha_t \sum_{r\in\mathcal{D}} (e^r \cdot y_{tr})$$

$$(\mathrm{C}_1) \quad \sum_{t=1}^{T} x_{vt} \leq C_v \qquad \forall\, v \in V,$$

$$(\mathrm{C}_2) \quad \sum_{v\in V} x_{vt} \leq L_t \qquad \forall\, t \in [T],$$

$$(\mathrm{C}_3) \quad \sum_{r\in\mathcal{D}} y_{tr} = 1 \qquad \forall\, t \in [T],$$

$$(\mathrm{C}_4) \quad y_{tr} \leq \frac{1}{r} \sum_{v\in V} \ln(B_{vt}) \cdot x_{vt} \qquad \forall\, t \in [T],\ r \in \mathcal{D}\backslash\{0\},$$

$$(\mathrm{C}_5) \quad x_{vt},\, y_{tr} \in \{0,1\} \qquad \forall\, v \in V,\ t \in [T],\ r \in \mathcal{D}.$$

Here, the assignment variables x_{vt} play precisely the same role as they did in the original formulation (P), meaning that x_{vt} indicates whether vehicle v is scheduled at time t. We also make use of additional indicator variables y_{tr}, defined for each time period t and value $r \in \mathcal{D}$. Intuitively, y_{tr} indicates whether we are using e^r to slightly underestimate the boost $\prod_{v\in V} B_{vt}^{x_{vt}}$ at time t, leading to the linear term $e^r \cdot y_{tr}$ in the objective function. Constraints (C_1) and (C_2) are the original upper bounds on the number of times each vehicle is assigned throughout the planning horizon, and on the number of different vehicles assigned to each time period. Constraint (C_3) states that only one estimate is picked for each time period. Finally, constraint (C_4) ensures that, when we pick an underestimate of e^r for the boost $\prod_{v\in V} B_{vt}^{x_{vt}}$ in time period t (by setting $y_{tr}=1$), the assignment variables x_{vt} indeed generate a sufficiently large boost; it is easy to verify that the linear inequality $y_{tr} \leq (1/r)\sum_{v\in V} \ln(B_{vt}) \cdot x_{vt}$ guarantees this condition.

Ingredient 2: Defining the set $\mathcal{D}$. For the construction to follow, it is convenient to make use of two input parameters. First, $B^+_{\min}$ stands for the minimum value of any B_{vt}, taking into account only vehicle–period pairs with $B_{vt} > 1$—namely, $B^+_{\min} = \min[\{B_{vt}\colon v \in V, t \in [T]\} \cap (1,\infty)]$. By our initial assumption (see Section 4.3), the latter set is indeed nonempty. Second, $B_{\max} > 1$ is the maximum value of any B_{vt}. With these parameters, we begin by initializing $\mathcal{D} = \{0\}$. This set is then augmented by all breakpoints that are created when the interval $[\ln(B^+_{\min}), \Delta \cdot \ln(B_{\max})] \subseteq (0,\infty)$ is geometrically partitioned by powers of $1 + 1/M$, where $M = (\Delta/\epsilon) \cdot \ln(B_{\max})$. In other words,

$$\mathcal{D} = \left\{0, \ln(B^+_{\min}), \left(1+\frac{1}{M}\right)\ln(B^+_{\min}), \left(1+\frac{1}{M}\right)^2 \ln(B^+_{\min}), \ldots\right\}.$$

Proof of Theorem 3, item 1. This part of the theorem is rather straightforward. To show that the size of (IP_ϵ) is polynomial in the input size of (P) and in $1/\epsilon$, it suffices to show that the discretization set $\mathcal{D}$ satisfies this property. For this purpose, by definition of $\mathcal{D}$, we have

$$\begin{aligned}|\mathcal{D}| &= O\left(\log_{1+1/M} \frac{\Delta \cdot \ln(B_{\max})}{\ln(B^+_{\min})}\right)\\ &= O\left(M \cdot \left(\log \Delta + \log \frac{\log(B_{\max})}{\log(B^+_{\min})}\right)\right)\\ &= O\left(\frac{\Delta}{\epsilon} \cdot \ln(B_{\max}) \cdot \left(\log \Delta + \log \frac{\log(B_{\max})}{\log(B^+_{\min})}\right)\right). \quad \square\end{aligned}$$

Proof of Theorem 3, item 2. To prove that $\mathrm{OPT}(\mathrm{IP}_\epsilon) \geq (1-\epsilon) \cdot \mathrm{OPT}(\mathrm{P})$, letting x^* be a fixed optimal solution to (P), we argue that there exists a vector $y = y(x^*)$ such that (x^*, y) is a feasible solution to (IP_ϵ) with an objective value of at least $(1-\epsilon) \cdot \mathrm{OPT}(\mathrm{P})$.

To this end, for every time period t, let

$$y_{tr} = \begin{cases} 1 & \text{if } r = \left\lfloor \sum_{v\in V} \ln(B_{vt}) \cdot x^*_{vt} \right\rfloor_{\mathcal{D}}, \\ 0 & \text{otherwise,} \end{cases}$$

where $\lfloor \cdot \rfloor_{\mathcal{D}}$ is the operator of rounding down to the nearest number in $\mathcal{D}$. It is easy to verify that (x^*, y) is a feasible solution to (IP_ϵ): The constraints (C_1) and (C_2) are clearly satisfied, as they also appear in (P); constraint (C_4) is guaranteed to be satisfied by the way we defined y; and constraint (C_3) is taken care of by the fact that $0 \in \mathscr{D}$. Furthermore, the objective value of (x^*, y) with respect to (IP_ϵ) is precisely

$$\sum_{t=1}^{T} \alpha_t \sum_{r\in\mathscr{D}} (e^r \cdot y_{tr}) = \sum_{t=1}^{T} \alpha_t \cdot \exp\left\{ \left\lfloor \sum_{v\in V} \ln(B_{vt}) \cdot x^*_{vt} \right\rfloor_{\mathscr{D}} \right\}.$$

To attain a lower bound on the latter expression, we prove in Appendix A.2 the following lemma.

Lemma 3. *For every time period t,*

$$\exp\left\{ \left\lfloor \sum_{v\in V} \ln(B_{vt}) \cdot x^*_{vt} \right\rfloor_{\mathscr{D}} \right\} \geq (1-\epsilon) \cdot \prod_{v\in V} B_{vt}^{x^*_{vt}}.$$

As a result, we have just shown that

$$\begin{aligned} \mathrm{OPT}(\mathrm{IP}_\epsilon) &\geq \sum_{t=1}^{T} \alpha_t \cdot \exp\left\{ \left\lfloor \sum_{v\in V} \ln(B_{vt}) \cdot x^*_{vt} \right\rfloor_{\mathscr{D}} \right\} \\ &\geq (1-\epsilon) \cdot \sum_{t=1}^{T} \alpha_t \prod_{v\in V} B_{vt}^{x^*_{vt}} = (1-\epsilon) \cdot \mathrm{OPT}(\mathrm{P}). \end{aligned}$$

To conclude the proof of item 2, it remains to show that any feasible solution (x, y) to (IP_ϵ) can be efficiently translated to (P) without any loss in optimality. Clearly, x must be a feasible solution to (P), as the feasibility set of this problem is contained in that of (IP_ϵ). In addition, the objective value of x with respect to (P) is

$$\begin{aligned} \sum_{t=1}^{T} \alpha_t \prod_{v\in V} B_{vt}^{x_{vt}} &= \sum_{t=1}^{T} \alpha_t \cdot \exp\left\{ \sum_{v\in V} \ln(B_{vt}) \cdot x_{vt} \right\} \\ &\geq \sum_{t=1}^{T} \alpha_t \sum_{r\in\mathscr{D}} (e^r \cdot y_{tr}), \end{aligned}$$

where the latter inequality follows from constraints (C_3) and (C_4). □

It is worth pointing out that we consider in Appendix B an extension of the demand model (5) that includes cross terms—i.e., interactions between pairs of vehicles. We first show that this extension becomes provably harder to approximate. Then, we extend both analytical results on the greedy algorithm and on the approximate IP.

6. Computational Experiments

In this section, we conduct extensive computational experiments to evaluate the algorithms developed in Section 5 and Appendix B on randomly generated data. Specifically, we examine the performance and running time of the greedy algorithm and the approximate IP, both with and without cross terms.

First, we elaborate on the experiments that evaluate the greedy algorithm (Section 5.2) and the approximate integer program (Section 5.3) when there are no cross terms. Our algorithms are compared to the optimal solution, which is computed using exhaustive enumeration over all feasible solutions. This enumeration method becomes impractical for medium- to large-scale instances from practice, as the running time scales exponentially with the number of promotion vehicles. For this reason, we consider a setting with $T = 13$ time periods and $|V| = 5$ vehicles in our computational experiments. Typically, supermarkets make decisions for a selling season of one quarter composed of 13 weeks. As a consequence, to allow brute force enumeration, we have to limit the number of vehicles to five, even though this number can easily exceed 20 in practice (see additional details in Section 7).

Subsequently, we analyze the performance of the greedy algorithm and the approximate integer program in the presence of cross terms. In this case, we compare the two algorithms without cross terms to the integer program with cross terms (see Appendix B.4). From this comparison, we draw valuable insights into the extent of potentially lost profit when cross terms are present but ignored in the model.

The experiments described in this section were run on a standard desktop computer with an Intel Core i5-4690K@3.5 GHz CPU and 8 GB of RAM. The greedy algorithm and the exhaustive enumeration method were coded using Julia, whereas the approximate IP with and without cross terms was solved with Gurobi 6.0.2.

6.1. Performance Without Cross Terms

We first test our algorithms on a base setting, and later extend this analysis to additional settings. Here, we assume that $L_t = 2$ for every time period t and $C_v = 2$ for every vehicle v, while the parameters α_t and B_{vt} are drawn from a uniform distribution on the interval $[1, 2]$. For the precision parameter ϵ of the approximate IP, we experiment with the values 0.5, 0.25, 0.1, and 0.05. Table 2 presents both the running times (average and maximum) and performance ratios (average and minimum) over 200 random instances for $\epsilon = 0.5$ and $\epsilon = 0.25$, and over 10 instances for $\epsilon = 0.10$ and $\epsilon = 0.05$. Here, the performance ratio is defined as the objective value of our method (greedy algorithm or approximate IP) divided by the optimal objective value, which is computed through enumeration.

First, we note that the greedy algorithm and approximate IP perform well and outperform their theoretical guarantees over all instances. Additionally, we observe that the greedy algorithm runs extremely fast, in under a tenth of a second on all tested instances. Its running time is also significantly faster than that of the approximate IP, which slows down considerably as ϵ decreases (i.e., the $1-\epsilon$ guarantee improves) and becomes impractical for $\epsilon < 0.25$. The reason is that even though the size of the approximate IP grows polynomially in the input size and $1/\epsilon$, it remains an integer program.

Table 2. Performance and Running Time of the Greedy Algorithm and Approximate IP for Different Guarantees ($T = 13, |V| = 5, L_t = 2, C_v = 2, \alpha_t \sim U[1, 2], B_{vt} \sim U[1, 2]$)

	Running time in seconds		Performance ratio	
Algorithm	Average	Maximum	Average	Minimum
Greedy algorithm	0.002	0.09	0.9849	0.9200
Approximate IP ($\epsilon = 0.50$)	3.40	16.42	0.9937	0.9520
Approximate IP ($\epsilon = 0.25$)	61.27	3,726.82	0.9979	0.9814
Approximate IP ($\epsilon = 0.10$)	18,215.30	58,011.31	0.9997	0.9981
Approximate IP ($\epsilon = 0.05$)	45,129.46	223,221.91	1	1

That said, in practice, IPs are frequently terminated after a fixed time limit or when reaching a predefined number of iterations, often yielding near-optimal solutions. In the following, we investigate how the IP performs when the termination time is set to one second, five seconds, one minute, five minutes, and 10 minutes. It is worth noting that premature termination of the approximate IP removes our theoretical guarantee of $1 - \epsilon$ on its worst-case performance. Regardless, preliminary experimentation showed that $\epsilon = 0.05$ yields the best results for these time limits. The results over 50 random instances are presented in Table 3.

The results show that the greedy algorithm performs well on average, within 2% of optimal; even in the worst case, its optimality gap is 8%. Additionally, these results show that the approximate IP yields solutions with similar performance as the greedy algorithm, when terminated after one minute. It is important to note that the performance improves significantly before the one minute mark, after which it grows slowly as the termination time is increased. Therefore, we terminate the approximate IP with $\epsilon = 0.05$ after one minute in further experiments.

So far, the parameters L_t and C_v were constant in all of our experiments. We next examine settings where L_t and C_v are drawn from discrete uniform distributions. Specifically, all L_t values are uniformly distributed on $\{1, \ldots, 5\}$, and all C_v are uniformly distributed on $\{1, \ldots, 13\}$. Since $T = 13$ and $|V| = 5$, these ranges allow for all values that L_t and C_v can possibly take. In Table 4, we present the results over 200 random instances.

Table 4 demonstrates that the approximate IP, terminated at one minute, outperforms the greedy algorithm when L_t and C_v are randomly distributed, rather than being constant. On average, the solution of the approximate IP provides a performance guarantee within 0.8% relative to optimal, and its worst-case performance is within 6.4% of optimal. Note that the greedy algorithm is faster and still performs well, with an average performance within 2.9% of optimal. However, its worst-case performance is roughly within 10% of optimal, which is greatly improved by the terminated approximate IP.

Finally, the second part of Table 4 considers the case where α_t and B_{vt} are drawn from a discrete uniform distribution on $\{1, 2\}$, in contrast to the first part, where these parameters are drawn uniformly on $[1, 2]$. In this case, the average and worst-case performance of the greedy algorithm drop significantly, since any error by the greedy algorithm (in comparison to the optimal solution) leads to a larger loss in the objective function relative to the setting where the parameters are drawn from a continuous uniform distribution. On the other hand, the terminated approximate IP performs very well, and in fact, computes an optimal solution in all of the 200 instances. More generally, the approximate IP appears to outperform the greedy algorithm for settings where the boost parameters B_{vt} have different magnitudes. This situation may be relevant in practice, where often one or two promotion vehicles (e.g., flyers) yield a significantly larger boost in demand relative to other vehicles. An additional advantage of the

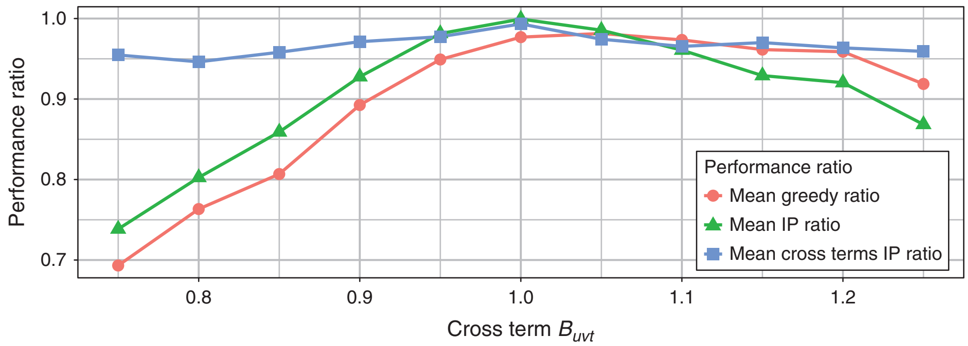
Table 3. Performance of the Greedy Algorithm and Approximate IP for Different Termination Times ($T = 13, |V| = 5, L_t = 2, C_v = 2, \alpha_t \sim U[1, 2], B_{vt} \sim U[1, 2], \epsilon = 0.05$)

	Performance ratio	
Algorithm	Average	Minimum
Greedy algorithm	0.9849	0.9200
Approximate IP (Limit: 1 s)	0.8264	0.7330
Approximate IP (Limit: 5 s)	0.9014	0.7876
Approximate IP (Limit: 1 m)	0.9813	0.9316
Approximate IP (Limit: 5 m)	0.9932	0.9712
Approximate IP (Limit: 10 m)	0.9979	0.9779

Table 4. Performance of the Greedy Algorithm and Approximate IP on Fully Randomized Instances ($T = 13, |V| = 5, L_t \sim U\{1, \ldots, 5\}, C_v \sim U\{1, \ldots, 13\}, \epsilon = 0.05$)

	Performance ratio	
	Average	Minimum
Algorithm with $\alpha_t, B_{vt} \sim U[1, 2]$		
Greedy algorithm	0.9717	0.9026
Approximate IP (Limit: 1 m)	0.9926	0.9365
Algorithm with $\alpha_t, B_{vt} \sim U\{1, 2\}$		
Greedy algorithm	0.9200	0.7407
Approximate IP (Limit: 1 m)	1	1

Figure 2. (Color online) Average Performance Ratios of the Greedy Algorithm, Approximate IP Without Cross Terms, and Approximate IP with Cross Terms Relative to the Optimal Objective



approximate IP approach is the ability to easily incorporate various linear constraints into the formulation, whereas the greedy algorithm is devised specifically for the basic model.

6.2. Performance with Cross Terms

Next, we consider the setting with cross-term effects where pairs of vehicles can cannibalize or complement each other. To analyze the impact of ignoring cross terms, we compare the solutions obtained by the greedy algorithm, the approximate IP that ignore cross terms, and the one that incorporates them (see Appendix B.4). To make a meaningful comparison, the solutions of all three algorithms are evaluated with respect to the optimal objective where the cross terms are present.

Generally, the parameter values remain as before, but we consider several settings for the newly introduced cross terms B_{uvt}. For simplicity, each setting is associated with a different fixed cross-term value ranging from 0.75 to 1.25 in increments of 0.05, and for each setting, we test 200 instances. Figure 2 presents the average performance ratios of the greedy algorithm, the approximate IP without cross terms, and the approximate IP with cross terms when we vary the cross-term values B_{uvt}. Here, the performance ratio is defined as the objective value of a given method divided by the optimal objective, obtained via exhaustive enumeration that takes cross terms into account.

As expected, Figure 2 confirms that the approximate IP without cross terms nearly coincides with the approximate IP with cross terms when B_{uvt} is close to 1. Small differences are incurred due to terminating these algorithms early. Generally, we observe that the performance of both algorithms without cross terms deteriorates as the cross terms become more significant. Surprisingly, the average performance of these algorithms declines more rapidly under cannibalizing cross terms ($B_{uvt} \leq 1$), relative to the case of complementary cross terms ($B_{uvt} \geq 1$). The former corresponds to scenarios where the effect of assigning a pair of vehicles is smaller when compared to the individual boosts of the promotion vehicles, whereas the latter corresponds to a situation in which assigning both vehicles simultaneously yields a larger boost. This suggests that both algorithms are rather robust to complementary cross terms. In particular, the greedy algorithm that ignores cross terms remains within 5% of the approximate IP that takes into account cross terms, even when the cross terms far exceed 1. On the other hand, this analysis highlights the importance of developing and employing algorithms that account for strong adverse cross terms. These tests show that the relative performance of both algorithms decreases rapidly when cross terms become smaller than 0.9, where we may lose more than 20% in profit.

Through experience and historical data, retailers often have a good understanding of the interaction effects between two vehicles and can use this information to assess which model to use. For settings with cannibalization effects ($B_{uvt} \leq 1$), one should preferably use the method that takes cross terms into account. Otherwise, the retailer can experience large profit losses. For settings with complimentary effects ($B_{uvt} \geq 1$), ignoring cross terms may be acceptable.

7. Case Study

In what follows, we study how our methodology performs in practice based on an actual case study. Our data set is described in Section 4 and consists of data collected from 18 stores of a large supermarket client of the Oracle Retail Global Business Unit. We begin by presenting the estimation methods of the model parameters. Then, we apply our algorithms to the resulting model and discuss the potential impact for the retailer.

7.1. Promotion Vehicle Boost Estimation

As shown in Section 4.2, the multiplicative model yields a good predictive accuracy out-of-sample (with

R^2 between 0.68 and 0.94 for the three items we considered). Specifically, we focus our attention on the following log-linear demand model, similar to the one described in (4):

$$\log(d_t^{s,i}) = \beta_0^{s,i} + \beta_1 t + \beta_2^i \log(p_t^{s,i}) + \beta_3^i \log(p_{t-1}^{s,i}) + \sum_{v=1}^{21} \gamma_v x_{vt}^{s,i} + \epsilon_t^{s,i}. \quad (6)$$

Here, $\beta_0^{s,i}$ denotes the store-item intercept, β_1 represents the trend coefficient, and $\epsilon_t^{s,i}$ are i.i.d. normally distributed random variables for all observations. In this model, the demand at time t depends on the item's current and past prices in the store $(p_t^{s,i}, p_{t-1}^{s,i})$, whose effects correspond to the parameters β_2^i and β_3^i, as well as on the item's different promotion vehicles in the store $(x_{vt}^{s,i})$, corresponding to the parameters γ_v. We are interested in the boost parameter of vehicle v at time t given by $B_v = e^{\gamma_v}$. As mentioned in Section 4.2, we assume that the boost of vehicle v is time independent—i.e., $\gamma_{vt} = \gamma_v$ for all t—as our data set is too sparse to accurately estimate a different γ_{vt} for each t. Note that we observed very few instances in our data set where more than one vehicle was used for the same item at the same time. As a result, we are unable to reliably estimate the cross effects between promotion vehicles (this extension of our model is described in Appendix B).

After concluding that the multiplicative model yields a better fit to the data, we decided to reestimate the parameters using the entire data set to identify $B_1 = e^{\gamma_1}, \ldots, B_{21} = e^{\gamma_{21}}$. In Table 5, we present the estimates $\hat{\gamma}_v$ and boost estimates $\hat{B}_v = e^{\hat{\gamma}_v}$ for seven out of the 21 vehicles present in our data. Out of 21 vehicles, only five vehicles are not significant at the 0.05 level, although three of them are significant at the 0.1 level. The vehicles that are not listed in Table 5 have smaller boosts in general, with the lowest value being 1.0134 (corresponding to one of the two insignificant vehicles). For vehicles that are statistically significant or near significant, the smallest boost is 1.0722. This confirms that the assumption $B_v \geq 1$ is satisfied by all estimates.

Table 5. Estimated Promotion Vehicle Parameters and Respective *p*-Values

Promotion vehicle	$\hat{B}_v$	$\hat{\gamma}_v$	*p*-value
(1) *Mailing Coupon*	1.2294	0.2065	$3.50 \cdot 10^{-3}$
(2) *Flyer Front*	1.3731	0.3171	$<2 \cdot 10^{-16}$
(3) *Flyer Mid*	1.8315	0.6051	$<2 \cdot 10^{-16}$
(4) *Flyer End*	1.7702	0.5711	$<2 \cdot 10^{-16}$
(5) *Display*	1.3843	0.3252	$1.41 \cdot 10^{-9}$
(6) *Bonus Snack*	1.5915	0.4647	$1.46 \cdot 10^{-9}$
(7) *TV Commercial*	1.3039	0.2654	$2.48 \cdot 10^{-9}$

According to the boost estimates, promoting the product in a flyer ($\hat{B}_3$ and $\hat{B}_4$) is clearly the most impactful promotion vehicle (depending on position), increasing sales by 83% or 77% relative to assigning no promotion vehicles and keeping all other factors unchanged. Additionally, the 59% increase in sales when a bonus snack is offered with soft drinks ($\hat{B}_6$) indicates that this is an effective vehicle as well. The other vehicles have smaller effects: between 22% and 38% relative increase in sales. This information is likely to be useful to the retailer as it sheds light on the impact of each vehicle on the weekly demand for a particular product. Since retailers have limitations on the number of vehicles that can be used in practice, this information may be crucial when vehicles have to be selected on a short-term notice.

Finally, having estimated the boosts B_v, we are left with estimating the parameters α_t. The time-dependent parameter α_t resembles the profit margin at time t multiplied by the demand term, which is affected by the price at time t. Thus, for every store s and item i individually, we compute the estimates $\hat{\alpha}_t$ as follows:

$$\hat{\alpha}_t = (p_t^{s,i} - c_t^{s,i}) \cdot \exp\{\hat{\beta}_0^{s,i} + \hat{\beta}_1 t + \hat{\beta}_2^i \log(p_t^{s,i}) + \hat{\beta}_3^i \log(p_{t-1}^{s,i})\},$$

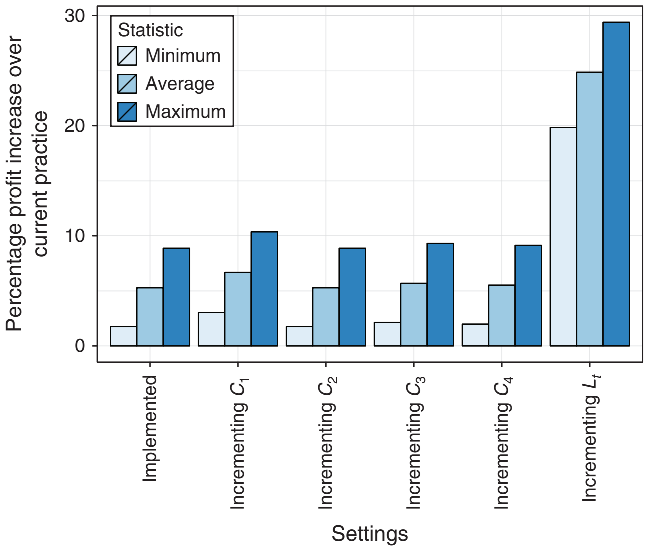
where $p_t^{s,i}$ and $c_t^{s,i}$ are the unit price and unit cost of item i in store s at time t.

7.2. Promotion Vehicle Optimization

To assess the impact of our methods, we optimize the vehicle assignments for one representative item over one year (i.e., $T = 52$ weeks). Prior to optimizing, we formulate the promotion vehicle scheduling problem by plugging the item's parameter estimates $\hat{\alpha}_t$ and $\hat{B}_v$ into the objective function. Additionally, the parameters L_t and C_v are set to their implemented values in the data set. In addition to this initial comparison with the implemented L_t and C_v, we will perform a sensitivity analysis on L_t and C_v to infer the variation in profits when additional vehicle assignments are allowed. Note that, in our case study, business rules of the third type (i.e., the requirement that a particular promotion vehicle has to be used at a specific time period) are not present. However, as we mentioned before, one can easily incorporate such rules via basic modifications.

Notably, as we optimize vehicle assignments over 52 weeks, it is impossible to compute the optimal solution using brute force enumeration. The subset IP formulation mentioned in Section 4.3 cannot be solved either, as it involves $2^{|V|} \cdot T = 2^{21} \cdot 52$ binary variables. Interestingly, we observed that the greedy algorithm and the approximate IP identified precisely the same vehicle assignments in all settings tested. In Figure 3, we present the resulting profit increases our model attains over the current practice in six different stores.[2] In each of these settings, the three vertical bars correspond

Figure 3. (Color online) Minimum, Average, and Maximum Percentage Profit Increase for Different Settings



to the minimum, average, and maximum percentage increase in profits over the six stores.

First, in the leftmost column, we consider optimizing the setting where the implemented L_t and C_v are used. This setting shows the impact that our optimization model can have on current practice. In particular, the average increase in profits over the six stores is slightly larger than 5%, which is a considerable increase in the grocery industry where profit margins are small. Over all six stores, the impact ranges from 2% to 9%. Consequently, even in the store with the least impact, using our model to optimize the promotion vehicles schedule yields a significant profit improvement of at least 2%. Note that this projected improvement assumes that there are no cross terms in the demand model. As mentioned earlier, our data set includes relatively few instances where the same pair of vehicles was used for the same item at the same time. As a result, there is no reliable way to estimate the cross terms by using this data set. Nevertheless, it is worth mentioning that the tests conducted in Section 6.2 suggest that the approximate IP provides results that seem to be robust to the presence of cross terms (assuming their magnitude is not very far from 1, which seems to be a reasonable assumption).

We next investigate the effect an increase in any of the C_v or L_t parameters can have on the profit relative to current practice. These what-if scenarios allow category managers to examine how changing the requirements affects future decisions. Since our algorithms run very quickly, one can efficiently test many such scenarios and gain a better understanding of the impact that varying some of the business rules can have.

The second to fifth settings in Figure 3 demonstrate the effect of, respectively, increasing the capacity of *Flyer Mid* (C_1), *Display* (C_2), *Bonus Snack* (C_3), and *TV Commercial* (C_4) by one unit over the entire year. To perform a fair comparison, we maintain the capacity L_t over this planning horizon as before. The figure illustrates that an increase in vehicle capacity leads to relatively small additional profits when compared to current capacities, which is to be expected when only one additional unit is allowed over an entire year. Nevertheless, this analysis can be very useful in deciding which vehicle capacity to increase. In our case, increasing the maximal number of allowed flyer promotions C_1 is the most profitable strategy.

The rightmost setting in Figure 3 shows the impact of increasing the capacity L_t by one unit for all time periods. To perform a fair comparison, we maintain all vehicle capacities $(C_1, \ldots, C_{21})$ as before, so that the total number of vehicles available over the entire year is the same in both scenarios. This alteration leads to a dramatic profit improvement relative to current practice. On average, increasing L_t by one unit in all time periods changes a 5% average profit gain into an approximately 25% profit increase relative to current practice. In particular, increasing time capacities allows the optimization model to assign several vehicles simultaneously and, consequently, to take advantage of the multiplicative effect in demand.

8. Concluding Remarks

In this paper, we introduce and study the problem of scheduling promotion vehicles, faced by supermarket category managers who wish to decide on how to spread multiple promotion vehicles over a finite planning horizon, so as to maximize profits. In this setting, our problem formulation incorporates important business rules from practice. Motivated by real data, we focus on a class of demand models in which promotion vehicles have a multiplicative effect on demand. We then show that the resulting optimization problem is NP-hard and, furthermore, cannot be efficiently approximated within some absolute constant. This intractability result leads us to present two algorithmic approaches: an efficient greedy algorithm with an approximation ratio of $\Delta + 1$, where Δ stands for the maximum number of vehicles that can be assigned at any period, and a polynomial-size integer program that yields a $1 - \epsilon$ approximation. Finally, we compare both approaches computationally in terms of performance and running time, along with a case study using data from an Oracle client. Under our model assumptions and for a particular item considered in our case study, these tests indicate that this optimization model can lead to a profit increase of 2% to 9% over current practice. In addition, the models and algorithms developed in this paper can be used to draw practical insights on the effects of promotion vehicles on demand and profits. Given the scalability of our approach, the retailer can use these methods to test various strategies, and to select the best promotion schedule for the upcoming selling season.

Open complexity question. As previously mentioned, we proposed two approximation algorithms for the most general formulation of the promotion vehicle scheduling problem. In addition, we developed a PTAS for the special case of uniform vehicle boosts and uniform base profits of time periods (see Online Appendix D)—i.e., $B_{vt} = B$ for any vehicle v and time period t, and in addition, $\alpha_1 = \cdots = \alpha_T$. Consequently, an interesting open question for future research is whether improved approximation guarantees (possibly a PTAS) can be obtained for a broader class of instances. For example, one can consider the case where all time periods have uniform base profits ($\alpha_1 = \cdots = \alpha_T$), with time-independent vehicle boosts ($B_{vt} = B_v$).

Handling multiple products. The fundamental problem considered in this paper focuses on efficiently scheduling promotion vehicles for a single product. However, one may be interested in extending the analysis to multiple products. Interestingly, our approximate integer programming approach (see Section 5.3 and Appendix B.4) is flexible enough to be leveraged into the multiproduct setting. Here, different products are related through additional capacity constraints, placing an upper bound on the number of products to which any vehicle can be assigned at any time period (e.g., "at most 4 products fit into a flyer advertisement at time period 10"). To capture this setting, we simply augment the original decision variables with a product index, so that x^i_{vt} now indicates whether vehicle v is assigned to product i at time period t. With these new variables, it remains to duplicate our current IP formulation over all products, and incorporate constraints of the form $\sum_i x^i_{vt} \leq L_{vt}$ to enforce the additional (cross-item) capacity constraints. Consequently, we can further extend our approximate IP results to this general setting.

Acknowledgments

The authors thank the department editor as well as the anonymous associate editor and reviewers for helpful comments and suggestions that helped improve the content and exposition of this paper.

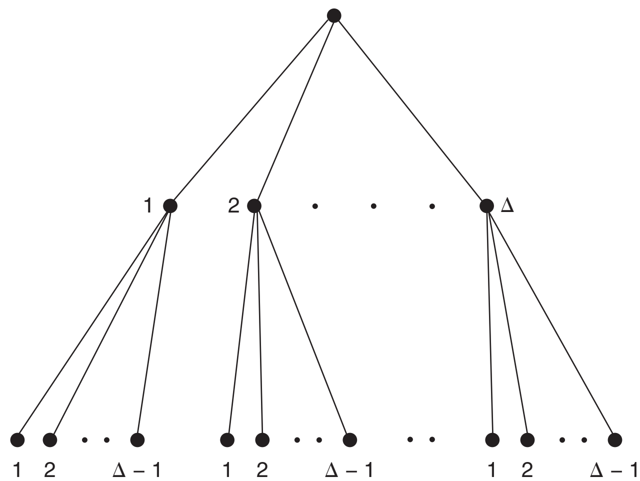
Appendix A. Additional Proofs from Section 5

A.1. Proof of Lemma 2

To better understand our construction, we advise the reader to consult Figure A.1, in which a depth-2 rooted tree $T = (N, E)$ is drawn. Here, the root r has Δ children, each of which has $\Delta - 1$ children of its own. With respect to this tree, we define an instance of the multiplicative model as follows:

- The set of vehicles is E, while the set of time periods is N. That is, the edges and vertices of T serve as vehicles and time periods, respectively.

Figure A.1. A Graph-Based Illustration of the Tight Example



- The base profit of the root r is $\alpha_r = 1 + 1/M$, where $M \geq \Delta$ is a parameter whose precise meaning will be explained shortly. In addition, the base profit of any other vertex v is $\alpha_v = 1$. All vertices, including the root, have capacity Δ.
- Each edge $e = (u, v)$ has a boost of M when assigned to any of its endpoints, u and v—that is, $B_{e,u} = B_{e,v} = M$. On the other hand, this edge has a boost of 1 when assigned to any other vertex. All edges have unit capacities, meaning that each edge can be assigned only once.

Let us first examine how the greedy algorithm operates. In step $s = 1$, we assign to the root r all Δ edges adjacent to it, obtaining a profit of $(1 + 1/M) \cdot M^{\Delta}$. From this point on, these edges are no longer available, as their remaining capacity becomes zero. Therefore, in steps $s = 2, \ldots, \Delta + 1$, we assign to each child of the root r all $\Delta - 1$ edges underneath it, with a combined profit of $\Delta \cdot M^{\Delta-1}$. From that point on, all edges have zero capacity, meaning that each of the $\Delta(\Delta - 1)$ leaves is not assigned any edge, making their combined profit $\Delta(\Delta - 1)$. To summarize, the greedy algorithm ends up with a total profit of

$$\text{Greedy}_{\Delta}(M) = \left(1 + \frac{1}{M}\right) \cdot M^{\Delta} + \Delta \cdot M^{\Delta-1} + \Delta(\Delta - 1).$$

However, one feasible solution is that of assigning, to each child of the root r, all Δ edges adjacent to it, showing that $\text{OPT}_{\Delta}(M) \geq \Delta \cdot M^{\Delta}$. Consequently, the asymptotic ratio between the profit obtained by the greedy algorithm and the optimal profit, as the parameter M tends to infinity, is

$$\lim_{M\to\infty} \frac{\text{Greedy}_{\Delta}(M)}{\text{OPT}_{\Delta}(M)} \leq \lim_{M\to\infty} \frac{(1 + 1/M) \cdot M^{\Delta} + \Delta \cdot M^{\Delta-1} + \Delta(\Delta - 1)}{\Delta \cdot M^{\Delta}} = \frac{1}{\Delta}.$$

A.2. Proof of Lemma 3

We say that period t is x^*-bad when $\prod_{v\in V} B_{vt}^{x^*_{vt}} = 1$, meaning that this period is either not assigned any vehicle or assigned only vehicles with $B_{vt} = 1$. In the opposite case, period t is called x^*-good, and we clearly have $\prod_{v\in V} B_{vt}^{x^*_{vt}} \in [B^+_{\min}, B^{\Delta}_{\max}]$. The proof proceeds by considering two cases, depending on whether period t is x^*-bad or x^*-good.

Case* 1: *t is x^*-bad. Here, since $0 \in \mathcal{D}$, we have by definition

$$\exp\left\{\left\lfloor \sum_{v\in V} \ln(B_{vt})\cdot x^*_{vt}\right\rfloor_{\mathcal{D}}\right\} \geq 1 = \prod_{v\in V} B_{vt}^{x^*_{vt}},$$

and the claim follows.

Case* 2: *t is x^*-good. Based on the construction of $\mathcal{D}$, we have

$$\begin{aligned}\exp&\left\{\left\lfloor \sum_{v\in V} \ln(B_{vt})\cdot x^*_{vt}\right\rfloor_{\mathcal{D}}\right\}\\ &\geq \exp\left\{\left(1+\frac{1}{M}\right)^{-1}\cdot \sum_{v\in V}\ln(B_{vt})\cdot x^*_{vt}\right\}\\ &\geq \exp\left\{\left(1-\frac{1}{M}\right)\cdot \sum_{v\in V}\ln(B_{vt})\cdot x^*_{vt}\right\}\\ &= \exp\left\{\left(1-\frac{\epsilon}{\Delta\cdot\ln(B_{\max})}\right)\cdot \sum_{v\in V}\ln(B_{vt})\cdot x^*_{vt}\right\}\\ &\geq e^{-\epsilon}\cdot\prod_{v\in V} B_{vt}^{x^*_{vt}}\\ &\geq (1-\epsilon)\cdot\prod_{v\in V} B_{vt}^{x^*_{vt}},\end{aligned}$$

where the equality above holds since $M = (\Delta/\epsilon)\cdot\ln(B_{\max})$, and the subsequent inequality is obtained by observing that $\sum_{v\in V}\ln(B_{vt})\cdot x^*_{vt} \leq \Delta\cdot\ln(B_{\max})$.

Appendix B. Extension to Cross Terms

B.1. Multiplicative Model with Cross Terms

Very often, using two vehicles simultaneously may induce an additional impact on demand, called a cross effect. For example, by broadcasting a TV commercial (vehicle 1), one can obtain the relative increase of $B_1 = 1.15$ in demand, and by distributing in-store flyers (vehicle 2), demand will be boosted by $B_2 = 1.07$. However, if these two vehicles are scheduled at the same time, they may cannibalize or complement each other, since some customers will be affected by both promotion vehicles. To capture this phenomenon, we introduce cross-term effects for each pair of vehicles $u < v$ and for each time period t, denoted by the parameters B_{uvt}. Note that if $B_{uvt} < 1$, vehicles u and v cannibalize each other (i.e., using them simultaneously has a lower effect than the product of using them separately), and if $B_{uvt} > 1$, vehicles u and v complement each other (i.e., using them simultaneously has a greater effect than the product of using them separately). Note that, to estimate these cross-effect parameters, one needs enough historical data on occurrences of having the two vehicles simultaneously. This extension of our problem can be very important in some practical settings as these cross-term effects may significantly change the optimal vehicle scheduling policy.

In this extension of the basic model, the promotion vehicle scheduling problem becomes

$$\begin{aligned}&(\mathrm{P_{CT}}) && \max \sum_{t=1}^{T}\alpha_t \prod_{v\in V} B_{vt}^{x_{vt}} \prod_{u<v} B_{uvt}^{x_{ut}x_{vt}}\\ &(\tilde{\mathrm{C}}_1) && \sum_{t=1}^{T} x_{vt} \leq C_v \quad \forall\, v\in V,\\ &(\tilde{\mathrm{C}}_2) && \sum_{v\in V} x_{vt} \leq L_t \quad \forall\, t\in[T],\\ &(\tilde{\mathrm{C}}_3) && x_{vt}\in\{0,1\} \quad \forall\, v\in V,\ t\in[T].\end{aligned}$$

As we will see shortly, this variant becomes provably harder to approximate in comparison to the basic model (P), while still allowing us to extend the main results of Section 5.

B.2. Inapproximability Results

In what follows, we prove that by incorporating cross terms into the objective function, one makes the multiplicative model significantly harder to deal with. To better understand the inherent difficulty in handling cross terms, it is worth mentioning that, for establishing the APX-hardness results in Section 5.1, our reduction mapped instances of Max-IS$_\Delta$ into ones of the basic multiplicative model with an arbitrary number of time periods. In contrast, we show that in the presence of cross terms, stronger inapproximability bounds can be obtained, even for a single time period.

For this purpose, we describe a simple gap-preserving reduction from the maximum independent set problem in general graphs (in contrast to regular graphs, as in Section 5.1). In this context, it is NP-hard to distinguish between graphs containing an independent set of cardinality $\Omega(|N|^{1-\epsilon})$ and graphs where the size of any independent set is $O(|N|^{\epsilon})$, for any fixed $\epsilon > 0$ (Håstad 1996). Here, N stands for the set of vertices in the underlying graph.

Theorem 4. *Even for a single time period, it is NP-hard to approximate the multiplicative model with cross terms within factor $O(B_{\max}^{O(|V|^{1-\epsilon})})$, for any fixed $\epsilon > 0$, where $B_{\max}$ is the maximum boost of any vehicle.*

Proof. Given an instance $G = (N, E)$ of the maximum independent set problem, consisting of a general undirected graph on n vertices, we create a corresponding instance of the multiplicative model with cross terms as follows:

- The set of vehicles is N, where each vehicle v has a unit capacity (i.e., $C_v = 1$).
- There is a single time period, with a base profit of $\alpha_1 = 1$ and a capacity of $L_1 = n$.
- Each vehicle $v \in N$ has a boost of $B_{v,1} = B > 1$. However, we fix the cross term $B_{u,v,1}$ of each pair of vehicles $u \neq v$ to be

$$B_{u,v,1} = \begin{cases} 1 & \text{if } (u,v)\notin E,\\ 1/B^{2n} & \text{if } (u,v)\in E.\end{cases}$$

That is, this term is neutral when u and v are not joined by an edge in G. Otherwise, when u and v are adjacent, this term is sufficiently small so that the entire profit is cannibalized.

In other words, we have just created the following instance:

$$\begin{aligned}&(\mathrm{P_{CT}}) && \max \prod_{v\in N} B^{x_{v,1}}\cdot \prod_{(u,v)\in E} B^{-2n\cdot x_{u,1}x_{v,1}}\\ &(\tilde{\mathrm{C}}_1) && x_{v,1}\leq 1 \quad \forall\, v\in N,\\ &(\tilde{\mathrm{C}}_2) && \sum_{v\in N} x_{v,1}\leq n,\\ &(\tilde{\mathrm{C}}_3) && x_{v,1}\in\{0,1\} \quad \forall\, v\in N.\end{aligned}$$

We proceed by showing that, letting U^* be a maximum-cardinality independent set in G, our reduction guarantees that

$$\mathrm{OPT}(\mathrm{P_{CT}}) = \begin{cases}\Omega(B^{\Omega(n^{1-\epsilon})}) & \text{when } |U^*| = \Omega(n^{1-\epsilon}),\\ O(B^{O(n^{\epsilon})}) & \text{when } |U^*| = O(n^{\epsilon}).\end{cases}$$

Indeed, when $|U^*| \geq \mu \cdot n^{1-\epsilon}$ for some constant $\mu > 0$, consider the solution obtained by setting $x_{v,1} = 1$ if and only if $v \in U^*$. The resulting objective value is precisely

$$\prod_{v \in N} B^{x_{v,1}} \cdot \prod_{(u,v) \in E} B^{-2n \cdot x_{u,1} x_{v,1}} = B^{|U^*|} \geq B^{\mu \cdot n^{1-\epsilon}},$$

where the first equality holds since $x_{u,1} x_{v,1} = 0$ for every $(u,v) \in E$, or otherwise, two vertices in U^* must be connected by an edge, meaning that U^* cannot be an independent set.

On the other hand, suppose that $|U^*| \leq \mu \cdot n^{\epsilon}$ for some constant $\mu > 0$. Here, any solution where at least $|U^*| + 1$ of the variables $\{x_{v,1}\}_{v \in N}$ take a value of 1 necessarily has $x_{u,1} x_{v,1} = 1$ for some $(u,v) \in E$, or otherwise, U^* cannot be a maximum-cardinality independent set. However, when that happens, $\mathrm{OPT}(\mathrm{P_{CT}}) \leq B^n \cdot B^{-2n} = B^{-n}$. In the opposite scenario, at most $|U^*|$ of the variables take a value of 1, meaning that the objective value can be upper bounded by

$$\prod_{v \in N} B^{x_{v,1}} \cdot \prod_{(u,v) \in E} B^{-2n \cdot x_{u,1} x_{v,1}} \leq B^{|U^*|} \leq B^{\mu \cdot n^{\epsilon}}. \quad \square$$

B.3. Applicability of the Greedy Algorithm

A careful inspection of the greedy algorithm proposed in Section 5.2 reveals that our analysis holds in a much broader setting. In fact, the only place we used the explicit product-form contribution of each period t to the objective function, $\alpha_t \prod_{v \in V} B_{vt}^{x_{vt}}$, is in computing the best subset of vehicles to pick so that the latter expression is maximized. However, this analysis works even when each period t has its own objective function F_t: $2^V \to \mathbb{R}_+$, specifying an arbitrary nonnegative contribution for each subset of vehicles assigned to time t. In particular, the latter function could incorporate cross terms, taking the form

$$F_t(U) = \alpha_t \prod_{v \in U} B_{vt} \prod_{u < v \in U} B_{uvt}.$$

For this reason, the multiplicative model with cross terms can also be approximated within factor $\Delta + 1$, where $\Delta = \max_t L_t$, as long as we can efficiently optimize the function $F_t(\cdot)$. One particularly interesting case where this is indeed possible is when $\Delta = O(1)$, where this function can be optimized by enumerating all $|V|^{L_t} \leq |V|^{\Delta}$ subsets of vehicles with cardinality at most L_t.

B.4. Approximate IP with Cross Terms

This section is dedicated to proving that, even when cross-terms are incorporated into the objective function, the multiplicative model can still be formulated as an approximate integer program of polynomial size.

Theorem 5. *Given an accuracy parameter $\epsilon > 0$, we can efficiently construct an integer program $(\mathrm{IP}_{\mathrm{CT},\epsilon})$ that satisfies the following properties:*

1. *The combined number of variables and constraints in $(\mathrm{IP}_{\mathrm{CT},\epsilon})$ is polynomial in the input size of $(\mathrm{P_{CT}})$ and in $1/\epsilon$.*
2. *$(\mathrm{IP}_{\mathrm{CT},\epsilon})$ provides a $(1-\epsilon)$-approximation to $(\mathrm{P_{CT}})$. That is, $\mathrm{OPT}(\mathrm{IP}_{\mathrm{CT},\epsilon}) \geq (1-\epsilon) \cdot \mathrm{OPT}(\mathrm{P_{CT}})$, and moreover, any solution to $(\mathrm{IP}_{\mathrm{CT},\epsilon})$ can be efficiently translated to $(\mathrm{P_{CT}})$ without any loss in optimality.*

To avoid redundancies, since the basics of our approach are thoroughly discussed in Section 5.3, we focus on highlighting the main obstacles in handling cross terms and explain how these are resolved.

Ingredient 1: The integer program. Let $\mathcal{D} \subseteq \mathbb{R}_+$ be a finite discretization set, as defined in ingredient 2 below, consisting of nonnegative real numbers, with $0 \in \mathcal{D}$. Based on this set, our integer program $(\mathrm{IP}_{\mathrm{CT},\epsilon})$ is defined as follows:

$$(\mathrm{IP}_{\mathrm{CT},\epsilon}) \quad \max \sum_{t=1}^{T} \alpha_t \sum_{r \in \mathcal{D}} (e^r \cdot y_{tr})$$

$$(\tilde{C}_1) \quad \sum_{t=1}^{T} x_{vt} \leq C_v \quad \forall v \in V,$$

$$(\tilde{C}_2) \quad \sum_{v \in V} x_{vt} \leq L_t \quad \forall t \in [T],$$

$$(\tilde{C}_3) \quad \sum_{r \in \mathcal{D}} y_{tr} = 1 \quad \forall t \in [T],$$

$$(\tilde{C}_4) \quad y_{tr} \leq \frac{1}{r} \left(\sum_{v \in V} \ln(B_{vt}) \cdot x_{vt} + \sum_{u<v} \ln(B_{uvt}) \cdot z_{uvt} \right) \quad \forall t \in [T],\ r \in \mathcal{D} \setminus \{0\},$$

$$(\tilde{C}_5) \quad z_{uvt} \leq x_{ut},\ z_{uvt} \leq x_{vt},\ z_{uvt} \geq x_{ut} + x_{vt} - 1,\ z_{uvt} \geq 0 \quad \forall u, v \in V,\ u < v,\ t \in [T],$$

$$(\tilde{C}_6) \quad x_{vt},\ y_{tr} \in \{0,1\} \quad \forall u \in V,\ t \in [T],\ r \in \mathcal{D}.$$

As in Section 5.3, the binary variable y_{tr} indicates whether we are using e^r to slightly underestimate the boost $\prod_{v \in V} B_{vt}^{x_{vt}} \prod_{u<v} B_{uvt}^{x_{ut} x_{vt}}$ at time t. In addition, z_{uvt} plays the role of $x_{ut} x_{vt}$, to linearize constraint $(\tilde{C}_4)$. It is easy to verify that, for any binary assignment to the x-variables, constraint $(\tilde{C}_5)$ guarantees that $z_{uvt} = x_{ut} x_{vt}$, even without an integrality requirement on z_{uvt}.

Ingredient 2: Defining the set $\mathcal{D}$. In what follows, we use $B_{\max} > 1$ to denote the maximum absolute value of any of the individual boosts B_{vt} and cross terms B_{uvt}, over all periods. Using ideas similar to those of Section 5.3, we begin by initializing $\mathcal{D} = \{0\}$. This set is then augmented by all breakpoints that are created when the interval $[\epsilon/2, \Delta^2 \cdot \ln(B_{\max})] \subseteq (0, \infty)$ is geometrically partitioned by powers of $1 + 1/M$, where $M = (\Delta^2/\epsilon) \cdot \ln(B_{\max})$. With this definition,

$$\mathcal{D} = \left\{ 0, \frac{\epsilon}{2}, \left(1 + \frac{1}{M}\right) \cdot \frac{\epsilon}{2}, \left(1 + \frac{1}{M}\right)^2 \cdot \frac{\epsilon}{2}, \ldots \right\}.$$

Proof of Theorem 5, item 1. To show that the size of $(\mathrm{IP}_{\mathrm{CT},\epsilon})$ is polynomial in the input size of $(\mathrm{P_{CT}})$ and in $1/\epsilon$, it suffices to show that the discretization set $\mathcal{D}$ satisfies this property. For this purpose, by definition of $\mathcal{D}$, we have

$$|\mathcal{D}| = O\left(\log_{1+1/M} \frac{\Delta \cdot \ln(B_{\max})}{\epsilon} \right)$$
$$= O\left(M \cdot \left(\log \Delta + \log \log(B_{\max}) + \log \frac{1}{\epsilon} \right) \right)$$
$$= O\left(\frac{\Delta^2}{\epsilon} \cdot \ln(B_{\max}) \cdot \left(\log \Delta + \log \log(B_{\max}) + \log \frac{1}{\epsilon} \right) \right). \quad \square$$

Proof of Theorem 5, item 2. To prove that $\mathrm{OPT}(\mathrm{IP}_{\mathrm{CT},\epsilon}) \geq (1-\epsilon) \cdot \mathrm{OPT}(\mathrm{P_{CT}})$, letting x^* be a fixed optimal solution to $(\mathrm{P_{CT}})$, we argue that there are vectors $y = y(x^*)$ and $z = z(x^*)$ such that (x^*, y, z) is a feasible solution to $(\mathrm{IP}_{\mathrm{CT},\epsilon})$ with an objective value of at least $(1-\epsilon) \cdot \mathrm{OPT}(\mathrm{P_{CT}})$.

To this end, for every time period t, let

$$y_{tr} = \begin{cases} 1 & \text{if } r = \left\lfloor \sum_{v \in V} \ln(B_{vt}) \cdot x_{vt}^* + \sum_{u<v} \ln(B_{uvt}) \cdot x_{ut}^* x_{vt}^* \right\rfloor_{\mathcal{D}}, \\ 0 & \text{otherwise,} \end{cases}$$

and for every pair of vehicles $u < v$, let $z_{uvt} = x^*_{ut} x^*_{vt}$. Note that, in the optimal solution x^*, we must have the relation $\prod_{v\in V} B_{vt}^{x_{vt}} \prod_{u<v} B_{uvt}^{x_{ut}x_{vt}} \geq 1$ for any time period t, and consequently, $\lfloor \sum_{v\in V} \ln(B_{vt}) \cdot x^*_{vt} + \sum_{u<v} \ln(B_{uvt}) \cdot x^*_{ut} x^*_{vt} \rfloor_{\mathcal{D}}$ is indeed well defined above, as $0 \in \mathcal{D}$. It is easy to verify that (x^*, y, z) is a feasible solution to $(\mathrm{IP}_{\mathrm{CT},\epsilon})$: the constraints $(\tilde{\mathrm{C}}_1)$ and $(\tilde{\mathrm{C}}_2)$ are clearly satisfied, as they also appear in $(\mathrm{P}_{\mathrm{CT}})$; constraint $(\tilde{\mathrm{C}}_4)$ is guaranteed to be satisfied by the way we defined y; constraint $(\tilde{\mathrm{C}}_3)$ is taken care of by the fact that $0 \in \mathcal{D}$; and constraint $(\tilde{\mathrm{C}}_5)$ follows from the definition of z. Furthermore, the objective value of (x^*, y, z) with respect to $(\mathrm{IP}_{\mathrm{CT},\epsilon})$ is precisely

$$\sum_{t=1}^{T} \alpha_t \sum_{r\in\mathcal{D}} (e^r \cdot y_{tr}) = \sum_{t=1}^{T} \alpha_t \cdot \exp\left\{ \left\lfloor \sum_{v\in V} \ln(B_{vt}) \cdot x^*_{vt} + \sum_{u<v} \ln(B_{uvt}) \cdot x^*_{ut} x^*_{vt} \right\rfloor_{\mathcal{D}} \right\}.$$

To derive a lower bound on the latter term, we prove in Appendix B.5 the following lemma.

Lemma 4. *For every time period t,*

$$\exp\left\{ \left\lfloor \sum_{v\in V} \ln(B_{vt}) \cdot x^*_{vt} + \sum_{u<v} \ln(B_{uvt}) \cdot x^*_{ut} x^*_{vt} \right\rfloor_{\mathcal{D}} \right\} \geq (1-\epsilon) \cdot \prod_{v\in V} B_{vt}^{x^*_{vt}} \prod_{u<v} B_{uvt}^{x^*_{ut}x^*_{vt}}.$$

As a result, we have just shown that

$$\begin{aligned} \mathrm{OPT}(\mathrm{IP}_{\mathrm{CT},\epsilon}) &\geq \sum_{t=1}^{T} \alpha_t \exp\left\{ \left\lfloor \sum_{v\in V} \ln(B_{vt}) \cdot x^*_{vt} + \sum_{u<v} \ln(B_{uvt}) \cdot x^*_{ut} x^*_{vt} \right\rfloor_{\mathcal{D}} \right\} \\ &\geq (1-\epsilon) \cdot \sum_{t=1}^{T} \alpha_t \prod_{v\in V} B_{vt}^{x^*_{vt}} \prod_{u<v} B_{uvt}^{x^*_{ut}x^*_{vt}} \\ &= (1-\epsilon) \cdot \mathrm{OPT}(\mathrm{P}_{\mathrm{CT}}). \end{aligned}$$

To conclude the proof of item 2, it remains to show that any feasible solution (x, y, z) to $(\mathrm{IP}_{\mathrm{CT},\epsilon})$ can be efficiently translated to $(\mathrm{P}_{\mathrm{CT}})$ without any loss in optimality. Clearly, x must be a feasible solution to $(\mathrm{P}_{\mathrm{CT}})$, as the feasibility set of this problem is contained in that of $(\mathrm{IP}_{\mathrm{CT},\epsilon})$. In addition, the objective value of x with respect to $(\mathrm{P}_{\mathrm{CT}})$ is

$$\begin{aligned} \sum_{t=1}^{T} \alpha_t \prod_{v\in V} B_{vt}^{x_{vt}} \prod_{u<v} B_{uvt}^{x_{ut}x_{vt}} &= \sum_{t=1}^{T} \alpha_t \cdot \exp\left\{ \sum_{v\in V} \ln(B_{vt}) \cdot x_{vt} + \sum_{u<v} \ln(B_{uvt}) \cdot x_{ut} x_{vt} \right\} \\ &\geq \sum_{t=1}^{T} \alpha_t \sum_{r\in\mathcal{D}} (e^r \cdot y_{tr}), \end{aligned}$$

where the last inequality follows from constraints $(\tilde{\mathrm{C}}_3)$ and $(\tilde{\mathrm{C}}_4)$. □

B.5. Proof of Lemma 4

We say that period t is x^*-bad when $\prod_{v\in V} B_{vt}^{x^*_{vt}} \prod_{u<v} B_{uvt}^{x^*_{ut}x^*_{vt}} \leq e^{\epsilon/2}$. In the opposite case, period t is called x^*-good, and we clearly have $\prod_{v\in V} B_{vt}^{x^*_{vt}} \prod_{u<v} B_{uvt}^{x^*_{ut}x^*_{vt}} \in [e^{\epsilon/2}, B_{\max}^{\Delta^2}]$. The proof proceeds by considering two cases, depending on whether period t is x^*-bad or x^*-good.

Case 1:** t **is** x^***-bad. Here, we have by definition

$$\begin{aligned} \prod_{v\in V} B_{vt}^{x^*_{vt}} \prod_{u<v} B_{uvt}^{x^*_{ut}x^*_{vt}} &\leq e^{\epsilon/2} \\ &\leq 1+\epsilon \\ &\leq \frac{1}{1-\epsilon} \cdot \exp\left\{ \left\lfloor \sum_{v\in V} \ln(B_{vt}) \cdot x^*_{vt} + \sum_{u<v} \ln(B_{uvt}) \cdot x^*_{ut} x^*_{vt} \right\rfloor_{\mathcal{D}} \right\}, \end{aligned}$$

where the second inequality holds since $e^{\epsilon/2} \leq 1+\epsilon$ when $\epsilon \in [0,1]$, and the third inequality holds since the expression within the $\lfloor \cdot \rfloor_{\mathcal{D}}$ operator cannot be negative (otherwise, x^* is not optimal) and since $0 \in \mathcal{D}$. The desired claim follows by rearranging the above inequality.

Case 2:** t **is** x^***-good. Based on the construction of $\mathcal{D}$, we have

$$\begin{aligned} &\exp\left\{ \left\lfloor \sum_{v\in V} \ln(B_{vt}) \cdot x^*_{vt} + \sum_{u<v} \ln(B_{uvt}) \cdot x^*_{ut} x^*_{vt} \right\rfloor_{\mathcal{D}} \right\} \\ &\quad\geq \exp\left\{ \left(1 + \frac{1}{M}\right)^{-1} \cdot \left(\sum_{v\in V} \ln(B_{vt}) \cdot x^*_{vt} + \sum_{u<v} \ln(B_{uvt}) \cdot x^*_{ut} x^*_{vt} \right) \right\} \\ &\quad\geq \exp\left\{ \left(1 - \frac{1}{M}\right) \cdot \left(\sum_{v\in V} \ln(B_{vt}) \cdot x^*_{vt} + \sum_{u<v} \ln(B_{uvt}) \cdot x^*_{ut} x^*_{vt} \right) \right\} \\ &\quad= \exp\left\{ \left(1 - \frac{\epsilon}{\Delta^2 \cdot \ln(B_{\max})}\right) \cdot \left(\sum_{v\in V} \ln(B_{vt}) \cdot x^*_{vt} + \sum_{u<v} \ln(B_{uvt}) \cdot x^*_{ut} x^*_{vt} \right) \right\} \\ &\quad\geq e^{-\epsilon} \cdot \prod_{v\in V} B_{vt}^{x^*_{vt}} \prod_{u<v} B_{uvt}^{x^*_{ut}x^*_{vt}} \\ &\quad\geq (1-\epsilon) \cdot \prod_{v\in V} B_{vt}^{x^*_{vt}} \prod_{u<v} B_{uvt}^{x^*_{ut}x^*_{vt}}, \end{aligned}$$

where the equality above holds since $M = (\Delta^2/\epsilon) \cdot \ln(B_{\max})$, and the subsequent inequality is obtained by observing that $\sum_{v\in V} \ln(B_{vt}) \cdot x^*_{vt} + \sum_{u<v} \ln(B_{uvt}) \cdot x^*_{ut} x^*_{vt} \leq \Delta^2 \cdot \ln(B_{\max})$.

Endnotes

[1] We tried both AIC and BIC sequentially to obtain a robust model in terms of which independent variables are significant. We observed that both criteria yield similar outcomes, suggesting that the set of independent variables in our estimated model is robust. Note that both criteria removed two promotion vehicles from the additive demand model.

[2] Following the suggestion of our industry collaborators, we decided to focus on the six most relevant stores for this project. These stores have a good data collection process and accurately recorded the historical use of promotion vehicles. In addition, our case study focuses on a specific item that does not sell the same way across all 18 stores. We are therefore constrained to focus on the stores that have a large volume of sales and a significant revenue for the particular item of interest.

References

Alimonti P, Kann V (2000) Some APX-completeness results for cubic graphs. *Theoret. Comput. Sci.* 237(1–2):123–134.

Arkin EM, Hassin R (1998) On local search for weighted k-set packing. *Math. Oper. Res.* 23(3):640–648.

Berman P (2000) A $d/2$ approximation for maximum weight independent set in d-claw free graphs. *Nordic J. Comput.* 7(3):178–184.

Berman P, Fujito T (1999) On approximation properties of the independent set problem for low degree graphs. *Theory Comput. Systems* 32(2):115–132.

Berstein Y, Onn S (2008) Nonlinear bipartite matching. *Discrete Optim.* 5(1):53–65.

Berstein Y, Lee J, Maruri-Aguilar H, Onn S, Riccomagno E, Weismantel R, Wynn H (2008) Nonlinear matroid optimization and experimental design. *SIAM J. Discrete Math.* 22(3):901–919.

Bertsimas D, Weismantel R (2005) *Optimization Over Integers*, Vol. 13 (Dynamic Ideas, Belmont, MA).

Blattberg RC, Neslin SA (1990) *Sales Promotion: Concepts, Methods and Strategies* (Prentice Hall, Englewood Cliffs, NJ).

Bomze IM, Budinich M, Pardalos PM, Pelillo M (1999) The maximum clique problem. Du D-Z, Pardalos PM, eds. *Handbook of Combinatorial Optimization* (Springer, New York), 1–74.

Campo K, Gijsbrechts E, Nisol P (2000) Towards understanding consumer response to stock-outs. *J. Retailing* 76(2):219–242.

Caro F, Gallien J (2010) Inventory management of a fast-fashion retail network. *Oper. Res.* 58(2):257–273.

Chandra B, Halldorsson MM (2001) Greedy local improvement and weighted set packing approximation. *J. Algorithms* 39(2):223–240.

Chandrasekaran R, Kabadi SN, Murthy KG (1982) Some NP-complete problems in linear programming. *Oper. Res. Lett.* 1(3):101–104.

Chen X, Simchi-Levi D (2004) Coordinating inventory control and pricing strategies with random demand and fixed ordering cost: The infinite horizon case. *Math. Oper. Res.* 29(3):698–723.

Cohen MC, Leung N-HZ, Panchamgam K, Perakis G, Smith A (2017) The impact of linear optimization on promotion planning. *Oper. Res.* 65(2):446–468.

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

Cormen TH, Leiserson CE, Rivest RL, Stein C (2009) *Introduction to Algorithms*, 3rd Ed. (MIT Press, Cambridge, MA).

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

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

Gutin G (2013) Independence and cliques. Gross JL, Yellen J, Zhang P, eds. *Handbook of Graph Theory*, 2nd Ed. Discrete Mathematics and Its Applications (CRC Press, Boca Raton, FL), 475–489.

Halldórsson MM, Yoshihara K (1995) Greedy approximations of independent sets in low degree graphs. *Proc. 6th Internat. Sympos. Algorithms Comput.*, 152–161.

Hassin R, Tamir A (1989) Maximizing classes of two-parameter objectives over matroids. *Math. Oper. Res.* 14(2):362–375.

Håstad J (1996) Clique is hard to approximate within $n^{1-\epsilon}$. *Proc. 37th Annual Sympos. Foundations Comput. Sci.*, 627–636.

Hazan E, Safra S, Schwartz O (2006) On the complexity of approximating k-set packing. *Computational Complexity* 15(1):20–39.

Heilman CM, Nakamoto K, Rao AG (2002) Pleasant surprises: Consumer response to unexpected in-store coupons. *J. Marketing Res.* 39(2):242–252.

Hemmecke R, Köppe M, Lee J, Weismantel R (2010) Nonlinear integer programming. Jünger M, Liebling ThM, Naddef D, Nemhauser GL, Pulleyblank WR, Reinelt G, Rinaldi G, Wolsey LA, eds. *50 Years of Integer Programming* 1958–2008 (Springer, Berlin), 561–618.

Hochbaum DS (2007) Complexity and algorithms for nonlinear optimization problems. *Ann. Oper. Res.* 153(1):257–296.

Kleinberg J, Tardos É (2005) *Algorithm Design* (Addison-Wesley, Boston).

Kök AG, Fisher ML, Vaidyanathan R (2008) Assortment planning: Review of literature and industry practice. Agrawal N, Smith S, eds. *Retail Supply Chain Management* (Springer, New York), 99–153.

Köppe M (2012) On the complexity of nonlinear mixed-integer optimization. Lee J, Leyffer S, eds. *Mixed Integer Nonlinear Programming* (Springer, New York), 533–557.

Lee J, Onn S, Weismantel R (2009) Approximate nonlinear optimization over weighted independence systems. *SIAM J. Discrete Math.* 23(4):1667–1681.

Macé S, Neslin SA (2004) The determinants of pre- and postpromotion dips in sales of frequently purchased goods. *J. Marketing Res.* 41(3):339–350.

Mulmuley K, Vazirani UV, Vazirani VV (1987) Matching is as easy as matrix inversion. *Combinatorica* 7(1):105–113.

Onn S (2003) Convex matroid optimization. *SIAM J. Discrete Math.* 17(2):249–253.

Papadimitriou CH (1984) Polytopes and complexity. Pulleyblank WR, ed. *Progress in Combinatorial Optimization* (Academic Press, Orlando, FL), 295–305.

Papadimitriou CH, Yannakakis M (1982) The complexity of restricted spanning tree problems. *J. ACM* 29(2):285–309.

Pardalos PM, Xue J (1994) The maximum clique problem. *J. Global Optim.* 4(3):301–328.

Phillips RL (2005) *Pricing and Revenue Optimization* (Stanford University Press, Stanford, CA).

Schrijver A (1998) *Theory of Linear and Integer Programming* (Wiley-Interscience, Chichester, UK).

Talluri KT, van Ryzin GJ (2004) *The Theory and Practice of Revenue Management* (Springer Science+Business Media, New York).

The Reinvestment Fund (2011) Understanding the grocery industry. https://www.reinvestment.com/wp-content/uploads/2015/12/Understanding_the_Grocery_Industry-Brief_2011.pdf.

Van Donselaar K, van Woensel T, Broekmeulen R, Fransoo J (2006) Inventory control of perishables in supermarkets. *Internat. J. Production Econom.* 104(2):462–472.

Van Heerde HJ, Leeflang PSH, Wittink DR (2000) The estimation of pre- and postpromotion dips with store-level scanner data. *J. Marketing Res.* 37:383–395.

Williamson DP, Shmoys DB (2010) *The Design of Approximation Algorithms* (Cambridge University Press, New York).

Wittink DR, Addona MJ, Hawkes WJ, Porter JC (1988) SCAN*PRO: The estimation, validation and use of promotional effects based on scanner data. Internal paper, Cornell University.

Wolsey LA, Nemhauser GL (1999) *Integer and Combinatorial Optimization* (Wiley-Interscience, New York).

Yi T, Murty KG, Spera C (2002) Matchings in colored bipartite networks. *Discrete Appl. Math.* 121(1–3):261–277.