Analytics in Promotional Pricing and Advertising

by

Lennart Baardman

B.Sc., University of Groningen (2013)
M.A.St., University of Cambridge (2014)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2019

Author . . . .
Sloan School of Management
May 17, 2019

Certified by . . . .
Georgia Perakis
William F. Pounds Professor of Management
Thesis Supervisor

Accepted by . . . .
Dimitris Bertsimas
Boeing Professor of Operations Research
Co-director, Operations Research Center

Analytics in Promotional Pricing and Advertising

by

Lennart Baardman

Submitted to the Sloan School of Management
on May 17, 2019, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

Big data and the internet are shifting the paradigm in promotional pricing and advertising. The amount of readily available data from both point-of-sale systems and web cookies has grown, enabling a shift from qualitative design to quantitative tools. In this work, we address how firms can utilize the power of analytics to maximize profits in both their offline and online channels.

First, we consider an online setting, in which an advertiser can target ads to the customer in question. The goal of the advertiser is to determine how to target the right audience with their ads. We study this problem as a Multi-Armed Bandit problem with periodic budgets, and develop an Optimistic-Robust Learning algorithm with bounded expected regret. Practically, simulations on synthetic and real-world ad data show that the algorithm reduces regret by at least 10-20% compared to benchmarks.

Second, we consider an offline setting, in which a retailer can boost profits through the use of promotion vehicles such as flyers and commercials. The goal of the retailer is to decide how to schedule the right promotion vehicles for their products. We model the problem as a non-linear bipartite matching-type problem, and develop provably-good algorithms: a greedy algorithm and an approximate integer program of polynomial size. From a practical perspective, we test our methods on actual data and show potential profit increases of 2-9%.

Third, we explore a supply chain setting, in which a supplier offers vendor funds to a retailer who promotionally prices the product to the customer. Vendor funds are trade deals in which a supplier offers a retailer a short-term discount on a specific product, encouraging the retailer to discount the product. We model the problem as a bilevel optimization model and show that a pass-through constrained vendor fund mitigates forward-buying and coordinates the supply chain on the short term.

Finally, we present a pilot study on the impact of promotional pricing on retail profits. We assess the potential impact of our promotion planning tool on historical data from a large retailer. Our results suggest a 9.94% profit improvement for the retailer.

Thesis Supervisor: Georgia Perakis
Title: William F. Pounds Professor of Management

Acknowledgements

First of all, I want to thank my advisor Georgia Perakis for her unwavering support. Georgia, to me, you personify what caring for another means. I will not know how my time as a PhD student would have been different without your mentorship, but I am sure that your guidance is one of the most important reasons for me being where I am. Your dedication is unrivaled, and has constantly pushed me forward. I want to thank you for all the research opportunities, teaching opportunities, and support in personal growth. My appreciation is incredibly hard to put in words. I am eternally grateful for all that you have done for me, and one day, I hope that I will be able to return the favor.

I want to thank Stephen Graves and Retsef Levi for being members of my thesis committee, their valuable advice on my thesis, and their help in pursuing an academic career. I would also like to thank Karen Zheng for your help in furthering my career. I am very grateful for the MIT ORC co-directors Dimitris Bertsimas and Patrick Jaillet whose vision and leadership have made the center an incredibly special place. In additions, I would like to express my gratefulness for everything that Laura Rose and Andrew Carvalho have done to help me, and for ensuring that the operations of the center run smoothly.

Special thanks go to Maarten van der Vlerk, Kees Jan Roodbergen, and Hector Carlo, as without your belief in my skills and without expressing those beliefs, I might not have pursued a PhD. Maarten, thank you for being a mentor in the first years of my studies. Kees Jan, thank you for being an amazing mentor, guiding me into research, and showing me the excitement in it. Hector, thank you for your advice to apply to premier PhD programs, as I might not have done so otherwise.

Without my brilliant collaborators, my research would have not been as productive. I would like to express my thanks to Danny Segev, Maxime Cohen, Divya Singhvi, Tamar Cohen-Hillel, and Elaheh Fata, who I have been fortunate in having as research collaborators during my time at MIT. Danny, thank you for the many research conversations, each of which taught me something new and has been of help ever since. Maxime, thank you for being such a great collaborator and someone who has supported me ever since. Divya, thank you for your amazing mentality, for being such a great person to work with, and for a lifelong friendship. Tamar, thank you for your incredible creativity, for taking the time to talk about anything, as well as a friendship that will never end. Elaheh, thank you for the extraordinarily productive time we have spent together. My thanks also go to Libby Aiello, Liam Fenlon, Yaakov Helman, Jeremy Kalas, Caroline Pech, and Xubo Sun, who I have had the privilege to work with while they were undergraduate students at MIT. I also extend my thanks to Setareh Borjian Boroujeni, Chen Dong, Igor Levin, Kiran Panchamgam, and Abhishek Pani, who have been important in developing my understanding of practical problems.

Personally, I have been privileged to meet wonderful people through the MIT ORC. Thank you to Max Biggs, Daniel Chen, Rim Hariss, Anna Papush, Elisabeth Paulson, Omar Skali Lami, Ioannis Spanditakis, and Charles Thraves for all your advice on presentations, support in teaching, and all the other wonderful moments. Additional thanks for the many conversations, advice, and trips go to Jonathan Amar, Ali Aouad, Martin Copenhaver, Adam Elmachtoub, Kimia Ghobadi, Nicolas Guenon des Mesnards, Swati Gupta, Kris Johnson Ferreira, Andrew Li, Shwetha Mariadassou, Velibor Misic, Nishanth Mundru, Yee Sian Ng, Kenza Sbai, John Silberholz, Raghav Singal, Somya Singhvi, Deeksha Sinha, Bradley Sturt, Joline Uichanco, and all the other people who I have had the privilege to meet with.

I want to thank my friends Michael Hu, Nick Koning, Jochem Kleine, Arjan Hylkema, Ilse Schouten, and Marianne Kuijpers. Michael, I am extremely grateful to be able to call myself your friend, and I want to thank you for all the time we have spent and will spend together. Nick, our uncountably many conversations on an endless variety of topics have always been a tremendous source of joy, yet it is difficult to express how grateful I am for our friendship and your support. Jochem and Arjan, thank you for spending countless hours of fun with me during the weekends. Ilse and Marianne, thank you for being there whenever I return home.

Finally, I would like to thank my family for being the foremost source of happiness in my life. Especially, I thank my grandparents for their immense contribution to my upbringing that has brought me here. Most of all, my thanks go to my parents and brother who have had an unmeasurable impact on my life. Without your unconditional love and support, I would not have been here today. Thank you, for everything.

Udo, Marianne, and Jesper, I dedicate this thesis to you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, the application of analytical techniques to problems in operations management and marketing has received increasing attention in academia. This development has been fueled by the parallel interest of practitioners in making use of the new opportunities brought by big data and the internet. Collaborating with practitioners, we observe that many firms are unable to leverage these opportunities, as they do not have the capabilities to implement advanced analytics. In this thesis, we develop new quantitative techniques to major problems in the interface between operations management and marketing.

In particular, we study the impact of using analytics to make better decisions in promotional pricing and advertising. We consider decision-making in both online and offline environments. For example, we consider an online advertising problem in which a firm needs to decide how to target the right customers with their ads. However, we also consider an offline advertising problem in which a retailer needs to determine how to schedule the right promotion vehicles such as flyers and commercial. In between, we study the promotional pricing problem from a supply chain perspective, specifically how a supplier can encourage their retail clients to promote their products to the customer. In each case, we develop new algorithms and test their potential impact on real advertising and retail data.

1.1 Motivation

Regardless of the advances in analytics by academia, the adoption of analytics in practice has not been widespread. The primary issue is that most advanced methods require knowledge of probability, statistics, machine learning, optimization, and the corresponding context in practice. Recognizing this gap, numerous large software firms, equipped with this knowledge, have started developing the software that is needed to bridge the gap. In its entirety, the results in this thesis are based on research conducted jointly with several of these large software firms in order to address this gap. The objective of this research is to address important problems in practice and to devise novel analytical tools that are key for practitioners.

This thesis focuses on problems in the interface of operations management and marketing, specifically promotional pricing and advertising. Until recently, few commercial tools were available in this area, while this area offers a lot of potential for the use of analytics. Predictive analytics can use the available data to fit demand forecasting models, while in turn, prescriptive analytics can utilize the demand models to solve promotion optimization models. The problems considered in this thesis are motivated by the industry need for predictive and prescriptive tools. Due to the novelty of these problems, they also provide academic opportunities to develop new analytical methods.

From the industry point of view, we show that these analytics models improve profits by as much as 2-20% for several large firms. In the industries that we focus on (online search ads and grocery stores), margins are relatively small, and a few percent additional profits can have a significant absolute impact. For example, online advertising revenues in the US in the first half of 2018 have grown to $49.5 billion, which is a 22.8% growth compared to the first half of 2017 (PricewaterhouseCoopers 2018). Moreover, grocery store revenues are even larger at $582 billion in 2014 and forecasted to be above $600 billion as of 2018 (Mazzone & Associates 2014). These numbers show the current value, while the growth numbers indicate the even greater potential once commercial tools will have been developed in the future.

From an academic perspective, we find many opportunities to both apply analytics and develop new methods. The existing and new methods that are needed vary substantially between problems. For example, in the online space, advertisers receive feedback on their decisions very quickly and in large amounts. Hence, they can rapidly change the prices and advertisements that they show. In turn, this allows the decision-maker to use online learning algorithms that adapt decisions in the short-term to the demand that is observed. On the other hand, in the brick-and-mortar store space, data is received more slowly and in lesser volume. Therefore, retailers need to plan their promotional pricing and advertising well in advance. In this case, decision-makers use statistics and machine learning to fit demand models which are then used in optimization models to find the right long-term schedule. Additionally, as we can gather personalized data on the internet, we can also personalize prices and advertisements to the customer in question. In contrast, even if we were able to collect individual data in retail stores, it is impossible to change the store layout, flyers, or commercials to the customer in question. These new capabilities and impossibilities create a host of new problems, some of which we develop data-driven tools for in this thesis.

1.2 Contributions

The main contribution of this thesis is the application of new analytics methods to new promotional pricing and advertising problems motivated by collaborations with practitioners. In this thesis, we consider three new fundamental problems in promotional pricing and advertising: the online advertising portfolio optimization problem in online advertising, the promotion vehicle scheduling problem in offline advertising, and the vendor fund problem in supply chain pricing.

From a practical perspective, we introduce and solve novel and different operational challenges that are currently faced by firms. There is limited published research that analyzes these problems from an analytical perspective. As all problems lie in the area of promotional pricing and advertising, their motivation is similar. In each problem, the goal is to either assign advertisements or prices in order to maximize revenue or profit subject to business rules such as budgets, capacity constraints, and promotional limits. On the other hand, there are substantial differences between advertising and pricing on the internet and in brick-and-mortar stores. For example, the level of personalization, the granularity of data, and frequency of feedback vary substantially.

From a methodological aspect, these practical differences create the opportunity to develop new methods. Different techniques from analytics allow us to build universally implementable models and algorithms to solve each problem. In practice, these algorithms need to be able to cope with large amounts of data, while still solving problems quickly. Regularly, these two interests contradict, as computing optimal solutions in a short time span is sometimes intractable. Therefore, we develop new, fast, and near-optimal approximation algorithms. The construction of these heuristics involves a broad mix of optimization, statistics, and machine learning methods.

In the case of online advertising, we devise algorithms that quickly adapt their decisions to the fast pace feedback. In particular, we propose the new Multi-Armed Bandit (MAB) problem with periodic budgets. For this problem, we construct a new learning algorithm that employs ideas from the existing Upper Confidence Bound (UCB) algorithms and robust optimization. In the case of offline advertising, we have to plan for a longer time horizon as feedback is received at a slower pace. Therefore, we propose to solve a mixed integer non-linear optimization problem. This problem is solved using either a greedy algorithm or approximate integer program. In the case of promotional pricing in supply chains, we need to solve and understand the relationship between the supplier's and the retailer's problem. For this reason, we use a bilevel optimization model solved with linear programming relaxations. The following provides more details on our contributions towards developing these algorithms.

First, we present the development of a new learning algorithm, with an analytical bound on its regret, to optimize the targeting of advertisements to customers in an online advertising setting. Online advertising enables advertisers to reach customers with personalized ads. As most online ads are priced using real-time auctions, advertisers need to bid on the right targets for their ads. As the outcomes of these auctions are random, a key complexity of the problem is that the revenue and cost of online ads are also uncertain. Collaborating with one of the largest ad-tech firms in the world, we develop new algorithms that help advertisers bid optimally on target portfolios while taking into account some limitations inherent to online advertising. We study this problem as a Multi-Armed Bandit (MAB) problem with periodic budgets. At the beginning of each time period, the advertiser needs to determine which portfolio of targets to select to maximize the expected total revenue (revenue from clicks/conversions), while maintaining the total cost of auction payments within the advertising budget. We formulate the problem and develop an Optimistic-Robust Learning (ORL) algorithm that uses ideas from Upper Confidence Bound (UCB) algorithms and robust optimization. We prove that the expected cumulative regret of the algorithm is bounded. Additionally, simulations on synthetic and real-world data show that the ORL algorithm reduces regret by at least 10-20% compared to benchmarks.

Second, we devise greedy and integer optimization algorithms, with theoretical guarantees as well as practical validation of their optimality, to schedule promotion vehicles in an offline advertising setting. 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 non-linear 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%.

Third, we study the promotional pricing problem from a supply chain perspective, specifically how a supplier can encourage their retail clients to promote their products to the customer. We analyze how pass-through constrained vendor funds impact promotion planning of both suppliers and retailers. Vendor funds are trade deals in which a supplier offers a retailer a short-term discount on a specific product, encouraging the retailer to discount the product. Prior proposals for vendor funds have had significant shortcomings, such as the need for sharing sales data between the retailer and the supplier. Instead, we propose the pass-through constrained vendor fund in which the supplier requires the retailer to pass-through a minimal fraction of the discount. The vendor fund offer and selection problem is modeled as a bilevel optimization problem in which a supplier wishes to determine what pass-through constrained vendor fund to offer to a retailer that can accept or reject the offer. First, we formulate the lower-level retailer model as an integer quadratic optimization model to help retailers decide on which vendor funds to accept. Using Lagrangian relaxation methods we create an efficient algorithm with theoretical guarantees and near-optimal performance on retail data. Second, we analyze a bilevel supplier model to determine which vendor fund a supplier should offer. We show that the vendor fund with pass-through constraint mitigates forward-buying by the retailer and coordinates supply chains on the short-term.

Finally, we demonstrate the potential impact of promotion planning on retail data. We present a retail application of data analytics and optimization for promotion planning. In recent years, specifically grocers, and retailers more generally, gained interest in improving their promotion planning system. Our promotion planning tool estimates demand models from data, and then solves a linear program to set the future promotions. We present the various challenges we faced in applying this data-driven solution approach. To assess the potential impact of our promotion tool, the models were tested on historical data from 2012-2014 in collaboration with a large retailer located in the Midwest of the United States. We first describe the data analysis we conducted, including the product selection and the store clustering method. Second, we present our results on demand estimation and promotion optimization. We believe that the results and methods presented are general, and can be applied to many retailers and product categories. Third, we present an empirical validation suggesting a 9.94% profit improvement for the retailer. This successful empirical validation convinced the retailer to set up a pilot experiment for a period of four months in 2015. We conclude the chapter by presenting our promotion recommendations, and discussing the potential broader impact to retailers.

The organization of the thesis is as follows. In Chapter 2, we propose learning algorithms to solve the online advertising portfolio optimization problem with periodic budgets. In Chapter 3, we solve the problem of scheduling promotion vehicles to boost profits. Chapter 4 considers the problem of creating pass-through constrained vendor funds that aid promotional pricing in supply chains. Chapter 5 presents the impact that data analytics can have on promotion planning. Finally, Chapter 6 concludes and summarizes this thesis. The chapters in this thesis are self-contained, and hence, can be read in any order.

Chapter 2

Learning Optimal Online Advertising Portfolios with Periodic Budgets

2.1 Introduction

Due to the tremendous growth in personalized data and online channels, advertisers have gained the ability to create ad campaigns that target advertisements to specific customer segments through a variety of online channels. Each such combination of ad, segment, and channel forms a different target. An example of a target can be showing a particular website link to users searching for a particular keyword on Google. In general, publishers of online advertisements (e.g., Google, Facebook) have a limited supply of advertising slots on their platform when compared to the large demand from advertisers for showing advertisements. For this reason, many publishers run real-time auctions to determine which advertiser gets to show their ad to the platform's visitors. The market associated with these online advertising auctions is large and growing rapidly. According to PricewaterhouseCoopers (2017), in the United States alone, online advertising revenues were on the order of \$40.1 billion in the first half of 2017 and had increased by 22.6% when compared to the first half of 2016. Thus, it is important for the publishers to create the right auction platform and for the advertiser to place the right bids in these auctions. Previous studies have largely focused on the auction and ad allocation problem of the publishers, while only limited work has analyzed the bidding problem from the perspective of advertisers.

Through a collaboration with an online advertising intermediary, we focus on the online advertising portfolio optimization problem that advertisers face in practice. We observe that publisher's platforms require online advertisers to maintain a portfolio of targets that they are interested in, and to update the bids and the budget for the portfolio on a periodic basis (often hourly or daily). At the beginning of a time period, the platform allows the advertiser to update their target portfolio based on historical data. During the time period, whenever a user visits, the platform will place a bid if the advertiser selected the corresponding target. At the end of a time period, the platform gives the advertiser feedback on the target portfolio's performance in the form of revenue and cost data. The reason that we need to state our interests periodically, and not individually for every auction, is that collecting and sharing this data at the large scale of online advertising generates inherent time delays (Google Ads 2019a). The periodic advertising revenue of a target comes from clicks or conversions, i.e., revenues are gained when customers click on ads and possibly generate purchases, or any other intermediate outcome that the marketer may care about. At the same time, the periodic advertising cost of a target is dictated by the payments that need to be made in the auctions during the period, i.e., costs are incurred when auctions are won and possibly result in customer clicks.

The goal of the advertiser is to select the portfolio of targets (at the beginning of the period) that will result in the largest advertising revenue (by the end of the period). Many platforms expect the advertiser to specify daily or hourly budgets (Google Ads 2019b). Simultaneously, for strategic purposes, most advertisers only want to spend a limited advertising budget during the period. Outspending this periodic budget leads to lost revenues as the advertiser is unable to participate in any further ad auctions in a period once the period's advertising costs exceed its budget. We capture these two constraints through a budget constraint that the advertiser needs to specify at the beginning of each period. This is different from the existing models in the literature, as those address the problem where, in each period of the time horizon, the advertiser selects the bid for a single target and the single budget constraint is slowly depleted over the entire time horizon. In this work, we study the case where, in each period of the time horizon, the advertiser selects a portfolio of targets to bid on and within that time period the budget constraint needs to be met.

Before a time period, the advertising revenues and costs of that new period are unknown to the advertiser, due to their dependence on random factors such as the conversion behavior of customers and the bidding of competing advertisers. In addition, the large number of targets (possibly millions) leads to limited and sparse data per ad, which makes estimating the expected revenue and cost of each target difficult. Hence, to ensure that we bid optimally on the available targets, we need to collect sufficient data and learn which targets yield a high return on investment (ROI). Working together with our online advertising partner, we observe that the difficulty of estimating revenue and costs is not addressed by the current online advertising portfolio optimization software and literature. Current methods periodically solve an optimization model that incorporates deterministic estimates of the expected revenue and cost, which are assumed to be accurate. However, in reality, the estimates may be far from the target's true expected revenue and cost, especially when the number of observed revenues, costs and ad related events are sparse. In the long run, this may cause the repeated selection of less profitable targets leading to lost profits. In this work, our goal is to construct a policy that is able to learn every target's periodic advertising revenue and cost, yet not sacrifice too much in terms of revenue. As a consequence, we will formulate a complex problem where revenues and costs of targets need to be learned (exploring value), while bidding as often as possible on the most efficient targets (exploiting value). To address this problem, we will cast the online advertising portfolio optimization problem into an exploration-exploitation framework (particularly, a Multi-Armed Bandit problem).

2.1.1 Contributions

The main contribution of this chapter is the introduction of the exploration-exploitation framework to the online advertising portfolio optimization problem. This is accompanied by the development of the Optimistic-Robust Learning (ORL) algorithm, that is a new type of algorithm for the Multi-Armed Bandit (MAB) problem. In our analysis, we show strong results on the expected regret of our proposed algorithm, both in terms of a theoretical guarantee as well as computational results. In more detail, our contributions are the following:

1. *Online advertising portfolio optimization as a Multi-Armed Bandit with periodic budgets.* In comparison to most online advertising literature that focuses on the publisher's perspective, we focus on the advertiser's problem of bidding for the right targets in online ad auctions over a finite time horizon. In reality, advertisers split their advertising campaigns (often last for weeks or months) into time periods (often last hours or days). In each time period, the advertiser's goal is to bid on a portfolio of targets that maximizes advertising revenues under the constraint that advertising costs should not exceed the period's ad-

vertising budget. In the stochastic MAB context, in each time period, the agent wishes to pull a set of arms that maximizes revenue while keeping the cost of pulling those arms below the period's budget. This problem differs in two ways from the traditional advertiser's problem and the standard MAB problem. Firstly, the decision-making process features both the offline optimization of the target porfolio and the online learning of each target's revenue and cost estimates. Previous literature on the advertiser's problem does not account for this online component. Secondly, the portfolio selection entails pulling multiple arms in a time period bounded by a budget constraint, in which the cost parameters are uncertain. Existing literature on the MAB problem does not study this pulling of an uncertain set of arms. Section 2.2 describes the online advertising process in more details, and Section 2.3 introduces the exploration-exploitation framework to the online advertising portfolio optimization problem and formulates it as an MAB with periodic budgets.

2. *Optimistic-Robust Learning algorithm incorporating upper confidence bound and robust optimization techniques.* We show that the MAB problem with periodic budgets is difficult to solve, even for an oracle that knows the expected revenue and cost of each arm. Thus, we use robust optimization techniques to create an oracle policy that is similar to the optimal policy. Robust optimization methods help to avoid the case where the cost of pulling arms exceeds the budget. As the expected revenues and costs of targets are unknown, this oracle policy is not implementable. Hence, we devise the ORL algorithm that is based on Upper Confidence Bound (UCB) algorithms, which are well known in the MAB literature, and additionally use robust optimization to protect against exceeding the budget. This algorithm is able to explore the possible arms (learn the expected revenue and cost), and exploit the valuable arms (generate revenue while staying within budget). Section 2.4 describes the ORL algorithm.

3. *Bound on the expected regret of the Optimistic-Robust Learning algorithm.* We show how the parameters of the ORL algorithm can be tuned to achieve a bound on expected regret. We observe that the regret is bounded by a function that is polylogarithmic in the number of time periods and polynomial in the number of arms. The problem that we consider can also be written as a standard MAB problem in which the arms represent collections of targets, but here, the bound on expected regret for usual MAB algorithms is exponential

in the number of arms. Section 2.5 presents and proves the bound on expected regret.

4. *Strong computational performance of the Optimistic-Robust Learning algorithm against benchmarks.* To benchmark the ORL algorithm, we devise a Sample Average Approximation (SAA) type of algorithm that does not explore unknown targets or account for budget violations. This approach passively updates the revenue and cost estimates over time. Through simulations under various settings, we observe that the ORL algorithm significantly reduces regret by more than 20% compared to a passive learning approach. Moreover, we develop a UCB type of algorithm that explores unknown targets but does not account for budget violation. This approach actively tries unseen targets to learn about every target's revenue and cost estimate. In the same simulations, we observe that the regret of the ORL algorithm is at least 10% smaller that of the active learning algorithm. Section 2.6.1 introduces the benchmarks, and Section 2.6.2 discusses the results from simulations with synthetic data.

5. *Significant regret reduction in simulations based on real-world data.* Working together with a large online advertising intermediary, we test the algorithm on instances based on the data from advertising portfolios of our industry partner's clients. In these simulations we observe an improvement of 10-15% over the SAA-based and UCB-based algorithms. Section 2.6.3 introduces the client data and tests the strength of the ORL algorithm on it.

2.1.2 Literature Review

In this work, we study the online advertising portfolio optimization problem, which connects to a number of different research areas. On the practical side, this problem is related to the literature on online advertising. From the theoretical perspective, it is related to the Multi-Armed Bandit (MAB) problems. Regarding methodology, it is related to the literature on Upper Confidence Bound (UCB) algorithms. These topics cover areas in operations research, operations management, revenue management, marketing, and machine learning.

From a practical point of view, online advertising has recently attracted attention from researchers in the field of pricing and revenue management. Most of this literature has focused on the publisher's problem. Publishers use a variety of methods to sell their advertisement space, ranging from real-time bidding where advertisers bid on advertisement slots to ad contracts where advertisers purchase a fixed number of advertisement slots at a pre-specified price. Using an MAB framework, Cohen et al. (2018b) consider a dynamic pricing model that allows publishers to optimally price ad impressions (generated by visiting users) to advertisers. Candogan and Pekeč (2018) devise an algorithm that creates market-clearing prices for publishers that allocate ad slots to advertisers whose valuations for targets might differ depending on the bundle of targets that is won in the auction. In this context, where many publishers use second-price auctions, Golrezaei et al. (2019) develop a boosted second-price auction that improves revenues when the bidding advertisers are heterogeneous in their valuation for ad slots.

The literature on the advertiser's side of the problem is limited. Rusmevichientong and Williamson (2006) analyze the advertiser's problem of selecting the right set of targets when advertising on a search platforms over time. This problem is modeled as a stochastic knapsack problem and the authors develop an adaptive algorithm able to solve this problem. In contrast to our model, this paper assumes that the cost of bidding on a keyword is deterministic. Borgs et al. (2007) consider the problem from a game-theoretic perspective, where multiple bidders with limited budgets participate in auctions for multiple targets and multiple ad slots. More recently, in Balseiro and Gur (2018), the problem of selecting how much to bid on a single target is viewed as a sequential game with incomplete information, as bidders do not know their valuation distribution or their competitor's valuation distribution. This paper shows that an adaptive pacing strategy attains asymptotic optimality when competitor's bids are independent and identically distributed. Pani et al. (2017) consider the online advertising portfolio optimization problem where an advertiser can solve the optimal bid determination problem at very large scale, given a set of targets and budget. Contrasting our model that is based on exploration-exploitation, this paper assumes that the revenue and cost of each target are known, and hence, the paper focuses on finding a fast algorithm to solve this deterministic problem.

On the theory side, the literature on the MAB problem is extensive. Originally proposed by Robbins (1952), Lai and Robbins (1985), and Auer et al. (2002), the standard stochastic MAB problem is an example of the trade-off between exploration and exploitation. In the original setting an agent has n arms available, each delivering revenues from unknown distributions once pulled. The agent's goal is to learn the revenue of each arm and pull the best one in each time period to maximize its expected total revenue. Fundamentally, the MAB problem is characterized by the trade-off between pulling the best arm based on the gathered information (exploitation) and pulling other arms to gather more information (exploration). Many different types of algorithms have been constructed to balance this trade-off and generate good solutions to this problem. In particular, our algorithm relates to the type of UCB algorithms (Auer 2002). However, our algorithm also uses ideas from robust optimization (Bertsimas et al. 2011) as there are significant differences between the standard MAB and the MAB with periodic budgets.

The original MAB setting assumes that pulling arms is free of cost, but this does not describe the real world sequential decision-making scenarios where pulling an arm can be costly. Some examples of such scenarios are the bid optimization problem in online ad auctions (Borgs et al. 2007) and the real-time bidding problem in ad exchanges (Chakraborty et al. 2010). To address this, Tran-Thanh et al. (2010) introduce the budget-limited MAB. In this problem, every time period, the agent pulls an arm that generates a random revenue and incurs a fixed cost for the agent. The agent has a budget that is diminished every time by the incurred cost, and once the budget is depleted the agent stops pulling arms. This budget constraint is a 'global' constraint that holds over all the time periods that the algorithm runs. Tran-Thanh et al. (2010) discuss that the standard MAB model is not able to describe this problem and propose an algorithm for this problem that achieves a sublinear regret as a function of the budget. Tran-Thanh et al. (2012) devise a new algorithm for this problem that has a logarithmic regret in the budget.

Subsequently, Amin et al. (2012), Ding et al. (2013), Tran-Thanh et al. (2014), Xia et al. (2016) use UCB algorithms to study this problem of maximizing revenue subject to a budget constraint in the setting where both revenues and costs are randomly drawn from unknown distributions. In particular, Badanidiyuru et al. (2018) extends to the problem of multiple 'global' constraints. In this case, whenever an arm is pulled, multiple resources are consumed. Each of these resources has their own budget constraint, and once one of the resources depletes, the algorithm has to stop pulling arms. In contrast to the UCB algorithm, Ferreira et al. (2018) propose a Thompson sampling algorithm to solve this problem. In Agrawal and Devanur (2014), this problem is extended with a concave objective and 'global' convex constraints. The main difference between this literature and our work is that the previous literature considers an agent that pulls one arm in a time period, whereas we assume that at the beginning of each period the agent needs to select which set of arms to pull. In previous works, the budget constraint is 'global', meaning the algorithm stops when the cost incurred over all time periods exceeds the budget. In our work, the budget constraint is 'local', meaning the cost incurred during each time period should stay below the period's budget.

To the best of our knowledge, in all previous work on the budgeted MAB problem the budget constraint applies to the cost of all arm pulls over the time horizon (i.e., bounding the advertising cost of an entire campaign). However, there are applications such as ad campaigns (often running for weeks or months) for which the time horizon is divided into time periods (often lasting a day or several hours) where each period has its own allocated budget (Pani et al. 2017). In addition, the advertiser is not able to decide which targets to bid on one by one, but has to select a portfolio of targets at the beginning of the period. The reason for this is that ad platforms only send revenue and cost feedback in a delayed manner at the end of a period (Google Ads 2019a). In such scenarios, at the beginning of each time period, a set of arms is chosen that maximizes the expected total revenue while respecting the period's budget constraint. This work aims to address the gap between the case where there exists a single budget for the entire time horizon and the case that each time period has its own budget limit.

From a methodological point of view, this paper casts the online advertising portfolio optimization problem in an exploration-exploitation framework. In doing this, we apply the UCB algorithm (Auer 2002, Auer et al. 2002), which is a natural algorithm for MAB problems, to help both learn the value of targets and optimize the portfolio of targets. The study of UCB-based algorithms has extended well beyond the traditional MAB problem, such as linear contextual MAB problems (Dani et al. 2008, Rusmevichientong and Tsitsiklis 2010). An additional challenge in our MAB with periodic budgets is that during the arm selection process the to-be realized costs are unknown. Hence, the selected set of arms might in fact be infeasible to pull. We use techniques from the robust optimization literature to bound the probability that the realized cost of the chosen arms violates the budget constraint. Similar probabilistic guarantees on the feasibility of a solution to a robust optimization problem can be found in Bertsimas et al. (2004). The interested reader is referred to Bertsimas and Sim (2004), Bertsimas et al. (2011), Gabrel et al. (2014) for an overview on robust optimization.

2.2 Online Advertising Process

In this section, we describe the process by which online advertising platforms function. The process is visualized in Figures 2.1 and 2.2 in two steps: Figure 2.1 describes the problem of selecting a portfolio of targets given estimates of the revenue and cost (offline decision-making) and Figure 2.2 describes how the new revenue and cost data is used to update their estimates (online learning).

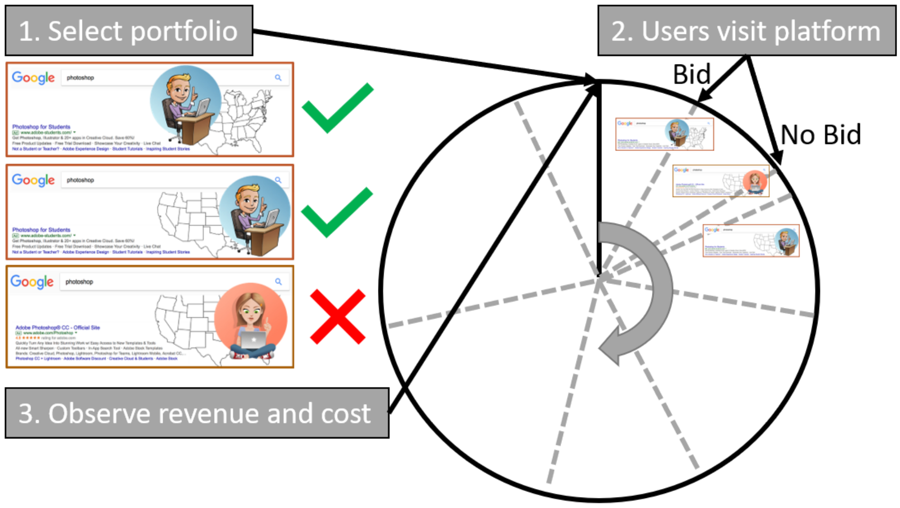


Figure 2.1: Process of online advertising portfolio optimization on an advertising platform during one time period: a portfolio of targets is selected at the beginning of the period, bids are placed on selected targets if they appear during the period, and the resulting revenue and cost are observed at the end of the period.

In Figure 2.1, the clock represents a period, which is often an hour or a day long. At the beginning of the period, the advertiser decides on a portfolio of targets to bid on and communicates these to the platform. During the period, users visit the platform and if a visiting user is among the targets in the portfolio, the platform will activate a bid for the advertiser. Inherently, the revenue and cost from each user are random to the advertiser, and hence, the period's revenue and cost for each target are random as well. The period's revenue of a target depends on the number of visiting users, whether the auctions are won, whether the customers click the ad, and whether the customers purchase the item. The period's cost of a target depends on the number of visiting users, whether the auctions are won, and whether the customers click the ad. At the end of the period, the platform gives the advertiser feedback on the total revenue and cost that each target produced during the period.

The nature of this process implies that, at the beginning of each period, the advertiser's objective is to select a portfolio of targets that maximizes advertising revenue, while keeping the advertising cost within the period's budget. Staying within the budget is important to avoid lost revenues. The advertiser cannot participate in any further auctions once the budget is depleted, which could lead to lost revenues. Once a target portfolio is selected at the beginning of a time period, it will not be modified for that specific time period, for two reasons. Firstly, the revenue and cost data are received in a delayed manner (Google Ads 2019a). If no new information is received during the period, there is no reason to update the portfolio. Secondly, the rate with which users visit the platform is very high. If the advertiser would update the portfolio after every user visit, then the frequency of updates would be very high as well. Physical constraints, such as computational power and the network connection speed and bandwidth, limit the rate of portfolio modification. Hence, it is impossible for the advertiser to update the target portfolio after every user visit. Nevertheless, from one period to another, we can use the new revenue and cost information to modify our target portfolio. This illustrates the offline decision-making level, where a target portfolio is chosen and not changed during an entire time period, and online learning level, where the data obtained during previous periods can be used to make better decisions for the next period.

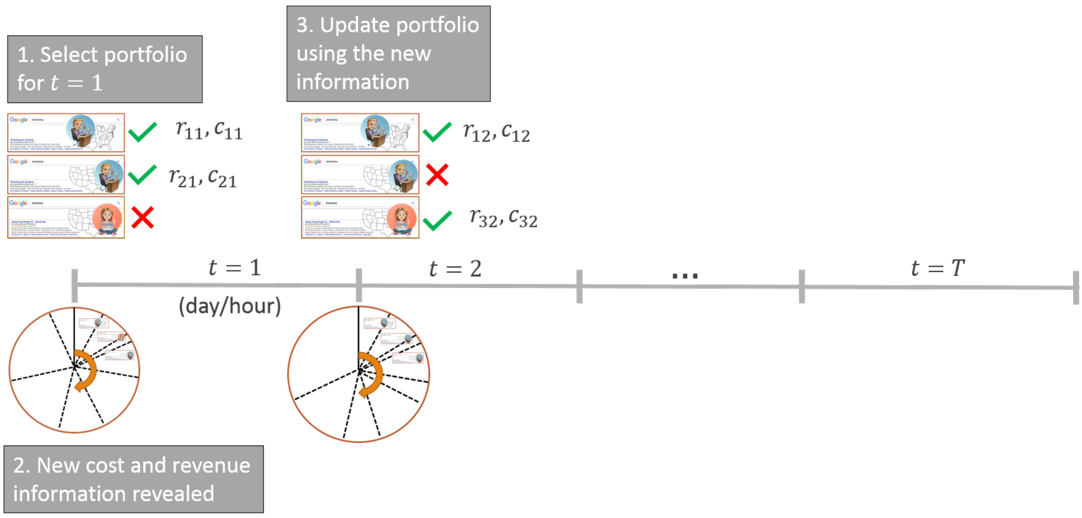


Figure 2.2: Process of online advertising portfolio optimization for an advertising campaign of T time periods: a target portfolio is selected for the first time period and feedback on their revenue and cost is received, this information can be used to update the portfolio of targets for the second time period and so on.

In Figure 2.2, the timeline depicts the entire ad campaign divided into T time periods. With the new information provided at the end of a time period, the question is how to update the portfolio for the next time period accordingly. After several time periods, some data is available to select a target portfolio that provides good short-term profits. However, this target portfolio might not yield the best long-term profits, as more data might have given a better indication of the most profitable portfolio. Therefore, this problem lends itself to an exploration-exploitation framework, where we want to learn the expected revenue and cost of each target while maximizing the expected revenue, or equivalently, minimizing the expected regret. Naturally, we formulate this problem as an altered Multi-Armed Bandit (MAB) problem. In the MAB context, in each period, an agent wants to pull a set of arms that yield maximal revenue while their cost is below the period's budget. Here, the arms represent the available targets, and a period represents the interval between the time at which a target portfolio is given (by the advertiser to the platform) and the time at which feedback on that target portfolio is received (by the advertiser from the platform).

2.3 Multi-Armed Bandit Problem with Periodic Budgets

In this section, we formulate the online advertising portfolio optimization problem as the MAB problem with periodic budgets. First, we describe how the budget affects the expected revenue. Then, we determine how an oracle policy that does not need to learn expected revenues and costs would maximize the expected revenue. Finally, we define the MAB problem with periodic budget as finding a learning policy that minimizes the expected regret.

2.3.1 Expected Revenue Maximization

Consider the case in which there exists a total of n arms (targets), indexed by i. Over a time horizon of T time periods, indexed by t, we are interested in maximizing expected revenue by selecting a set of arms to pull while being constrained by a budget B in each period. Note that the model and results presented in this section as well as the rest of the chapter can be generalized to the case that each period has a different budget value. However, for the ease of presentation we assume, without loss of generality, that all periods have the same budget value. Let the binary decision variable x_{it}^{Π} indicate whether the learning algorithm Π pulls arm i in period t. Additionally, for each arm there is an associated revenue and cost. In particular, let r_{it} and c_{it} be the random variables representing the to-be-realized revenue and cost if arm i is pulled in period t. For ease of notation, we assume that these observations are independently drawn from distributions supported on $[0, 1]$ with stationary expected revenue μ_i^r and cost μ_i^c, yet any bounded distributions suffice. We assume that in each period we receive feedback about the realized revenues and costs of the selected arms.

In each period, we are constrained by a budget B that is assumed to be at least 1. This assumption guarantees that at least one of the arms can be pulled in each period. The implications of this budget are that in a period where the cost is within the budget, we will obtain the usual revenue. However, if the cost in a period exceeds the budget, we will incur a revenue penalty. In particular, we let $L(\sum_{i=1}^{n} c_{it}x_{it}, B)$ be the loss function describing which fraction of the portfolio's realized revenue, $\sum_{i=1}^{n} r_{it}x_{it}$, will be obtained if the portfolio's realized cost is

$\sum_{i=1}^{n} c_{it}x_{it}$ and the period's budget is B. Altogether, this means that the realized total revenue equals

$$R_T = \sum_{t=1}^{T}\sum_{i=1}^{n} r_{it}x_{it} \cdot L\left(\sum_{i=1}^{n} c_{it}x_{it}, B\right). \tag{2.1}$$

We remark that our analysis applies to any general loss function. Nevertheless, in what follows we consider the following two specific loss functions: the fractional loss function and the indicator loss function. The fractional loss function resembles reality and is given by $L\left(\sum_{i=1}^{n} c_{it}x_{it}, B\right) = \min\{1, \frac{B}{\sum_{i=1}^{n} c_{it}x_{it}}\}$. If the portfolio's total realized cost is within budget, then the portfolio's revenue is fully obtained. But, if the portfolio's total realized cost exceeds the budget, then its revenue is multiplied by a factor describing roughly how long during the period we were able to bid. The indicator loss function is given by $L\left(\sum_{i=1}^{n} c_{it}x_{it}, B\right) = \mathbb{I}\left(\sum_{i=1}^{n} c_{it}x_{it} \leq B\right)$, where the full revenue is obtained if the realized cost is within the budget and no revenue is obtained if the realized cost exceeds the budget. In general, it is realistic to assume that any loss function equals 1 if the cost is within budget, and is a non-increasing function of the portfolio's total realized cost otherwise. In this work, we assume an indicator loss function, i.e., no revenue is obtained if the total realized cost exceeds the budget. We remark that our analysis in this work carries over to any general loss function. Thus, we can write the realized total revenue as

$$R_T = \sum_{t=1}^{T}\sum_{i=1}^{n} r_{it}x_{it}\mathbb{I}\left(\sum_{i=1}^{n} c_{it}x_{it} \leq B\right), \tag{2.2}$$

where $\mathbb{I}(\cdot)$ is the indicator function. In reality, the realized revenue and cost in (2.1) are not known beforehand. Therefore, we select arms to maximize the expected total revenue,

$$\mathbb{E}[R_T] = \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{n} r_{it}x_{it}\mathbb{I}\left(\sum_{i=1}^{n} c_{it}x_{it} \leq B\right)\right] = \sum_{t=1}^{T}\sum_{i=1}^{n} \mu_i^r x_{it}\mathbb{P}\left(\sum_{i=1}^{n} c_{it}x_{it} \leq B\right). \tag{2.3}$$

For every time period t, we want to maximize the inner expression of (2.3), denoted by (P):

$$\max_{x_t \in \{0,1\}^n} \sum_{i=1}^{n} \mu_i^r x_{it}\mathbb{P}\left(\sum_{i=1}^{n} c_{it}x_{it} \leq B\right). \tag{2.4}$$

Our goal is to construct a learning algorithm Π that does not know the expected revenues and costs (μ_i^r, μ_i^c) nor realized revenues and costs (r_{it}, c_{it}), but yields revenues that are close to the optimal revenue of (P). To analyze the closeness in revenue, we will use the notion of expected regret: if we knew the expected revenues and costs, how much more revenue could have been gained over time?

2.3.2 Oracle Policy

In order to formulate the expected regret, we need to understand how an oracle policy that knows expected revenues and costs would function. From (2.4), we observe that directly maximizing the expected revenue in each period is difficult. More precisely, the second term, the probability that the cost is within the budget, is hard to optimize over, as it is hard to compute. Computing this probability requires convolutions of non-identically distributed random variables, which is NP-hard (Möhring 2001). Thus, even though a natural idea for the oracle would be to solve (P), we will focus our attention on a more tractable version. For this purpose, let us lower bound the probability of staying within budget by $1-\alpha$. Then, we can solve the following problem to find the optimal arms to pull given the required probability of feasibility, we call this problem (P_α):

$$\max_{x_t \in \{0,1\}^n} \sum_{i=1}^{n} \mu_i^r x_{it} \tag{2.5a}$$

$$\text{s.t. } \mathbb{P}\left(\sum_{i=1}^{n} c_{it} x_{it} \leq B\right) \geq 1-\alpha \tag{2.5b}$$

Now, the optimal solution to (P) can be obtained by solving (P_α) for all $\alpha \in [0,1]$, and then selecting the solution corresponding to the α that maximizes $\sum_{i=1}^{n} \mu_i^r x_{it}(1-\alpha)$. We denote the optimal α by α^*, and note that $(P) = (P_{\alpha^*})$. Yet, for the same reason as before, (P_α) is still intractable for a fixed α. Nonetheless, in the rest of this chapter, we propose the following analyzable oracle. The oracle policy applies ideas from robust optimization to relax (P_α) into a tractable problem (RP_Λ):

$$\max_{x_t \in \{0,1\}^n} \sum_{i=1}^{n} \mu_i^r x_{it} \tag{2.6a}$$

$$\text{s.t. } \sum_{i=1}^{n} (\mu_i^c + \Lambda)\, x_{it} \leq B \tag{2.6b}$$

where we select Λ such that $\mathbb{P}\left(\sum_{i=1}^{n} c_{it} x_{it} \leq B\right) \geq 1-\alpha$. This relationship between Λ and α is specified in Proposition 2.3.1. By selecting the smallest Λ that still guarantees $\mathbb{P}\left(\sum_{i=1}^{n} c_{it} x_{it} \leq B\right) \geq 1-\alpha$, we ensure that the optimal solution of (RP_Λ) equals the optimal solution of (P_α). Hence, by setting Λ in such a way, we are able to extrapolate our expected regret bound from the case where the oracle solves (RP_Λ) to the case where the oracle solves (P_α). In particular, Proposition 2.3.1 describes the smallest Λ such that the probability of exceeding the budget is at most α.

Proposition 2.3.1. If $\Lambda \geq \sqrt{\ln\left(\frac{1}{\sqrt{\alpha}}\right)}$, then for any x_t feasible for (RP_Λ),

$$\mathbb{P}\left(\sum_{i=1}^{n} c_{it}x_{it} \leq B\right) \geq 1 - \alpha.$$

Proof. By definition of the solution to (RP_Λ) we know that $\sum_{i=1}^{n}(\mu_i^c + \Lambda)x_{it} \leq B$, and hence,

$$\mathbb{P}\left(\sum_{i=1}^{n} c_{it}x_{it} > B\right) \leq \mathbb{P}\left(\sum_{i=1}^{n} c_{it}x_{it} > \sum_{i=1}^{n}(\mu_i^c + \Lambda)x_{it}\right) = \mathbb{P}\left(\sum_{i=1}^{n}(c_{it} - \mu_i^c)x_{it} > \sum_{i=1}^{n}\Lambda x_{it}\right)$$
$$\leq e^{-\frac{2(\sum_{i=1}^{n}\Lambda x_{it})^2}{\sum_{i=1}^{n}x_{it}}} \leq e^{-\frac{2\sum_{i=1}^{n}\Lambda^2 x_{it}}{\sum_{i=1}^{n}x_{it}}} \leq e^{-\frac{2\sum_{i=1}^{n}\ln(1/\sqrt{\alpha})x_{it}}{\sum_{i=1}^{n}x_{it}}} = \alpha,$$

where the second inequality follows from Hoeffding's inequality. □

To formulate our notion of regret, we need to understand the structure of the optimal solution to (RP_Λ). We observe that (RP_Λ) is a robust version of the binary knapsack problem, which is an NP-hard problem that cannot be solved to optimality in polynomial time, unless P=NP. Nonetheless, by making an assumption on the parameters of (RP_Λ), the oracle policy is to apply the greedy algorithm that finds an optimal or near-optimal solution to (RP_Λ). The greedy algorithm picks the arms with the largest reward-to-cost ratios. If we order the reward-to-cost ratios in decreasing order (i.e., $\frac{\mu_i^r}{\mu_i^c+\Lambda} > \frac{\mu_{i+1}^r}{\mu_{i+1}^c+\Lambda}$), then the first h arms form the greedy solution in which $x_{it} = 1$ for all $1 \leq i \leq h$ and $x_{jt} = 0$ for all $h < j \leq n$.

The optimality gap of this solution depends on how much of the budget is leftover. For example, in the case where the expected cost of the first h arms exactly depletes the budget (i.e., $\sum_{i=1}^{h}(\mu_i^c + \Lambda) = B$), the greedy solution is optimal. On the other hand, when a small part of the budget is leftover after subtracting the expected cost of the first h arms (i.e., there exists a $\delta > 0$ such that $\sum_{i=1}^{h}(\mu_i^c + \Lambda) = B - \delta$), then the greedy solution is near-optimal. In the context of online advertising, the assumption that a portfolio's expected cost fully utilizes the budget B or only leaves a small part of the budget $\delta > 0$ leftover is justified. In the case of online advertising, we mentioned that the number of arms is large (possibly millions) and the expected costs of each arm are rather close and small compared to the budget. Generally, this means that the leftover budget should be negligible compared to the entire revenue, leading to a negligible optimality gap.

In fact, the following proposition shows that the optimality gap shrinks as the number of arms n increases, as long as the periodic budget B is not constant and grows with the number of arms n. In practice, the budget normally increases as more arms are introduced, because it creates more opportunities for additional revenues. Even though the budget might not increase very steeply as the number of arms grows, this result holds regardless of the rate of growth. Thus, the result holds even if the budget grows slowly, for example, $\log^*(n)$ (that is the number of times to iteratively apply the logarithm function on n before the result is less than one).

Proposition 2.3.2. If B increases to infinity as n increases, then the greedy solution converges to the optimal solution of (RP_Λ) as n increases.

Proof. Suppose the greedy solution to (RP_Λ) takes the first h items ordered decreasingly based on their $\frac{\mu_i^r}{\mu_i^c+\Lambda}$ ratio. Define $\delta = B - \sum_{i=1}^{h}(\mu_i^c + \Lambda)$. If $\delta = 0$, then in fact there is no optimality gap and the greedy algorithm's solution to RP_Λ is optimal, and thus, the regret bound in Theorem 2.5.1 holds for the case that the oracle is stronger and can solve RP_Λ to optimality.

Hence, let us focus on the case that $\delta > 0$. We have that $\delta < \mu_{h+1}^c + \Lambda$, otherwise the greedy algorithm could pick the $(h+1)$-th item. Let $rev(\text{greedy}(RP_\Lambda))$ and $rev(\text{OPT}(RP_\Lambda))$ denote the total revenue gained by the greedy and optimal algorithms on (RP_Λ). As discussed in Section 2.3, $rev(\text{greedy}(RP_\Lambda)) = \sum_{i=1}^{h}\mu_i^r$. Note that

$$rev(\text{greedy}(RP_\Lambda)) \geq \frac{\mu_{h+1}^r}{\mu_{h+1}^c + \Lambda}(B - \delta) \tag{2.7}$$

as the greedy algorithm utilized only $B - \delta$ of the budget and picked items with better revenue-to-cost ratio than item $h+1$. Furthermore, observe that

$$rev(\text{greedy}(RP_\Lambda)) \leq rev(\text{OPT}(RP_\Lambda)) \leq rev(\text{greedy}(RP_\Lambda)) + \delta\frac{\mu_{h+1}^r}{\mu_{h+1}^c + \Lambda} \tag{2.8}$$

since the right-hand side of (2.8) is the solution to the LP relaxation of (RP_Λ), as it fully consumes the budget with the best set of items, hence it is an upper bound for $rev(\text{OPT}(RP_\Lambda))$.

From (2.8) we can derive the following:

$$\begin{aligned}
rev(\text{OPT}(RP_\Lambda)) &\leq rev(\text{greedy}(RP_\Lambda))(1+\frac{\delta}{\mu^c_{h+1}+\Lambda}\times\frac{\mu^r_{h+1}}{rev(\text{greedy}(RP_\Lambda))}) \\
&\leq rev(\text{greedy}(RP_\Lambda))(1+\frac{\delta}{\mu^c_{h+1}+\Lambda}\times\frac{\mu^c_{h+1}+\Lambda}{B-\delta}) \\
&= rev(\text{greedy}(RP_\Lambda))(1+\frac{\delta}{B-\delta}) \\
&\leq rev(\text{greedy}(RP_\Lambda))(1+\frac{\mu^c_{h+1}+\Lambda}{B-\mu^c_{h+1}-\Lambda}),
\end{aligned} \tag{2.9}$$

where the second inequality follows from (2.7). Furthermore, for (2.9) we used the fact that $\delta < \mu^c_{h+1}+\Lambda$. If B is a growing function of n, as n increases the $\frac{\mu^c_{h+1}+\Lambda}{B-\mu^c_{h+1}-\Lambda}$ ratio decreases. More specifically is the limit of B as n goes to infinity is infinity, then $rev(\text{greedy}(RP_\Lambda))$ converges to $rev(\text{OPT}(RP_\Lambda))$ as n goes to infinity, by applying the squeeze theorem. □

2.3.3 Expected Regret Minimization

When the expected revenue and cost of each arm are known, Proposition 2.3.1 allows the oracle policy to pull the optimal set of arms for a given feasibility probability α. However, in reality, the expected revenue and cost of each arm are unknown and have to be learned. Fortunately, we can learn from the revenue and cost data that was observed in past periods. In the MAB problem, our goal is to develop an adaptive learning algorithm Π that minimizes the expected regret: the revenue loss due to not knowing the expected revenue and cost. Given that the oracle policy is the greedy solution to (RP_Λ), we define the (cumulative) expected regret as follows,

$$\mathbb{E}[Regret^\Pi_T] = \mathbb{E}\left[R^*_T - R^\Pi_T\right] = \sum_{t=1}^{T}\mathbb{E}\left[\sum_{i=1}^{h} r_{it}\mathbb{I}\left(\sum_{i=1}^{h} c_{it}\leq B\right) - \sum_{i=1}^{n} r_{it}x^\Pi_{it}\mathbb{I}\left(\sum_{i=1}^{n} c_{it}x^\Pi_{it}\leq B\right)\right], \tag{2.10}$$

where R^*_T represents the revenue of the oracle policy given α, and R^Π_T represents the revenue of the policy given by learning algorithm Π given α. In other words, we use R^*_T and x^*_{it} to denote the revenue and solution of the oracle, which knows the expected revenue and cost are known, while we use R^Π_T and x^Π_{it} for the algorithm, which does not know the expected revenue and cost.

2.4 Optimistic-Robust Learning Algorithm

In what follows, we propose the Optimistic-Robust Learning (ORL) algorithm Π that uses optimistic estimates of the expected revenues, and uses robust estimates of the expected costs. More precisely, for each arm i, the ORL algorithm uses the sample average of revenue and cost, $\overline{r}_{it}$ and $\overline{c}_{it}$, the oracle's robustness, $\Lambda \geq 0$, as well as a revenue bonus $\Gamma_{it} \geq 0$ and a cost bonus $\Delta_{it} \geq 0$. In every period t, the ORL algorithm solves the following optimistic-robust problem (ORP_Λ):

$$\max_{x_t \in \{0,1\}^n} \sum_{i=1}^{n} (\overline{r}_{it} + \Gamma_{it})\, x_{it}^{\Pi} \tag{2.11a}$$

$$\text{s.t.} \sum_{i=1}^{n} (\overline{c}_{it} + \Lambda + \Delta_{it})\, x_{it}^{\Pi} \leq B, \tag{2.11b}$$

where Γ_{it} and Δ_{it} are selected as later described in Theorem 2.5.1. Generally, the sample average of revenue, $\overline{r}_{it}$, is a consistent estimate of the expected revenue μ_i^r, but due to limited data the average might deviate considerably from the expectation. Therefore, we use $\overline{r}_{it} + \Gamma_{it}$ as an optimistic overestimate of μ_i^r. Being optimistic about the possible revenue of an arm encourages the algorithm to explore arms that have not been pulled before and is a central feature to learning in MAB problems. In this case the optimism comes in the form of an upper confidence bound on revenues as in other Upper Confidence Bound (UCB) type algorithms. Similarly, we use the sample average of cost, $\overline{c}_{it}$, as a consistent estimate for the expected cost μ_i^c. For the same reason that the average can differ from the expectation, we use $\overline{c}_{it} + \Lambda + \Delta_{it}$ as a robust overestimate of μ_i^c. As we do in the case where the expected revenues and costs are known, this allows us to remain robust and avoid the possibility that the total cost of the pulled arms exceeds the budget.

In contrast to Λ in the case where the expected revenues and costs are known, the parameters Γ_{it} and Δ_{it} can change with time t and the number of times arm i has been pulled, $\tau_{it} = \sum_{s=1}^{t-1} x_{is}^{\Pi}$. The intention is that we gain more trust in the sample averages as we gather more data about revenues and costs. This allows us to be less optimistic by shrinking Γ_{it} and less robust by shrinking Δ_{it}.

To describe the ORL algorithm in more detail, we presume that (ORP_Λ) is solved using the same greedy algorithm as is used when expected revenue and costs are known. This means that the algorithm will select the arms with the highest adjusted reward-to-cost ratios $\frac{\overline{r}_{it}+\Gamma_{it}}{\overline{c}_{it}+\Lambda+\Delta_{it}}$

until the budget is depleted. The ORL algorithm runs as follows for a given structure on the oracle's robustness Λ, the revenue bonus Γ_{it}, and the cost bonus Δ_{it}:

Algorithm 2.4.1. (Optimistic-Robust Learning)

1. Initialize the sample averages of revenue $\overline{r}_{i1}$ and cost $\overline{c}_{i1}$ by pulling each arm at least once.

2. Learn the optimal set of arms over the time horizon. For time period $t = 1, \ldots, T$:

   a. Select arms through solving (ORP_Λ). The optimal solution to (ORP_Λ) is given by setting $x_{it}^{\Pi} = 1$ for the arms with the largest estimated reward-to-cost ratios, $\frac{\overline{r}_{it}+\Gamma_{it}}{\overline{c}_{it}+\Lambda+\Delta_{it}}$, as long as $\sum_{i=1}^{n} (\overline{c}_{it} + \Lambda + \Delta_{it})\, x_{it}^{\Pi} \leq B$, and setting $x_{it}^{\Pi} = 0$ for all other arms. If the estimated cost of every arm exceeds the budget, we can still pick the arm i with the largest estimated reward-to-cost ratio, $\frac{\overline{r}_{it}+\Gamma_{it}}{\overline{c}_{it}+\Lambda+\Delta_{it}}$, because the realized cost of an arm is at most 1 and the budget is at least 1.

   b. Receive feedback r_{it} and c_{it} from their respective distributions.

   c. Update $\Gamma_{i,t+1}$ and $\Delta_{i,t+1}$ according to the learning policy. Update the number of pulls of arm i by time $t+1$ to $\tau_{i,t+1} = \sum_{s \leq t} x_{is}^{\Pi}$, the sample average of the revenue of arm i by time $t+1$ to $\overline{r}_{i,t+1} = \frac{\sum_{s \leq t} x_{is}^{\Pi} r_{is}}{\tau_{i,t+1}}$, and the sample average of the cost of arm i by time $t+1$ to $\overline{c}_{i,t+1} = \frac{\sum_{s \leq t} x_{is}^{\Pi} c_{is}}{\tau_{i,t+1}}$.

2.5 Regret Bound

In order to use the ORL algorithm, we need to specify how the parameters Γ_{it} and Δ_{it} are set. Theorem 2.5.1 establishes a theoretical bound on the expected regret of the ORL algorithm for the appropriate Γ_{it} and Δ_{it}. For ease of presentation, without loss of generality, we assume that all periods have the same budget value. We remark that our results are generalizable to the case that the period's budgets are fixed but not necessarily the same.

Theorem 2.5.1. Let r_{it} and c_{it} be independent and identical random variables supported on $[0,1]$ for any $i = 1, \ldots, n$ and $t = 1, \ldots, T$ and let $\eta = \min_{i=1,\ldots,n} \mu_i^c$. Let Γ_{it} and Δ_{it} used in the ORL algorithm be as follows

$$\Gamma_{it} = \frac{\overline{r}_{it}}{\overline{c}_{it} + \Lambda}\Delta_{it} + \frac{(\eta + \Lambda + 1)\,(\overline{c}_{it} + \Lambda + \Delta_{it})}{(\eta + \Lambda)(\overline{c}_{it} + \Lambda)}\sqrt{\frac{\ln(t)}{\tau_{it}}}, \text{ and } \Delta_{it} = -\min\{\sqrt[4]{\frac{1}{\tau_{it}}}, \Lambda\} + \sqrt{\frac{\ln(t)}{\tau_{it}}}.$$

Then, the expected regret of the ORL algorithm is bounded as follows:

$$\mathbb{E}[Regret_T^{\Pi}] \leq \sum_{i=1}^{h} \mu_i^r \left(\sum_{j=h+1}^{n} \left(\sum_{k=1}^{j} \left(2 \left(\frac{2\eta + 2\Lambda + 2}{\left(\frac{\mu_k^r}{\mu_k^c+\Lambda} - \frac{\mu_j^r}{\mu_j^c+\Lambda}\right)(\eta+\Lambda)^2} \right)^2 \ln(T) + \frac{4\pi^2}{3} \right) \right.\right.$$
$$+ \sum_{l=1}^{h+1} \frac{(h+1)^4}{(\sum_{i=1}^{h+1}(\mu_i^c + \Lambda) - B)^4} + \frac{\pi^2}{6} \Bigg) + \sum_{l=i+1}^{n} \left(\left(\frac{2\eta + 2\Lambda + 2}{\left(\frac{\mu_i^r}{\mu_i^c+\Lambda} - \frac{\mu_l^r}{\mu_l^c+\Lambda}\right)(\eta+\Lambda)^2} \right)^2 \ln(T) + \frac{2\pi^2}{3} \right)$$
$$+ \sum_{m=1}^{i} \left(\max_{1\leq k\leq m} \left\{ \max\{16\ln(T)^2, 4\frac{\ln(T)}{\Lambda^2}\}, \left(\frac{2\eta + 2\Lambda + 2}{\left(\frac{\mu_k^r}{\mu_k^c+\Lambda} - \frac{\mu_m^r}{\mu_m^c+\Lambda}\right)(\eta+\Lambda)^2} \right)^2 \ln(T) \right\} \right.$$
$$\left.\left. + \sum_{l=1}^{m-1} \frac{2\pi^2}{3} + \frac{\pi^2}{6} \right) \right),$$

that is, $\mathbb{E}[Regret_T^{\Pi}] \in O(\ln(T)^2)$.

Most importantly, we observe that the expected regret bound in Theorem 2.5.1 is polylogarithmic in time. As a function of time, this is close to the best bound on expected regret that can be achieved for many instances of MAB problems (that is, a logarithmic bound). In addition, it is important to note that the bound is polynomial in the number of arms. To understand this importance, consider an MAB problem that views each solution of (ORP_{Λ}) as an arm that yields some total revenue and incurs some total cost. Any MAB algorithm would play a single arm (a single solution) in every period and learn the specific total revenue and cost of that solution. Expected regret bounds that are known in the MAB literature would apply to this case. However, as there is an exponential number of solutions, these bounds are exponential in the number of targets. Instead, the above bound on expected regret is polynomial in the number of targets, which is important in the case of online advertising portfolios where there can be many targets.

Furthermore, this theorem functions as a guide to the right tuning of the parameters. We note that the revenue bonus Γ_{it} consists of two parts. The first term indicates a link between the need for optimism and robustness; if we become less robust, we also need to be less optimistic. The second term shows that as we pull an arm more (τ_{it} increases), we gather more data, and we have more trust that $\bar{r}_{it}$ is a good estimate of μ_i^r. Similarly, the cost bonus Δ_{it} also consists of two terms. The first part is a static robustness parameter with a similar function as that of Λ for the oracle policy; protecting us against the case where the to-be-realized cost c_{it} far exceeds the mean μ_i^c. The second part indicates that pulling an arm more often (τ_{it} increases), should give us more faith that $\bar{c}_{it}$ estimates μ_i^c correctly.

2.5.1 Proof of Theorem 2.5.1

In what follows, we prove the main result of Theorem 2.5.1. Initially, we decompose the expected regret into the probabilities of certain unlikely events. Afterwards, we prove in more detail how each of these probabilities are bounded. Taken together, this leads us to a bound on the expected regret.

Rewriting the expected regret. In this proof, we bound the expected regret (2.10) that is given by

$$\begin{aligned}\mathbb{E}\left[Regret_T^{\Pi}\right] &= \sum_{t=1}^{T} \mathbb{E}\left[\sum_{i=1}^{h} r_{it}\mathbb{I}\left\{\sum_{i=1}^{h} c_{it} \leq B\right\} - \sum_{i=1}^{n} r_{it}x_{it}^{\Pi}\mathbb{I}\left\{\sum_{i=1}^{n} c_{it}x_{it}^{\Pi} \leq B\right\}\right] \\ &= \sum_{t=1}^{T}\sum_{i=1}^{h} \mathbb{E}\left[r_{it}\left(\mathbb{I}\left\{\sum_{i=1}^{h} c_{it} \leq B\right\} - x_{it}^{\Pi}\mathbb{I}\left\{\sum_{i=1}^{n} c_{it}x_{it}^{\Pi} \leq B\right\}\right)\right] \\ &\quad - \sum_{t=1}^{T}\sum_{j=h+1}^{n} \mathbb{E}\left[r_{jt}x_{jt}^{\Pi}\mathbb{I}\left\{\sum_{i=1}^{n} c_{it}x_{it}^{\Pi} \leq B\right\}\right].\end{aligned}$$

First, we use the independence of the selection (x_{it}^{Π}), the realized revenue (r_{it}), and the realized cost (c_{it}) between arms,

$$\begin{aligned}\mathbb{E}\left[Regret_T^{\Pi}\right] &= \sum_{t=1}^{T}\sum_{i=1}^{h} \mu_i^r \mathbb{E}\left[\mathbb{I}\left\{\sum_{i=1}^{h} c_{it} \leq B\right\} - x_{it}^{\Pi}\mathbb{I}\left\{\sum_{i=1}^{n} c_{it}x_{it}^{\Pi} \leq B\right\}\right] \\ &\quad - \sum_{t=1}^{T}\sum_{j=h+1}^{n} \mu_j^r \mathbb{E}\left[x_{jt}^{\Pi}\mathbb{I}\left\{\sum_{i=1}^{n} c_{it}x_{it}^{\Pi} \leq B\right\}\right].\end{aligned}$$

Next, we use the definition of the expectation of indicator random variables,

$$\begin{aligned}\mathbb{E}\left[Regret_T^{\Pi}\right] &= \sum_{t=1}^{T}\sum_{i=1}^{h} \mu_i^r \mathbb{P}\left(\sum_{i=1}^{h} c_{it} \leq B \cap \left\{\sum_{i=1}^{n} c_{it}x_{it}^{\Pi} > B \cup x_{it}^{\Pi} = 0\right\}\right) \\ &\quad - \sum_{t=1}^{T}\sum_{i=1}^{h} \mu_i^r \mathbb{P}\left(\sum_{i=1}^{h} c_{it} > B \cap \left\{\sum_{i=1}^{n} c_{it}x_{it}^{\Pi} \leq B \cap x_{it}^{\Pi} = 1\right\}\right) \\ &\quad - \sum_{t=1}^{T}\sum_{j=h+1}^{n} \mu_j^r \mathbb{P}\left(\sum_{i=1}^{n} c_{it}x_{it}^{\Pi} \leq B \cap x_{jt}^{\Pi} = 1\right).\end{aligned}$$

Clearly, we can construct an upper bound to this expression by removing the negative terms, and we can use the union bound to split the probability,

$$\mathbb{E}\left[Regret_T^{\Pi}\right] \leq \sum_{t=1}^{T}\sum_{i=1}^{h}\mu_i^r \mathbb{P}\left(\sum_{i=1}^{h} c_{it} \leq B \cap \left\{\sum_{i=1}^{n} c_{it}x_{it}^{\Pi} > B \cup x_{it}^{\Pi} = 0\right\}\right) \tag{2.12}$$

$$\leq \sum_{t=1}^{T}\sum_{i=1}^{h}\mu_i^r \left(\mathbb{P}\left(\sum_{i=1}^{h} c_{it} \leq B \cap \sum_{i=1}^{n} c_{it}x_{it}^{\Pi} > B\right) + \mathbb{P}\left(\sum_{i=1}^{h} c_{it} \leq B \cap x_{it}^{\Pi} = 0\right)\right). \tag{2.13}$$

Bounding the expected regret by three unlikely events. We begin by bounding the first probabilistic term in (2.13). This term describes the case where the oracle's solution is feasible but the algorithm's solution turns out to be infeasible. In particular, we observe that if this probability is positive, then at least one of the arms $j = h+1, \ldots, n$ must be pulled. If this was not the case, then the realized cost of the pulled arms will be at most the realized cost of the first h arms, which is within budget, leading to a contradiction. Thus, using this fact and the union bound we obtain $\mathbb{P}\left(\sum_{i=1}^{h} c_{it} \leq B \cap \sum_{i=1}^{n} c_{it}x_{it}^{\Pi} > B\right) \leq \mathbb{P}\left(\bigcup_{j=h+1}^{n} x_{jt}^{\Pi} = 1\right) \leq \sum_{j=h+1}^{n} \mathbb{P}\left(x_{jt}^{\Pi} = 1\right)$. So, we need to analyze the probability that a suboptimal arm $j > h$ is pulled by the algorithm. Given that the suboptimal arm j is pulled there are two cases: i) either at least one better arm $i < j$ is not pulled, or ii) all better arms $i < j$ are pulled. These cases and the union bound yield the following bound,

$$\sum_{j=h+1}^{n} \mathbb{P}\left(x_{jt}^{\Pi} = 1\right) \leq \sum_{j=h+1}^{n}\left(\mathbb{P}\left(\left\{\bigcup_{i=1}^{j-1} x_{it}^{\Pi} = 0\right\} \cap x_{jt}^{\Pi} = 1\right) + \mathbb{P}\left(\bigcap_{i=1}^{j} x_{it}^{\Pi} = 1\right)\right)$$

$$\leq \sum_{j=h+1}^{n}\left(\sum_{i=1}^{j} \mathbb{P}\left(x_{it}^{\Pi} = 0 \cap x_{jt}^{\Pi} = 1\right) + \mathbb{P}\left(x_{1t}^{\Pi} = \ldots = x_{jt}^{\Pi} = 1\right)\right).$$

We then bound the second probabilistic term in (2.13) that describes the case where the oracle's solutions is feasible and there exists an optimal arm that is not picked by the ORL algorithm. The probability that both events described in this term occur is bounded above by the probability that one of the events occurs. Thus, we can analyze the following probability instead, $\mathbb{P}\left(\sum_{i=1}^{h} c_{it} \leq B \cap x_{it}^{\Pi} = 0\right) \leq \mathbb{P}\left(x_{it}^{\Pi} = 0\right)$. In contrast to before, we now need to analyze the probability that an optimal arm $i \leq h$ is not pulled by the algorithm. Again, given that the optimal arm i is not pulled we split into two cases: iii) either at least one suboptimal arm $j > h \geq i$ is pulled, or iv) not a single suboptimal arm $j > h \geq i$ is pulled. In conjunction with the union bound, this means we can analyze the following bound,

$$\begin{aligned}\mathbb{P}\left(x_{it}^{\Pi}=0\right) &\leq \mathbb{P}\left(x_{it}^{\Pi}=0 \cap \left\{\bigcup_{j=h+1}^{n} x_{jt}^{\Pi}=1\right\}\right)+\mathbb{P}\left(x_{it}^{\Pi}=0 \cap \left\{\bigcap_{j=h+1}^{n} x_{jt}^{\Pi}=0\right\}\right)\\ &\leq \sum_{j=h+1}^{n} \mathbb{P}\left(x_{it}^{\Pi}=0 \cap x_{jt}^{\Pi}=1\right)+\mathbb{P}\left(x_{it}^{\Pi}=x_{h+1,t}^{\Pi}=\ldots=x_{nt}^{\Pi}=0\right).\end{aligned}$$

With the two bounds provided by cases (i)-(iv) described above, we create an upper bound in terms of three unlikely events,

$$\mathbb{E}\left[Regret_T^{\Pi}\right] \leq \sum_{t=1}^{T}\sum_{i=1}^{h}\mu_i^r\left(\sum_{j=h+1}^{n}\sum_{k=1}^{j}\mathbb{P}\left(x_{kt}^{\Pi}=0\cap x_{jt}^{\Pi}=1\right)+\sum_{j=h+1}^{n}\mathbb{P}\left(x_{1t}^{\Pi}=\ldots=x_{jt}^{\Pi}=1\right)\right. \tag{2.14}$$

$$\left.+\sum_{j=h+1}^{n}\mathbb{P}\left(x_{it}^{\Pi}=0\cap x_{jt}^{\Pi}=1\right)+\mathbb{P}\left(x_{it}^{\Pi}=x_{h+1,t}^{\Pi}=\ldots=x_{nt}^{\Pi}=0\right)\right) \tag{2.15}$$

$$\leq \sum_{t=1}^{T}\sum_{i=1}^{h}\mu_i^r\sum_{j=h+1}^{n}2\sum_{k=1}^{j}\mathbb{P}\left(x_{kt}^{\Pi}=0\cap x_{jt}^{\Pi}=1\right) \tag{2.16}$$

$$+\sum_{t=1}^{T}\sum_{i=1}^{h}\mu_i^r\sum_{j=h+1}^{n}\mathbb{P}\left(x_{1t}^{\Pi}=\ldots=x_{jt}^{\Pi}=1\right) \tag{2.17}$$

$$+\sum_{t=1}^{T}\sum_{i=1}^{h}\mu_i^r\mathbb{P}\left(x_{it}^{\Pi}=x_{h+1,t}^{\Pi}=\ldots=x_{nt}^{\Pi}=0\right). \tag{2.18}$$

Proving bounds on the three unlikely events. Next, we show that each of these three probabilities are small. Before proving this, let us briefly discuss Lemma 2.5.2 that will be used later in the proofs. This lemma can be proved easily using Hoeffding's inequality.

Lemma 2.5.2. If r_{it} and c_{it} are independent and identically distributed random variables supported on $[0,1]$, for any $i=1,\ldots,n$ and $t=1,\ldots,T$, then

$$\mathbb{P}\left(|\bar{r}_{it}-\mu_i^r|>\sqrt{\frac{\ln(t)}{\tau_{it}}}\right)\leq\frac{2}{t^2} \text{ and } \mathbb{P}\left(|\bar{c}_{it}-\mu_i^c|>\sqrt{\frac{\ln(t)}{\tau_{it}}}\right)\leq\frac{2}{t^2}.$$

Bounding the probability of picking a suboptimal arm over an optimal arm. The event in (2.16) should be unlikely, because after pulling arms sufficiently enough we should learn that all arms $i<j$ are better than the suboptimal arm j. In terms of the algorithm, for a suboptimal arm j to be picked over a better arm $i<j$, we need that the estimated reward-to-cost ratio of arm j is better than that of arm i. In Lemma 2.5.3, we analyze this

event $\frac{\overline{r}_{it}+\Gamma_{it}}{\overline{c}_{it}+\Lambda+\Delta_{it}} \leq \frac{\overline{r}_{jt}+\Gamma_{jt}}{\overline{c}_{jt}+\Lambda+\Delta_{jt}}$ and we prove that its probability is small after pulling each arm sufficiently.

Lemma 2.5.3. Let $\Gamma_{it} = \frac{\overline{r}_{it}}{\overline{c}_{it}+\Lambda}\Delta_{it} + \frac{(\eta+\Lambda+1)(\overline{c}_{it}+\Lambda+\Delta_{it})}{(\eta+\Lambda)(\overline{c}_{it}+\Lambda)}\sqrt{\frac{\ln(t)}{\tau_{it}}}$, and let $L_{ijt} = \left(\frac{2\eta+2\Lambda+2}{\left(\frac{\mu_i^r}{\mu_i^c+\Lambda}-\frac{\mu_j^r}{\mu_j^c+\Lambda}\right)(\eta+\Lambda)(\overline{c}_{jt}+\Lambda)}\right)^2 \ln(t)$, then for all $1 \leq i < j \leq n$,

$$\mathbb{P}\left(x_{it}^{\Pi} = 0 \cap x_{jt}^{\Pi} = 1 \cap \tau_{jt} \geq L_{ijt}\right) \leq \frac{4}{t^2}.$$

Proof. See Appendix A.1.1. □

Bounding the probability of picking too many arms. Additionally, the event in (2.17) should be improbable, because we learn that pulling more than the optimal h arms is too costly after pulling arms sufficiently enough. Specifically, if all of the first j arms are pulled by the ORL algorithm, then their estimated cost should have been within the budget. In Lemma 2.5.4, this event is written as $\sum_{i=1}^{j}(\overline{c}_{it}+\Lambda+\Delta_{it}) \leq B$ and we prove that its probability of occurring is small.

Lemma 2.5.4. Let $\Delta_{it} = -\min\{\sqrt[4]{\frac{1}{\tau_{it}}}, \Lambda\} + \sqrt{\frac{\ln(t)}{\tau_{it}}}$, then for all $h < j \leq n$,

$$\mathbb{P}\left(x_{1t}^{\Pi} = \ldots = x_{jt}^{\Pi} = 1 \cap \forall 1 \leq l \leq h+1, \tau_{lt} \geq \frac{(h+1)^4}{(\sum_{i=1}^{h+1}(\mu_i^c+\Lambda)-B)^4}\right) \leq \frac{1}{t^2}.$$

Proof. See Appendix A.1.2. □

Bounding the probability of picking too few arms. Furthermore, the event in (2.18) will also be rare, as after pulling arms sufficiently enough we learn that the optimal h arms can be pulled without violating the budget. The event that an optimal arm i and all suboptimal arms are not pulled can only occur when we overestimate the cost of pulling arms. In Lemma 2.5.5, after pulling each arm sufficiently many times, we prove that this probability is small.

Lemma 2.5.5. Let $\Gamma_{it} = \frac{\overline{r}_{it}}{\overline{c}_{it}+\Lambda}\Delta_{it} + \frac{(\eta+\Lambda+1)(\overline{c}_{it}+\Lambda+\Delta_{it})}{(\eta+\Lambda)(\overline{c}_{it}+\Lambda)}\sqrt{\frac{\ln(t)}{\tau_{it}}}$, let $\Delta_{it} = -\min\{\sqrt[4]{\frac{1}{\tau_{it}}}, \Lambda\} +$

$\sqrt{\frac{\ln(t)}{\tau_{it}}}$, and let $L_{ijt} = \left(\frac{2\eta+2\Lambda+2}{\left(\frac{\mu_i^r}{\mu_i^c+\Lambda}-\frac{\mu_j^r}{\mu_j^c+\Lambda}\right)(\eta+\Lambda)(\bar{c}_{jt}+\Lambda)}\right)^2 \ln(t)$, then for all $1 \leq i \leq h$,

$$\mathbb{P}\left(x_{it}^{\Pi} = x_{h+1,t}^{\Pi} = \ldots = x_{nt}^{\Pi} = 0 \cap \forall l > i, \tau_{lt} \geq L_{ilt} \cap \forall m \leq i, \tau_{mt} \geq \max_{1\leq k\leq m}\left\{\max\{16\ln(t)^2, 4\frac{\ln(t)}{\Lambda^2}\}, L_{kmt}\right\}\right)$$
$$\leq \sum_{l=i+1}^{n}\frac{4}{t^2} + \sum_{m=1}^{i}\sum_{l=1}^{m-1}\frac{4}{t^2} + \sum_{m=1}^{i}\frac{1}{t^2}.$$

Proof. See Appendix A.1.3. □

Merging results to find the expected regret bound. With these three lemmas we can construct the upper bound stated in the theorem. We split each probability (if it is needed) into two cases: one where enough pulls are made and one where not enough pulls are made,

$$\begin{aligned}
\mathbb{E}\left[Regret_T^{\Pi}\right] \leq & \sum_{t=1}^{T}\sum_{i=1}^{h} 2\mu_i^r \sum_{j=h+1}^{n}\sum_{k=1}^{j}\mathbb{P}\left(x_{kt}^{\Pi} = 0 \cap x_{jt}^{\Pi} = 1 \cap \tau_{jt} < L_{kjt}\right) \\
&+ \sum_{t=1}^{T}\sum_{i=1}^{h} 2\mu_i^r \sum_{j=h+1}^{n}\sum_{k=1}^{j}\mathbb{P}\left(x_{kt}^{\Pi} = 0 \cap x_{jt}^{\Pi} = 1 \cap \tau_{jt} \geq L_{kjt}\right) \\
&+ \sum_{t=1}^{T}\sum_{i=1}^{h} \mu_i^r \sum_{j=h+1}^{n}\mathbb{P}\left(x_{1t}^{\Pi} = \ldots = x_{jt}^{\Pi} = 1 \cap \exists 1 \leq l \leq h+1, \tau_{lt} < \frac{(h+1)^4}{(\sum_{i=1}^{h+1}(\mu_i^c+\Lambda) - B)^4}\right) \\
&+ \sum_{t=1}^{T}\sum_{i=1}^{h} \mu_i^r \sum_{j=h+1}^{n}\mathbb{P}\left(x_{1t}^{\Pi} = \ldots = x_{jt}^{\Pi} = 1 \cap \forall 1 \leq l \leq h+1, \tau_{lt} \geq \frac{(h+1)^4}{(\sum_{i=1}^{h+1}(\mu_i^c+\Lambda) - B)^4}\right) \\
&+ \sum_{t=1}^{T}\sum_{i=1}^{h} \mu_i^r \mathbb{P}\left(x_{it}^{\Pi} = x_{h+1,t}^{\Pi} = \ldots = x_{nt}^{\Pi} = 0 \cap \exists l > i, \tau_{lt} < L_{ilt}\right) \\
&+ \sum_{t=1}^{T}\sum_{i=1}^{h} \mu_i^r \mathbb{P}(x_{it}^{\Pi} = x_{h+1,t}^{\Pi} = \ldots = x_{nt}^{\Pi} = 0 \\
&\qquad\qquad \cap \exists m \leq i, \tau_{mt} < \max_{1\leq k\leq m}\left\{\max\{16\ln(t)^2, 4\frac{\ln(t)}{\Lambda^2}\}, L_{kmt}\right\}) \\
&+ \sum_{t=1}^{T}\sum_{i=1}^{h} \mu_i^r \mathbb{P}(x_{it}^{\Pi} = x_{h+1,t}^{\Pi} = \ldots = x_{nt}^{\Pi} = 0 \cap \forall l > i, \tau_{lt} \geq L_{ilt} \\
&\qquad\qquad \cap \forall m \leq i, \tau_{mt} \geq \max_{1\leq k\leq m}\left\{\max\{16\ln(t)^2, 4\frac{\ln(t)}{\Lambda^2}\}, L_{kmt}\right\}).
\end{aligned}$$

Next, we use the lemmas to bound the probabilities where enough pulls are made. Additionally, we observe that arms that were pulled insufficiently could only have been pulled at most a logarithmic number of times as a function of T. Altogether, this results in the upper bound on the expected regret of the ORL algorithm that is stated in Theorem 2.5.1:

$$
\begin{aligned}
\mathbb{E}\left[Regret_T^{\Pi}\right] \leq & \sum_{i=1}^{h} 2\mu_i^r \sum_{j=h+1}^{n} \sum_{k=1}^{j} \sum_{t=1}^{T} \mathbb{P}\left(x_{kt}^{\Pi} = 0 \cap x_{jt}^{\Pi} = 1 \cap \tau_{jt} < L_{kjt}\right) \\
& + \sum_{i=1}^{h} 2\mu_i^r \sum_{j=h+1}^{n} \sum_{k=1}^{j} \sum_{t=1}^{T} \frac{4}{t^2} \\
& + \sum_{i=1}^{h} \mu_i^r \sum_{j=h+1}^{n} \sum_{l=1}^{h+1} \sum_{t=1}^{T} \mathbb{P}\left(x_{1t}^{\Pi} = \ldots = x_{jt}^{\Pi} = 1 \cap \tau_{lt} < \frac{(h+1)^4}{(\sum_{i=1}^{h+1}(\mu_i^c + \Lambda) - B)^4} \right) \\
& + \sum_{i=1}^{h} \mu_i^r \sum_{j=h+1}^{n} \sum_{t=1}^{T} \frac{1}{t^2} \\
& + \sum_{i=1}^{h} \mu_i^r \sum_{l=i+1}^{n} \sum_{t=1}^{T} \mathbb{P}\left(x_{it}^{\Pi} = x_{h+1,t}^{\Pi} = \ldots = x_{nt}^{\Pi} = 0 \cap \tau_{lt} < L_{ilt}\right) \\
& + \sum_{i=1}^{h} \mu_i^r \sum_{m=1}^{i} \sum_{t=1}^{T} \mathbb{P}(x_{it}^{\Pi} = x_{h+1,t}^{\Pi} = \ldots = x_{nt}^{\Pi} = 0 \\
& \qquad \cap \tau_{mt} < \max_{1 \leq k \leq m} \left\{ \max\{16\ln(t)^2, 4\frac{\ln(t)}{\Lambda^2}\}, L_{kmt} \right\}) \\
& + \sum_{i=1}^{h} \mu_i^r \sum_{t=1}^{T} \left(\sum_{l=i+1}^{n} \frac{4}{t^2} + \sum_{m=1}^{i} \sum_{l=1}^{m-1} \frac{4}{t^2} + \sum_{m=1}^{i} \frac{1}{t^2} \right) \\
\leq & \sum_{i=1}^{h} 2\mu_i^r \sum_{j=h+1}^{n} \sum_{k=1}^{j} L_{kjT} + \sum_{i=1}^{h} 2\mu_i^r \sum_{j=h+1}^{n} \sum_{k=1}^{j} \frac{2\pi^2}{3} + \sum_{i=1}^{h} \mu_i^r \sum_{j=h+1}^{n} \sum_{l=1}^{h+1} \frac{(h+1)^4}{(\sum_{i=1}^{h+1}(\mu_i^c + \Lambda) - B)^4} \\
& + \sum_{i=1}^{h} \mu_i^r \sum_{j=h+1}^{n} \frac{\pi^2}{6} + \sum_{i=1}^{h} \mu_i^r \sum_{l=i+1}^{n} L_{ilT} + \sum_{i=1}^{h} \mu_i^r \sum_{m=1}^{i} \max_{1 \leq k \leq m} \left\{ \max\{16\ln(T)^2, 4\frac{\ln(T)}{\Lambda^2}\}, L_{kmT} \right\} \\
& + \sum_{i=1}^{h} \mu_i^r \left(\sum_{l=i+1}^{n} \frac{2\pi^2}{3} + \sum_{m=1}^{i} \sum_{l=1}^{m-1} \frac{2\pi^2}{3} + \sum_{m=1}^{i} \frac{\pi^2}{6} \right) \\
\leq & \sum_{i=1}^{h} \mu_i^r \left(\sum_{j=h+1}^{n} \left(\sum_{k=1}^{j} \left(2 \left(\frac{2\eta + 2\Lambda + 2}{\left(\frac{\mu_k^r}{\mu_k^c + \Lambda} - \frac{\mu_j^r}{\mu_j^c + \Lambda} \right)(\eta + \Lambda)^2} \right)^2 \ln(T) + \frac{4\pi^2}{3} \right) \right. \right. \\
& \left. \left. + \sum_{l=1}^{h+1} \frac{(h+1)^4}{(\sum_{i=1}^{h+1}(\mu_i^c + \Lambda) - B)^4} + \frac{\pi^2}{6} \right) + \sum_{l=i+1}^{n} \left(\left(\frac{2\eta + 2\Lambda + 2}{\left(\frac{\mu_i^r}{\mu_i^c + \Lambda} - \frac{\mu_l^r}{\mu_l^c + \Lambda} \right)(\eta + \Lambda)^2} \right)^2 \ln(T) + \frac{2\pi^2}{3} \right) \right. \\
& + \sum_{m=1}^{i} \left(\max_{1 \leq k \leq m} \left\{ \max\{16\ln(T)^2, 4\frac{\ln(T)}{\Lambda^2}\}, \left(\frac{2\eta + 2\Lambda + 2}{\left(\frac{\mu_k^r}{\mu_k^c + \Lambda} - \frac{\mu_m^r}{\mu_m^c + \Lambda} \right)(\eta + \Lambda)^2} \right)^2 \ln(T) \right\} \right. \\
& \left. \left. + \sum_{l=1}^{m-1} \frac{2\pi^2}{3} + \frac{\pi^2}{6} \right) \right).
\end{aligned}
$$

In the following, we prove the three lemmas that we used to establish the result of Theorem 2.5.1.

2.6 Computational Experiments

In this section, we analyze the computational performance of the Optimistic-Robust Learning (ORL) algorithm against two benchmarks: one based on Sample Average Approximation (SAA) algorithms and another based on Upper Confidence Bound (UCB) algorithms. We compare against the SAA algorithm as similar methods are widely used in the current practice of online advertising portfolio optimization. Additionally, we compare against the UCB algorithm as variations have been applied extensively and successfully for MAB problems. These algorithms are described in more detail in Section 2.6.1.

In what follows, we run a variety of simulations to test the performance of these algorithms. First, we simulate based on synthetic data, where the revenues and costs are drawn from Bernoulli and beta distributions whose parameters are set randomly. Second, we simulate based on real-world data, where we use client data, provided by our industry partner that is an online advertising intermediary, to produce empirical distributions that the revenue and costs can be drawn from. In these simulations, we let each algorithm select their portfolio, randomly draw revenues and costs for the selected targets, and update their estimates. We iterate through time periods until we reach the time period T, the end of the time horizon. After the simulation, we can evaluate the sample average of each algorithm's regret, a consistent estimator of the expected regret. Sections 2.6.2 and 2.6.3 discuss results for synthetic and real-world datasets, respectively.

2.6.1 Benchmark Algorithms

Before discussing the results, we describe the SAA-based and UCB-based algorithms in more detail. The SAA type of algorithm only uses the sample average revenue $\overline{r}_{it}$ and sample average cost $\overline{c}_{it}$ on past data as an estimate of expected revenue and expected cost. In other words, this algorithm periodically solves (ORP_Λ) where $\Lambda = 0$, $\Gamma_{it} = 0$, and $\Delta_{it} = 0$ for all $i = 1, \ldots, n$ and $t = 1, \ldots, T$. Essentially, the SAA algorithm is a passive learning algorithm. It uses new data to update the revenue and cost estimates, but without a revenue bonus it is not encouraged to explore, nor is it robustified against high cost realizations. The SAA algorithm is a natural benchmark as it is easy to implement, and widely used in online advertising practice. The SAA algorithm runs as follows:

Algorithm 2.6.1. (Sample Average Approximation)

1. Initialize the sample averages of revenue $\overline{r}_{i1}$ and cost $\overline{c}_{i1}$ by pulling each arm at least once.

2. Learn the optimal set of arms over the time horizon. For time period $t = 1, \ldots, T$:

    a. Select arms by setting $x_{it}^{\Pi} = 1$ for the arms with the largest reward-to-cost ratios, $\frac{\overline{r}_{it}}{\overline{c}_{it}}$, as long as $\sum_{i=1}^{n} \overline{c}_{it} x_{it}^{\Pi} \leq B$, and setting $x_{it}^{\Pi} = 0$ for all other arms. If the sample average cost of every arm exceeds the budget, we pick the largest reward-to-cost ratio arm i, $\frac{\overline{r}_{it}}{\overline{c}_{it}}$.

    b. Receive feedback r_{it} and c_{it} from their respective distributions.

    c. Update the number of pulls of arm i by time $t+1$ to $\tau_{i,t+1} = \sum_{s \leq t} x_{is}^{\Pi}$, the sample average of the revenue of arm i by time $t+1$ to $\overline{r}_{i,t+1} = \frac{\sum_{s \leq t} x_{is}^{\Pi} r_{is}}{\tau_{i,t+1}}$, and the sample average of the cost of arm i by time $t+1$ to $\overline{c}_{i,t+1} = \frac{\sum_{s \leq t} x_{is}^{\Pi} c_{is}}{\tau_{i,t+1}}$.

The main issue with the SAA-based algorithm is that lacks the ability to explore, and hence, can easily get stuck in a locally optimal solution. For example, consider a valuable target with a large expected revenue and small expected cost, and suppose that the initial revenue and cost observations are small and large, respectively. This discourages the SAA algorithm from picking this target. If this continues, then it is plausible that the SAA algorithm discards this target permanently. To have a more powerful benchmark that does not fall into this trap, we develop a UCB-based algorithm (Auer et al. 2002, gives more details on UCB algorithms). The UCB type of algorithm uses not only the sample average revenue $\overline{r}_{it}$ and sample average cost $\overline{c}_{it}$, but also the revenue bonus Γ_{it}. This means that the UCB algorithm is actively learning in the revenue-space, however it is not accounting for the losses due to exceeding the budget. Every period, it solves (ORP_{Λ}) with $\Lambda = 0$, $\Gamma_{it} \geq 0$ and $\Delta_{it} = 0$ for all $i = 1, \ldots, n$ and $t = 1, \ldots, T$. The UCB algorithm proceeds as follows when given a particular policy for the revenue bonus Γ_{it}:

Algorithm 2.6.2. (Upper Confidence Bound)

1. Initialize the sample averages of revenue $\overline{r}_{i1}$ and cost $\overline{c}_{i1}$ by pulling each arm at least once.

2. Learn the optimal set of arms over the time horizon. For time period $t = 1, \ldots, T$:

    a. Select arms by setting $x_{it}^{\Pi} = 1$ for the arms with the largest augmented reward-to-cost ratios, $\frac{\overline{r}_{it} + \Gamma_{it}}{\overline{c}_{it}}$, as long as $\sum_{i=1}^{n} \overline{c}_{it} x_{it}^{\Pi} \leq B$, and setting $x_{it}^{\Pi} = 0$ for all other arms. If the sample average cost of every arm exceeds the budget, we pick the largest augmented reward-to-cost ratio arm i, $\frac{\bar{r}_{it}+\Gamma_{it}}{\bar{c}_{it}}$.

b. Receive feedback r_{it} and c_{it} from their respective distributions.

c. Update $\Gamma_{i,t+1}$ according to the given revenue bonus policy. Update the number of pulls of arm i by time $t+1$ to $\tau_{i,t+1} = \sum_{s\leq t} x^{\Pi}_{is}$, the sample average of the revenue of arm i by time $t+1$ to $\bar{r}_{i,t+1} = \frac{\sum_{s\leq t} x^{\Pi}_{is} r_{is}}{\tau_{i,t+1}}$, and the sample average of the cost of arm i by time $t+1$ to $\bar{c}_{i,t+1} = \frac{\sum_{s\leq t} x^{\Pi}_{is} c_{is}}{\tau_{i,t+1}}$.

In essence, the difference between the SAA and UCB algorithms is that the UCB algorithm explores the value of targets, due to the positive valued Γ_{it} being added to the estimated revenue. However, both algorithms do not protect against high costs, as the oracle's robustness Λ and the cost bonus Δ_{it} are removed.

2.6.2 Synthetic Data

In the first set of experiments on synthetic data, we draw the revenue and cost realizations for each target in each time period from certain classes of distributions. The distributions that we consider are the Bernoulli and beta distributions. The support of these distributions is in $[0,1]$ which corresponds to the assumption in Theorem 2.5.1 that revenues and costs are bounded in $[0,1]$.

In each experiment we fix a parameter setting (number of targets n, budget B, and feasibility probability α) and choose distribution family (Bernoulli or beta). In a particular experiment, we run 10 simulations where we change the parameters of each target's revenue and cost distributions. If a target's revenue and cost are Bernoulli-distributed we need to specify one parameter (probability parameter p), while if a target's revenue and cost are beta-distributed we need to specify two parameters (shape parameters α^{beta} and β^{beta}). Before specifying the parameters, we randomly draw the expected revenue μ^r_i and cost μ^c_i for each target i. In the case of a Bernoulli distribution, we set its probability parameter p equal to the expected value μ^r_i or μ^c_i, for revenue and cost respectively. In the case of a beta distribution, we set the shape parameter α^{beta} equal to $\frac{1}{2}$ and the shape parameter β^{beta} equal to $\frac{1-\mu^r_i}{2\mu^r_i}$ for revenue and $\frac{1-\mu^c_i}{2\mu^c_i}$ for cost. The reason for setting $\alpha^{beta} = \frac{1}{2}$ is that the distribution has a right-tail when $\beta^{beta} \geq 1$, and is U-shaped when $\beta^{beta} < 1$. Often, targets will generate little revenue and incur little cost implying a right-tailed distribution. On the other hand, the U-shaped distribution approximates the

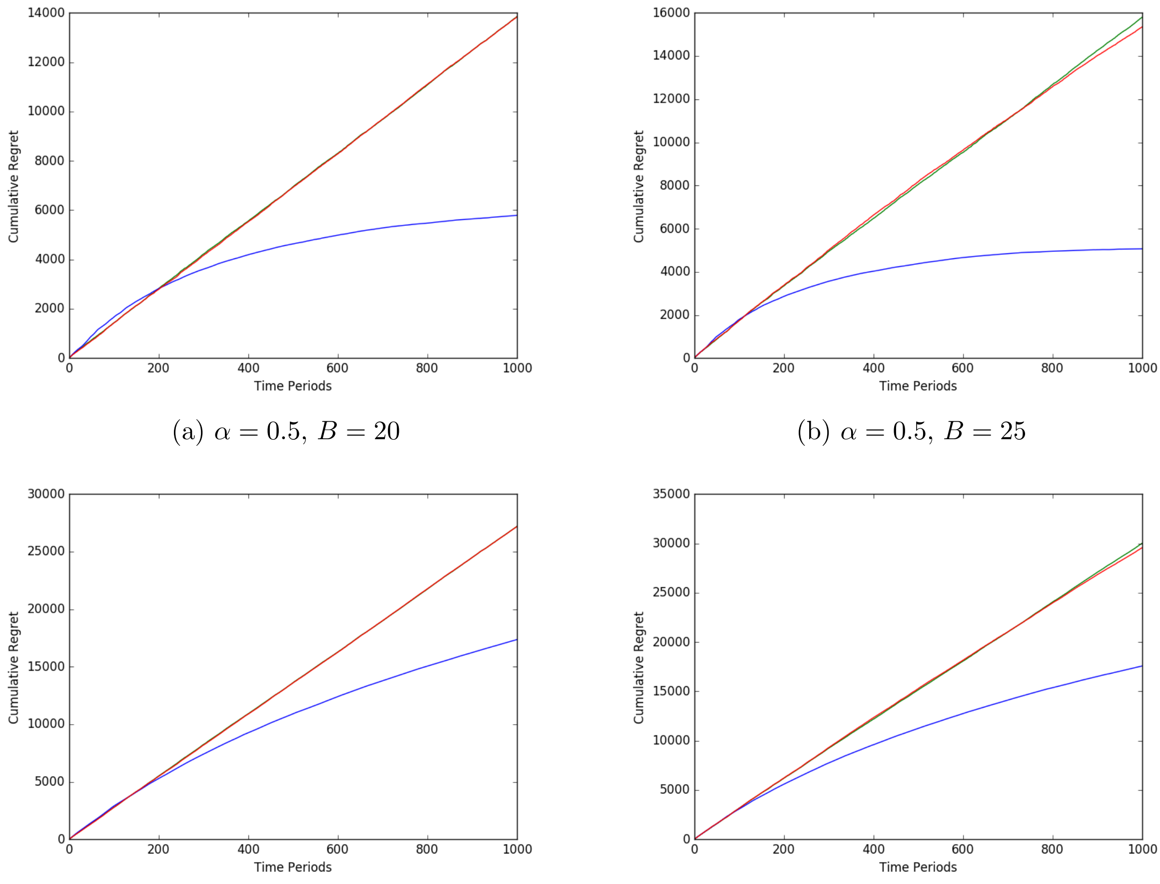
Bernoulli distribution. In particular, note that the beta distribution approaches the Bernoulli distribution when its shape parameters approach zero. Hence, one can regard the experiments under a Bernoulli distribution on revenues and costs as a special case of the experiments on the beta distribution.

As mentioned, in every experiment we fix the parameter setting as well as distribution family. In all experiments the number of targets is set as $n = 200$. We vary the budget B, the feasibility probability α, and the underlying distribution of costs and revenues between the Bernoulli and beta distributions. Within each experiment we run 10 simulations where different random seeds allow us to vary the revenue and cost distributions. In each time period of a simulation, we let the algorithms select their portfolios, draw the revenue and cost realizations from the distribution of every selected target, and update the estimates of each algorithm. Moreover, we compute the sample average of cumulative regret up to each time period. Thus, after T time periods, we obtain the sample average of cumulative regret for each algorithm.

Bernoulli-Distributed Revenues and Costs. First, we discuss the cumulative expected regret of the SAA, UCB and ORL algorithms when the revenues and costs of each target in each time period are drawn from the Bernoulli distribution. We select the Bernoulli distribution as its support is contained in the interval $[0, 1]$, which respects the modeling choices in Theorem 2.5.1. Additionally, it can model the realistic scenario where, if someone clicks on an advertisement, the advertiser needs to pay ($c_{it} = 1$), but the advertiser does not pay ($c_{it} = 0$) without a click. Similarly, it also models the case where, if someone clicks on the link and converts into a purchase, the advertiser gains revenue ($r_{it} = 1$), but the advertiser does not obtain any revenue ($r_{it} = 0$) without a conversion.

To inspect how the feasibility probability α affects the expected regret, we try two values of α, namely $\alpha = 0.5$ and $\alpha = 0.8$ (equivalent to $\Lambda = 0.59$ and $\Lambda = 0.33$). We also test the effect of the budget B on the expected regret, again trying two values of B, those are a high budget $B = 25$ and a low budget $B = 20$. For all the algorithms and parameter settings, we first pull each arm once to have initial estimates of the cost and revenue averages. Then, in each time period, the SAA, UCB and ORL algorithms determine which arms to pull, receive feedback on these pulls, and update their parameter estimates. This process repeats until we reach time period $T = 1000$.

Figure 2.3 shows the expected regret under the worst-case revenue loss function for two



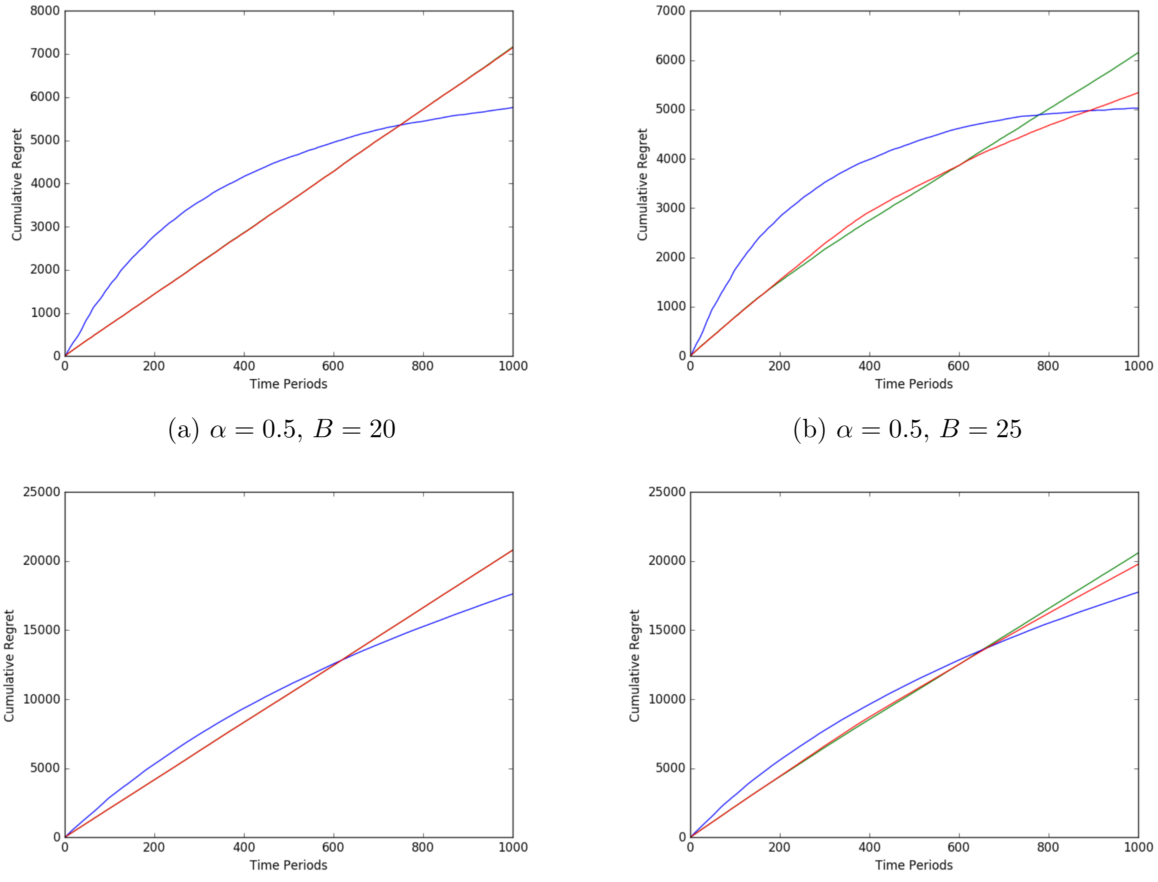
(a) $\alpha = 0.5,\ B = 20$ (b) $\alpha = 0.5,\ B = 25$

(c) $\alpha = 0.8,\ B = 20$ (d) $\alpha = 0.8,\ B = 25$

Figure 2.3: Expected cumulative regret under Bernoulli distributed revenues and costs with the worst-case loss function for different feasibility probabilities α and budgets B.

different feasibility probabilities α and budgets for the SAA, UCB and ORL algorithms. In these figures, the green curve corresponds to the SAA approach, the red curve corresponds to the UCB procedure, and the blue curve is associated to the ORL algorithm. The first pair of figures presents results for $\alpha = 0.5$, and the second pair presents results for $\alpha = 0.8$. In Figures 2.3(a) and 2.3(c), the green curve associated to the SAA algorithm is not visible as it closely coincides with the expected regret curve of the UCB algorithm. However, as the budget increases the two curves become distinguishable in Figures 2.3(b) and 2.3(d). Similarly, Figure 2.4 shows the expected regret with the fractional revenue loss function for two different feasibility probabilities α and budgets.

As expected, Figures 2.3 and 2.4 show that for both values of α and budgets as well as both the worst-case and the fractional loss functions, the ORL algorithm generates less regret and outperforms the SAA and UCB algorithms over the long run. Moreover, it can be seen that the SAA cumulative regret has a linear nature. This is in accordance with our expectation, because



(a) $\alpha = 0.5,\ B = 20$

(b) $\alpha = 0.5,\ B = 25$

(c) $\alpha = 0.8,\ B = 20$

(d) $\alpha = 0.8,\ B = 25$

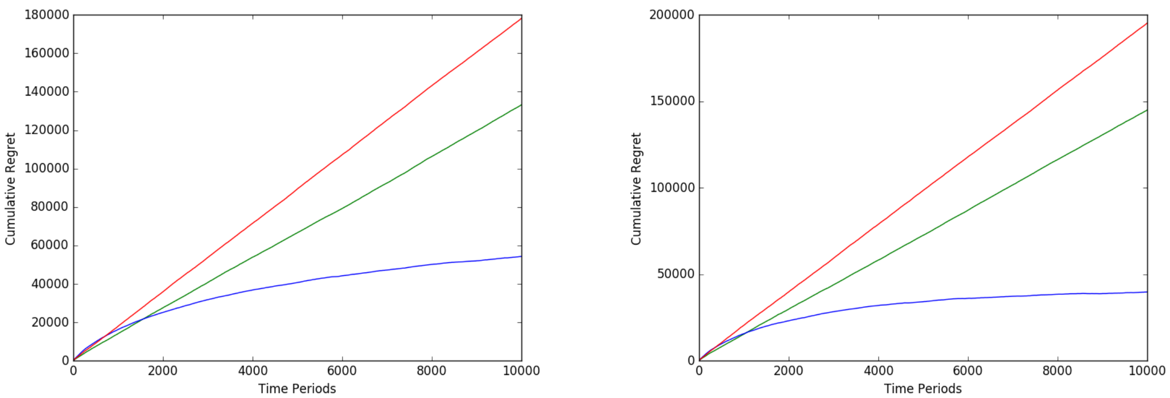
Figure 2.4: Expected cumulative regret under Bernoulli-distributed revenues and costs with the fractional loss function for different feasibility probabilities α and budgets B.

the SAA method is a greedy approach. This stems from the fact that if a target is not promising when it is initially picked by the SAA, then it could be discarded from further consideration. In other words, it avoids exploration and quickly starts to exploit the set of targets that seem most profitable. On the other hand, the UCB and ORL algorithms explore the value of targets since they are based on MAB algorithms. Therefore, it is expected that the performance of the UCB algorithm lies closer to that of the ORL algorithm than that of the SAA algorithm.

As the periodic budget B increases, this difference becomes more visible. The larger available budget allows for picking a larger portfolio, but this can also lead to more wrong choices that add to the regret. Observe that the gap between the SAA and UCB algorithms and the ORL algorithm increases over time. Unlike the expected regret curve of the SAA and UCB algorithms, the cumulative regret of ORL plateaus very quickly (after about 600 time periods) for $\alpha = 0.5$ regardless of budget and loss function. This efficiency of the ORL algorithm is due to its more sophisticated nature that allows exploring the arms that previously performed poorly.

Moreover, since the ORL algorithm is robust against large realized costs, it is less prone to violating the budget constraint and not gaining any revenue as a result. Whereas, over time, the SAA and UCB algorithms regularly incur regret due to violating the budget constraint. Finally, we observe that the ORL algorithm performs better when $\alpha = 0.5$. This indicates that robustifying too little against adverse costs is not as effective ($\alpha = 0.8$ or $\alpha = 1$ for the SAA and UCB algorithms). On the other hand, one can check that robustifying too much is too conservative leading to a weak expected regret. Overall, we note that an optimal α^* can be found through a line search on the value of α. Concluding, the tuned ORL algorithm improves significantly over the benchmark approaches, and these results are robust with respect to the feasibility probability α and the budget B.

Beta-Distributed Revenues and Costs. Next, we extend the simulations on synthetic data to targets with beta-distributed revenues and costs. The support of the beta distribution is also bounded on $[0, 1]$ corresponding to our assumptions in Theorem 2.5.1. However, in contrast to the Bernoulli distribution, the beta distribution is continuous and allows for more granular revenues and costs. These simulations are similar to those for the Bernoulli-distributed revenues and costs. Primarily, we can analyze whether the algorithms are robust to the revenue and cost distribution in terms of expected regret.



(a) $B = 20$ (b) $B = 25$

Figure 2.5: Expected cumulative regret under beta distributed revenues and costs for different budgets B.

Figure 2.5 shows the expected regret for two different budgets for the SAA, UCB, and ORL algorithms. As before, the green curve corresponds to the SAA algorithm, the red curve corresponds to the UCB algorithm, and the blue curve corresponds to the ORL algorithm. The primary observation is that the results on expected regret are consistent between the case of Bernoulli-distributed and beta-distributed revenues and costs. Again, the SAA and UCB algorithms incur significantly more regret than the ORL algorithm. This difference is particularly larger in the case where the budget is higher, as we observed in the case of the Bernoulli distribution.

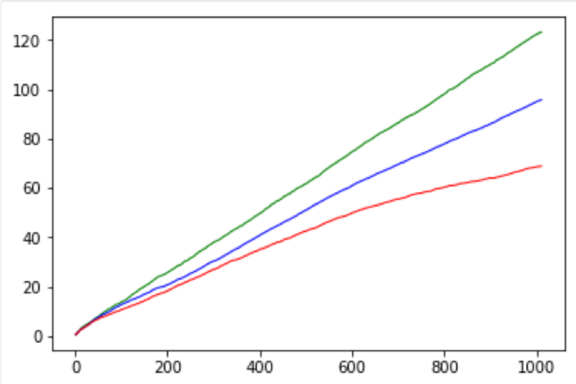
2.6.3 Real-World Data

In what follows, we use the advertising portfolio data of several clients, provided by our industry partner, to test the performance of the ORL algorithm against the benchmarks. Each observation in the dataset describes the associated client, target, time period, number of clicks, advertising revenue, and advertising cost. All datasets span a period of approximately half a year from December 2017 to June 2018. Altogether, the datasets contain more than 65,000 observations on average coming from at least 13,000 unique targets.

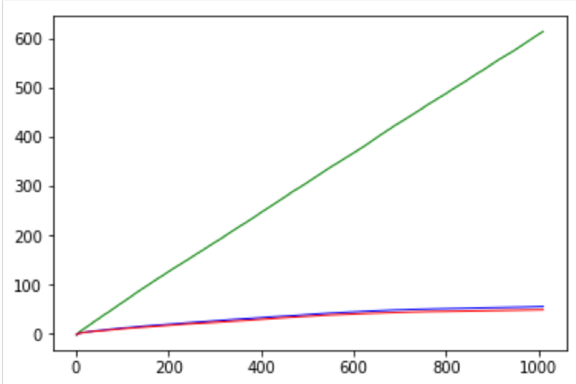
Notably, the available datasets are generated by the historical implementation of our industry partner's algorithm that makes bidding decisions for its clients. The workings of this algorithm are closer to that of the SAA-based algorithm, which is more likely to favor short-time exploitation over long-term exploration. This means that the generated data often features a small number of targets with many observations and a large number of targets with few observations. More specifically, out of about 6,500 targets, only 1,300 of them have at least one observation where revenues were obtained and costs incurred. As a consequence, to build a simulator that will present a fair measurement of the performance of the ORL algorithm, we filter out targets that have no revenue and cost data as they offer nothing to learn from.

We do not have access to the true underlying expected revenues and costs of each target, nor the true underlying distributions functions of revenue and cost. Hence, we are not able to draw the revenue and cost realizations in our simulation from their respective true underlying distributions. Instead, if a target has a reasonable amount of data, we can use the empirical distribution function as a reasonable estimate of the true distribution function. Therefore, in our simulations, we sample with replacement from the available data, which is similar to sampling from the empirical distribution function. By this method, we can create random sequences of revenues and costs for each target during the course of a simulated advertising campaign. In a similar vein, we view the empirical mean of revenue and cost of each target as the respective true expected revenue and cost.

On the subset of targets with more than one data point, we then perform the SAA, UCB



(a) $B = 1$



(b) $B = 10$

Figure 2.6: Expected cumulative regret on raw client data for feasibility probability $\alpha = 0.8$ and different budgets B on filtered data.

and ORL algorithms for both a small budget ($B = 1$) and a large budget ($B = 10$). Figure 2.6 demonstrates that, for a small budget, the UCB algorithm outperforms the ORL algorithm, while for a large budget, they have similar performance. This initial result might seem discouraging as it suggests that there is no need for cost robustification and the UCB algorithm performs sufficiently. However, focusing more on the client data reveals why this result should not be surprising. As mentioned earlier, the client data is created by our industry partner's bidding algorithm that acts similar to SAA. This algorithm is more likely to pick targets whose revenue and cost distributions have a reasonable mean and small variance. This is due to the fact that the SAA algorithm (and similar heuristics) lack sufficient exploration and exploit the data early in the process. Hence, they choose targets that demonstrate high estimated revenues and low estimated costs in the first few times they are picked and exploit this choice for the rest of the procedure. Due to lack of adequate exploration, such algorithms tend to ignore and not benefit from the targets with higher variance in their revenues and costs that also have high revenue and low cost expectations. Therefore, the bulk of targets of the client data have low variance in their costs and revenues as shown in Figure 2.7.

It is clear that if variances of costs of targets are low, then with high probability the realized cost values are close to their means and a learning algorithm's estimates. Thus, in such scenario, cost robustification of the ORL does not add much value to the algorithm and in fact it reserves a part of budget unused as buffer, while UCB utilizes that budget portion to bid on more targets and collect more revenue. This suggests that when variances of costs of targets are low, it is not essential to use a robust algorithm and UCB performs reasonably well. In other words, the power of the ORL algorithm is best observed when targets have higher variance in their costs.

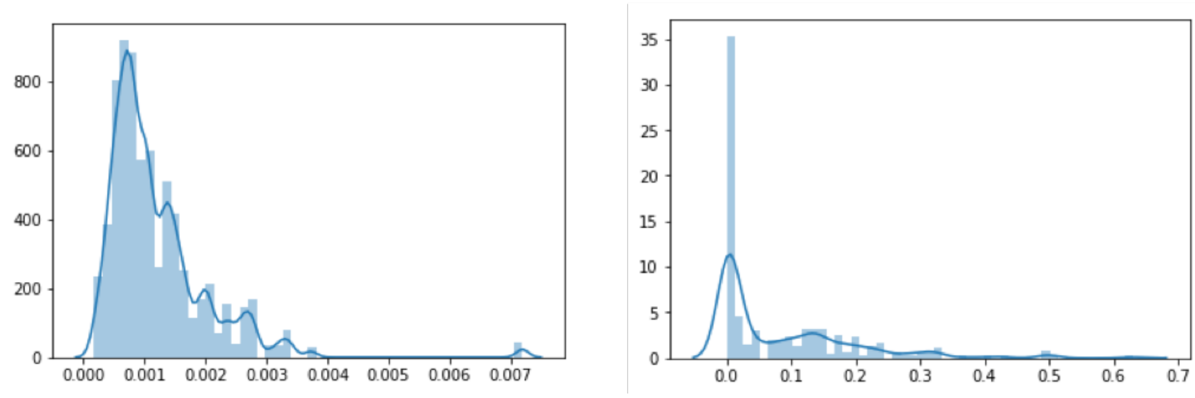
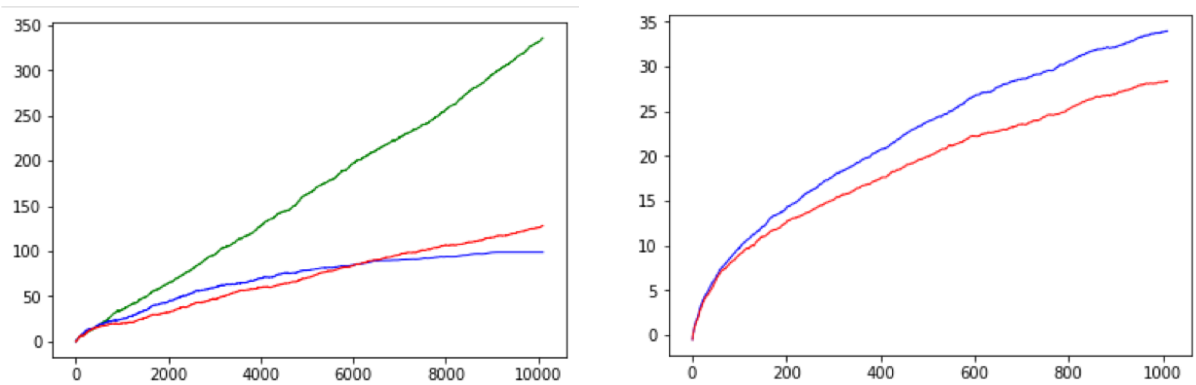


Figure 2.7: Majority of costs and revenues of targets of the client dataset have near-zero variance. Figure (a) shows an example for how costs/revenues of majority of the targets are distributed. Only a small fraction of these targets (less than 1%) have notable variance in their costs or revenues, see Figure (b) for an example of cost/revenue distribution of these targets.

Due to the nature of its creation, the client dataset does not provide us with high variance cost data points, while in real-world cases, revenues and costs of targets do not always have concentrated distributions. To make our dataset a better approximation of real-world scenarios, we artificially create targets with high revenue and cost variance in two ways: increasing the filtering threshold on the number of observations per target, and artificially increasing variance of revenues and costs for these targets by randomly coupling targets with each other.



(a) $B = 1$ (b) $B = 5$

Figure 2.8: Total expected regret on filtered client data (targets with less than 80 data points are filtered out, then randomly picked pairs of targets are coupled) for feasibility probability $\alpha = 0.8$ and different budgets.

Figure 2.8 shows the expected cumulative regret of the SAA, UCB and ORL algorithms on the dataset created after performing the above two manipulations on the client dataset. In particular the minimum number of data points for each targets is set as 80 and pairs of targets chosen uniformly at random are coupled to each other. It can be seen that for low budget (i.e., $B = 1$) the ORL algorithm performs better than the UCB algorithm, and for large budget (i.e., $B = 5$) the performance of both algorithms are close with UCB performing slightly better. This is not surprising as when the budget increases the number of targets that one can bid on also increases. Hence when the costs of some targets go above their estimates, it is likely that there exist other chosen targets whose costs are below their estimates, resulting in the total cost staying within the budget. Moreover, we want to remark that for these simulations we assumed that at the beginning of the process, the algorithms do not have access to any revenue or cost data. Therefore, as time goes on they gradually learn about the revenues and costs of the targets they select in their portfolios. This is not the case in real-world scenarios. In practice, at the beginning of the advertising campaign there is some available data of the revenue and cost of some targets from previous campaigns that can be used to further improve the portfolio selection process. Hence the performance of the ORL algorithm in practice is even better than what is depicted in Figure 2.8. In particular, in the case of Figure 2.8(a), the ORL algorithm starts further ahead in time in practice and outperform the SAA and UCB algorithms by higher margins.

2.7 Conclusions

In this chapter, we studied the problem of learning optimal online advertising portfolios with periodic budgets. In this problem, during an advertising campaign an advertiser aims to learn the best portfolio of targets to show its ads to with the goal of maximizing its total expected revenue. The advertising campaign is divided into smaller periods (hours or days), each with an advertising budget that cannot be exceeded by the advertising costs of that period. At the beginning of each period, the advertiser selects the portfolio of targets to show the ads to and collects information about the revenues and costs of those targets. This information is used to modify the target portfolio for the next period. Due to the uncertainties in the costs and revenues of targets, we used the MAB framework to explore the targets and exploit the most profitable set of them. We developed the ORL algorithm to address the trade-off between exploration and exploration, and we show that the ORL algorithm obtains polylogarithmic total expected regret. Moreover, we tested the performance of this algorithm and compared it with two standard learning benchmarks (the SAA and UCB algorithms) on synthetic and real-world datasets. We observed that the the ORL algorithm outperforms these benchmarks in both settings.

Chapter 3

Scheduling Promotion Vehicles to Boost Profits

3.1 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 Chapter 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 chapter 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 work 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 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).

3.1.1 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 work, 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 scheduling 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 3.3, we try to fit our data to both additive (linear) and multiplicative (non-linear) 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), that 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; (ii) Imposing a limit on the number of promotion vehicles that can be assigned to each time period. In Sections 3.2 and 3.3.3, we present a formal description of this model, its main business rules, and the relevant mathematical notation.

3.1.2 Contributions

From a theoretical perspective, the main contribution of this chapter 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 3.2 and 3.3, whereas the resulting optimization problem is formally described in Section 3.3.3.

- *Complexity results.* We show that, unlike standard (linear) matching problems, introducing 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 3.4.1.

- *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 3.4.2). 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 3.4.3). Finally, we also show that the special case when the vehicle boosts are uniform admits a polynomial-time approximation scheme (PTAS) (see Appendix B.4).

- *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.2). We also test this model computationally in Section 3.5.2, 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 3.5). 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 between 2% and 9% relative to current practice (see Section 5.5).

3.1.3 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. (2017b). In this work, we focus on tackling the latter task of scheduling promotion vehicles.

Sales promotions have extensively been studied in the literature, mostly in marketing and economics. We refer the reader to Blattberg and Neslin (1990) and to the references therein for a comprehensive review. However, with regards 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, 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 work, 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. 2017b). 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 multiplicative and additive forms for the promotion vehicle dependence.

Optimizing 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. (2017b), 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. (2017b) focus on the price promotion problem, and do not consider the question of how to effectively schedule promotion vehicles. In this work, 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, that has received a great deal of attention by 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 and capacity, assortment, pricing, and scheduling promotion vehicles. 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 therin), and pricing decisions (see, e.g., Phillips 2005, Cohen et al. 2017b). It is also worth noting that several prescriptive works in the marketing community study the impact of retail coupons (see, for example, 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 work 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 work 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, non-specialists 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, for example, Cormen et al. (2009) chapter 16, Kleinberg and Tardos (2005) chapter 4, and Williamson and Shmoys (2010), chapters 2 and 9), as well as integer programming (Schrijver 1998, Wolsey and Nemhauser 1999, Bertsimas and Weismantel 2005).

As previously mentioned, the concrete optimization problem considered in this work (see Section 3.3.3) 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 non-linear formulation. From this perspective, the problem of optimizing an arbitrary non-linear function over the bipartite matching polytope is known to be NP-hard (see, for instance, Chandrasekaran et al. 1982, Berstein and Onn 2008). To our knowledge, Section 3.4.1 and Appendix B.2.2, where we connect between the promotion vehicle scheduling problem and detecting large independent sets in certain graph classes, provide new inapproximability bounds for non-linear 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 (non-linear) 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 non-linear 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.

3.2 General Modeling Approach

In this chapter, we consider the problem formulation (P), formally defined in Section 3.3.3, 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 3.7 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 5 to 40 distinct promotion vehicles, examples include placing products at the end of an aisle (end-cap-display), dedicating an in-aisle display, sending our flyers, broadcasting TV commercials, radio advertisements, tasting stands, and in-store flyers. In our model, the retailer does not decide on prices. The reason for this is that the promotion price optimization and promotion vehicle scheduling are generally solved by different departments of the retailer. Hence, the focus of this chapter is on the promotion vehicle scheduling problem which has not been studied rigorously before. In order 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 $d_t(\cdot)$ 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 3.3, motivated by real data, we propose a class of models that aims 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 3.3 serve as justification for this assumption. In addition, we assume that the retailer has sufficient inventory to meet demand, so that predicting sales and demand are equal. This assumption is reasonable for grocery retailers, and in particular for non-perishable 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 4 in-store flyers and 2 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:

$$\sum_{t=1}^{T} x_{vt} \leq C_v; \quad \forall v \in V. \tag{3.1}$$

2. *Limited number of vehicles per time period.* For instance, during a given week, the retailer can use at most 4 vehicles. During another week, in which a holiday event occurs, at most 7 vehicles can be used. These rules are usually known upfront for the entire selling season. One can impose the following constraint in the formulation:

$$\sum_{v \in V} x_{vt} \leq L_t; \quad \forall t \in [T]. \tag{3.2}$$

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 to run 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 work.

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:

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

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 3.3.3, we explain how to easily incorporate constraints of the form $x_{vt} = 1$ or $x_{vt} = 0$ using basic preprocessing ideas.

3.3 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 $\{x_{vt}\}_{v \in V}$.

3.3.1 Data Description

Our dataset contains data collected from 18 stores of a large supermarket client of the Oracle Retail Global Business Unit. This dataset spans a period of roughly 2 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.

In order 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 3.2, especially for a non-perishable 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 4 weeks, and the 21 different promotion vehicles. Some of the most efficient promotion vehicles found in our dataset 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*), 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*).

3.3.2 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 Chapter 12 of Blattberg and Neslin (1990)). This effect occurs very often when consumers purchase larger quantities during price promotions. As a result, consumers stockpile the item for future consumption, especially for non-perishable 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) and 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 work, 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 following additive demand model:

$$d_t^{s,i} = \beta_0^{A,s,i} + \beta_1^A t + \beta_2^{A,i} p_t^{s,i} + \beta_3^{A,i} p_{t-1}^{s,i} + \sum_{v=1}^{19} \gamma_v^A x_{vt}^{s,i} + \epsilon_t^{s,i}, \tag{3.3}$$

and the following multiplicative demand model:

$$\log(d_t^{s,i}) = \beta_0^{M,s,i} + \beta_1^M t + \beta_2^{M,i} \log(p_t^{s,i}) + \beta_3^{M,i} \log(p_{t-1}^{s,i}) + \sum_{v=1}^{21} \gamma_v^M x_{vt}^{s,i} + \epsilon_t^{s,i}. \tag{3.4}$$

Due to the sparsity of our dataset, we are unable to estimate cross-terms 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 work 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 better fit to the data. To compare the additive and multiplicative demand models, we compute out-of-sample forecasting metrics for both models. Table 3.1 presents the out-of-sample R^2, Mean Absolute Percentage Error (MAPE), and Mean Absolute Error (MAE) of the

[1] We tried both AIC and BIC sequentially in order 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.

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 3.1: 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

Table 3.1 demonstrates that the multiplicative model (3.4) outperforms the additive model (3.3) in all forecasting metrics. In fact, the differences in these metrics are substantial, which lead 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 (3.4) 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 chapter we consider a broader class of models that subsume the multiplicative log-linear model (3.4) as a special case. The specifics of this class, which is referred to as the *multiplicative model*, are presented in Section 3.3.3. The alternative way of modeling the promotion vehicle scheduling problem, by considering an additive model, is reported in Appendix B.3. Even though the computational problem resulting from the latter class can be optimized efficiently, our analysis shows a worse fit to actual retail data.

3.3.3 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:

$$d_t(p_t, \{x_{vt}\}_{v\in V}) = h_t^M(p_t) \cdot \prod_{v\in V} B_{vt}^{x_{vt}}. \tag{3.5}$$

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 (3.5) that includes cross-term effects (i.e., interactions between pairs of vehicles) in Appendix B.2.

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

$$\begin{aligned}
(P) \quad \max \quad & \sum_{t=1}^{T} \alpha_t \prod_{v\in V} B_{vt}^{x_{vt}} \\
(C_1) \quad & \sum_{t=1}^{T} x_{vt} \leq C_v \qquad \forall\, v \in V \\
(C_2) \quad & \sum_{v\in V} x_{vt} \leq L_t \qquad \forall\, t \in [T] \\
(C_3) \quad & x_{vt} \in \{0,1\} \qquad \forall\, v \in V,\, t \in [T]
\end{aligned}$$

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 5 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 3.4.3. 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 (3.1) and (3.2). It is not difficult to verify that one can perform simple preprocessing modifications in order to capture constraints of the form $x_{vt} = 1$ or $x_{vt} = 0$, mentioned in Section 3.2. 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 LP 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 chapter.

3.4 Hardness and Approximability

3.4.1 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 3.4.1. 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:

$$\begin{aligned} (P) \quad \max \quad & \sum_{v \in N} |N|^{2 \sum_{e \in S_v} x_{ev}} \\ (C_1) \quad & \sum_{v \in N} x_{ev} \leq 1 && \forall\, e \in E \\ (C_2) \quad & \sum_{e \in E} x_{ev} \leq \Delta && \forall\, v \in N \\ (C_3) \quad & x_{ev} \in \{0, 1\} && \forall\, v \in N,\, e \in E \end{aligned}$$

Claim 3.4.2. Let U^* be a maximum-cardinality independent set in G. Then, $\text{OPT}(P) \geq |N|^{2\Delta} \cdot |U^*|$.

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:

$$\sum_{v \in N} |N|^{2 \sum_{e \in S_v} x_{ev}} \geq \sum_{v \in U^*} |N|^{2 \sum_{e \in S_v} x_{ev}} = \sum_{v \in U^*} |N|^{2|S_v|} = |N|^{2\Delta} \cdot |U^*| \ .$$

□

Claim 3.4.3. 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 $\frac{\mathcal{V}(x)}{|N|^{2\Delta}} - \frac{1}{|N|}$.

Proof. The important observation to make notice of is that, for every vertex $v \in N$, since $|S_v| = \Delta$ we have $|N|^{2 \sum_{e \in S_v} x_{ev}} \leq |N|^{2\Delta}$. 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 $|N|^{2 \sum_{e \in S_v} x_{ev}} \leq |N|^{2(\Delta-1)}$. Consequently, letting U_x denote the set of vertices for which equality holds,

$$|U_x| \cdot |N|^{2\Delta} \geq \sum_{v \in N} |N|^{2 \sum_{e \in S_v} x_{ev}} - |N \setminus U_x| \cdot |N|^{2(\Delta-1)} \geq \mathcal{V}(x) - |N|^{2\Delta-1} \ ,$$

meaning that $|U_x| \geq \frac{\mathcal{V}(x)}{|N|^{2\Delta}} - \frac{1}{|N|}$. 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. □

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 $\beta - \frac{1}{|N|}$, as one is able to compute an independent set of cardinality at least $\frac{\beta \cdot |N|^{2\Delta} \cdot |U^*|}{|N|^{2\Delta}} - \frac{1}{|N|} \geq (\beta - \frac{1}{|N|}) \cdot |U^*|$. 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. □

3.4.2 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 3.4.1. A result of this nature is not known at 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 $1, \dots, s-1$. 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 $1, \dots, s-1$. Our algorithm proceeds as follows:

1. We initialize $C_{1,v} = C_v$ for any vehicle v, with all time periods active, i.e., $A_1 = \{1, \dots, T\}$.
2. For $s = 1, \dots, T$:
   (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 non-increasing 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 $A_{s+1} \leftarrow A_s \setminus \{t^*\}$), 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 3.1. 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 3.1a 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 3.1a (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 3.1b, 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 continues this procedure until either all time periods become inactive or all vehicles are depleted.

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	

(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	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	

(b) Step 2

Figure 3.1: Example steps of the greedy algorithm

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

Theorem 3.4.4. 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 3.4.4 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 $v \in U_s^{t^*} \setminus R_{t^*}^*$, 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 3.4.5 below.

Lemma 3.4.5. 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 non-increasing 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 $v \in V$:

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

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

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,

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

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

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

□

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 $|U_s^{t^*} \setminus R_{t^*}^*| \leq |U_s^{t^*}| \leq \Delta$ in item 2). Therefore, summing the profits obtained by the greedy algorithm over all steps, we obtain a combined profit of at least $\frac{1}{\Delta+1}$ times the total profit of the optimal solution. □

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 B.1.1, 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 3.4.6. 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 Appendix B.4, 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., $\alpha_1 = \cdots = \alpha_T$).

3.4.3 Approximate Integer Program

As the tight example in Lemma 3.4.6 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 non-linear 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 3.4.7. 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, $\mathrm{OPT}(IP_\epsilon) \geq (1-\epsilon) \cdot \mathrm{OPT}(P)$, 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 non-negative real numbers, with $0 \in \mathcal{D}$. With respect to this set, our integer program (IP_ϵ) is defined as follows:

$$
\begin{array}{lll}
(IP_\epsilon) \quad \max \displaystyle\sum_{t=1}^{T} \alpha_t \sum_{r \in \mathcal{D}} (e^r \cdot y_{tr}) & & \\
(C_1) \quad \displaystyle\sum_{t=1}^{T} x_{vt} \leq C_v & & \forall\, v \in V \\
(C_2) \quad \displaystyle\sum_{v \in V} x_{vt} \leq L_t & & \forall\, t \in [T] \\
(C_3) \quad \displaystyle\sum_{r \in \mathcal{D}} y_{tr} = 1 & & \forall\, t \in [T] \\
(C_4) \quad y_{tr} \leq \dfrac{1}{r} \displaystyle\sum_{v \in V} \ln(B_{vt}) \cdot x_{vt} & & \forall\, t \in [T],\, r \in \mathcal{D} \setminus \{0\} \\
(C_5) \quad x_{vt},\, y_{tr} \in \{0,1\} & & \forall\, v \in V,\, t \in [T],\, r \in \mathcal{D}
\end{array}
$$

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 under-estimate 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 under-estimate 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 $y_{tr} \leq \frac{1}{r} \sum_{v \in V} \ln(B_{vt}) \cdot x_{vt}$ 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, $B^+_{\min} = \min[\{B_{vt} : v \in V, t \in [T]\} \cap (1, \infty)]$. By our initial assumption (see Section 3.3.3), the latter set is indeed non-empty. 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 $\left[\ln(B^+_{\min}), \Delta \cdot \ln(B_{\max})\right] \subseteq (0, \infty)$ is geometrically partitioned by powers of $1 + \frac{1}{M}$, where $M = \frac{\Delta}{\epsilon} \cdot \ln(B_{\max})$. In other words,

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

Proof. Theorem 3.4.7, 1. This part of the theorem is rather straightforward. In order 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:

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

□

Proof. Theorem 3.4.7, 2. To prove that $\mathrm{OPT}(IP_\epsilon) \geq (1-\epsilon)\cdot \mathrm{OPT}(P)$, 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 $(1-\epsilon)\cdot \mathrm{OPT}(P)$.

To this end, for every time period t, let

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

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 $0 \in \mathcal{D}$. Furthermore, the objective value of (x^*, y) with respect to (IP_ϵ) is precisely

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

In order to attain a lower bound on the latter expression, we prove the following lemma.

Lemma 3.4.8. For every time period t,

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

Proof. See Appendix B.1.2. □

As a result, we have just shown that:

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

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

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

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

It is worth pointing out that we consider in Appendix B.2 an extension of the demand model (3.5) 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.

3.5 Computational Experiments

In this section, we conduct extensive computational experiments to evaluate the algorithms developed in Section 3.4 and Appendix B.2 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 3.4.2) and the approximate integer program (Section 3.4.3) 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 5, even though this number can easily exceed 20 in practice (see additional details in Section 5.5).

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.2.4). 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@3.5GHz CPU and 8GB 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.

3.5.1 Performance without Cross-Terms

We first test our algorithms on a base setting, and later on 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 3.2 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.

Table 3.2: 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	3726.82	0.9979	0.9814
Approximate IP ($\epsilon = 0.10$)	18215.30	58011.31	0.9997	0.9981
Approximate IP ($\epsilon = 0.05$)	45129.46	223221.91	1	1

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.

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 1 second, 5 seconds, 1 minute, 5 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 3.3.

Table 3.3: Performance of the greedy algorithm and approximate IP for different termination times ($T = 13$, $|V| = 5$, $L_t = 2$, $C_v = 2$, $\alpha_t \sim U[1, 2]$, $B_{vt} \sim U[1, 2]$, $\epsilon = 0.05$)

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

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 1 minute. It is important to note that the performance improves significantly before the 1 minute mark, after which it grows slowly as the termination time is increased. Therefore, we terminate the approximate IP with $\epsilon = 0.05$ after 1 minute in further experiments.

So far, the parameters L_t and C_v were constant in all 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 3.4, we present the results over 200 random instances.

Table 3.4: Performance of the greedy algorithm and approximate IP on fully randomized instances ($T = 13$, $|V| = 5$, $L_t \sim U\{1, \ldots, 5\}$, $C_v \sim U\{1, \ldots, 13\}$, $\epsilon = 0.05$)

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

Table 3.4 demonstrates that the approximate IP, terminated at 1 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 3.4 considers with 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 performances 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 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 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.

3.5.2 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.2.4). In order 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 3.2 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 3.2 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

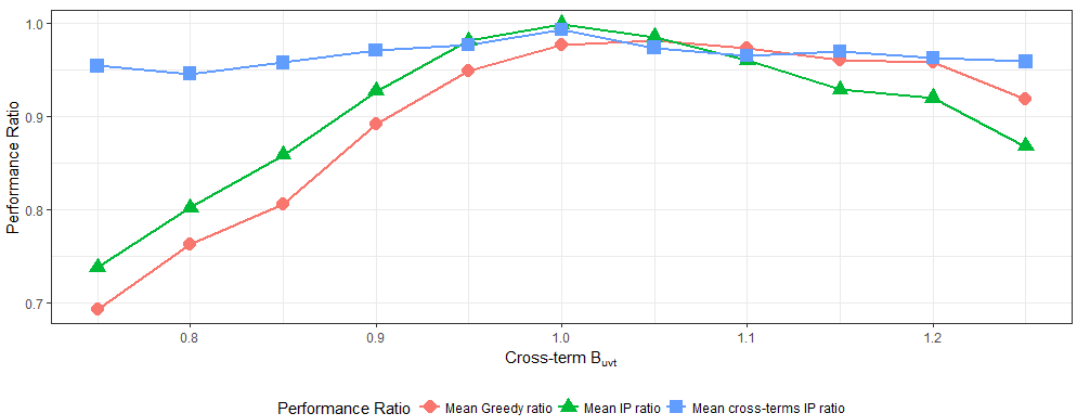


Figure 3.2: Average performance ratios of the greedy algorithm, approximate IP without cross-terms, and approximate IP with cross-terms relative to the optimal objective

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.

3.6 Case Study

In what follows, we study how our methodology performs in practice based on an actual case study. Our dataset is described in Section 3.3 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.

3.6.1 Promotion Vehicle Boost Estimation

As shown in Section 3.3.2, the multiplicative model yields a good predictive accuracy out-of-sample (with R^2 between 0.68 and 0.94 for the 3 items we considered). Specifically, we focus our attention on the following log-linear demand model, similar to the one described in (3.4):

$$\log(d_t^{s,i}) = \beta_0^{s,i} + \beta_1 t + \beta_2^i \log(p_t^{s,i}) + \beta_3^i \log(p_{t-1}^{s,i}) + \sum_{v=1}^{21} \gamma_v x_{vt}^{s,i} + \epsilon_t^{s,i}. \tag{3.6}$$

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 3.3.2, we assume that the boost of vehicle v is time independent, i.e., $\gamma_{vt} = \gamma_v$ for all t, as our dataset is too sparse to accurately estimate a different γ_{vt} for each t. Note that we observed very few instances in our dataset 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.2).

After concluding that the multiplicative model yields a better fit to the data, we decided to re-estimate the parameters using the entire dataset in order to identify $B_1 = e^{\gamma_1}, \ldots, B_{21} = e^{\gamma_{21}}$. In Table 3.5, we present the estimates $\widehat{\gamma}_v$ and boost estimates $\widehat{B}_v = e^{\widehat{\gamma}_v}$ for 7 out of the 21 vehicles present in our data. Out of 21 vehicles, only 5 vehicles are not significant at the 0.05-level, although 3 of them are significant at the 0.1-level. The vehicles that are not listed in Table 3.5 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 nearsignificant, the smallest boost is 1.0722. This confirms that the assumption $B_v \geq 1$ is satisfied by all estimates.

Table 3.5: Estimated promotion vehicle parameters and respective p-values

Promotion Vehicle	$\widehat{B}_v$	$\widehat{\gamma}_v$	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 ($\widehat{B}_3$ and $\widehat{B}_4$) 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 ($\widehat{B}_6$) 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 $\widehat{\alpha}_t$ as follows:

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

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.

3.6.2 Promotion Vehicle Optimization

In order 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 $\widehat{\alpha}_t$ and $\widehat{B}_v$ into the objective function. Additionally, the parameters L_t and C_v are set to their implemented

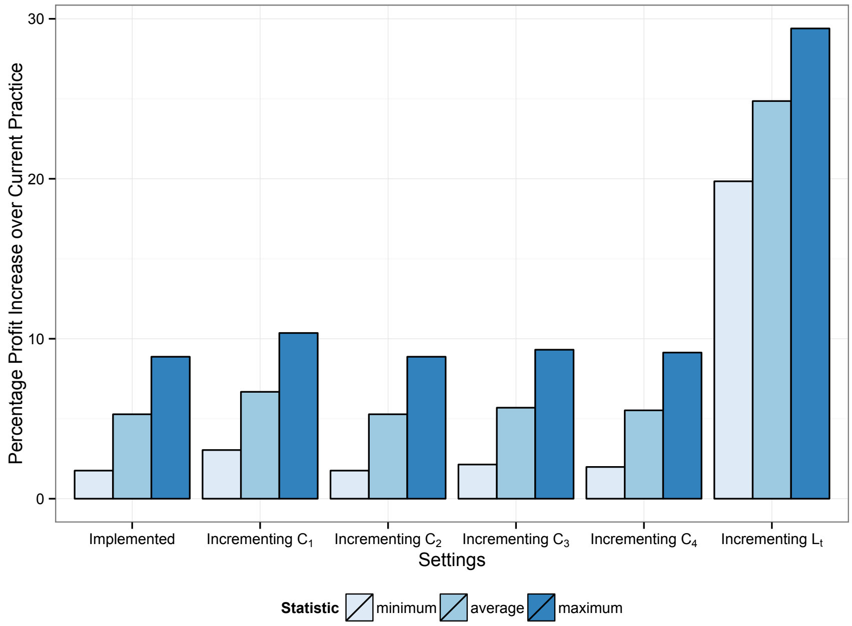


Figure 3.3: Minimum, average, and maximum percentage profit increase for different settings

values in the dataset. 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 3.3.3 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 3.3, 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 to the minimum, average, and maximum percentage increase in profits over the six stores.

[2]Following the suggestion of our industry collaborators, we decided to focus on the 6 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.

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 dataset 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 dataset. Nevertheless, it is worth mentioning that the tests conducted in Section 3.5.2 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 3.3 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. In order 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 order to decide 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 3.3 shows the impact of increasing the capacity L_t by one unit for all time periods. In order 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.

3.7 Conclusions

In this chapter, 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 chapter 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 Appendix B.4), 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 chapter 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 3.4.3 and Appendix B.2.4) is flexible enough to be leveraged into the multi-product 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.

Chapter 4

Pass-through Constrained Vendor Funds for Promotion Planning

4.1 Introduction

Vendor funds are an integral part of the promotion planning process of both suppliers and retailers. Promotion planning can be divided into three different categories (Blattberg and Neslin 1990): trade promotions (supplier-to-retailer, B2B), retailer promotions (retailer-to-customer, B2C), and customer promotions (supplier-to-customer, B2C). Vendor funds are trade deals in which the supplier offers the retailer a short-term discount on a specific product, encouraging the retailer to promote the product in order to increase sales. Clearly, vendor funds deal with trade promotion planning (B2B), and hence indirectly with retailer promotion planning (B2C). Generally, vendor funds are supposed to have beneficial effects to all supply chain partners during important selling periods, for example, around Christmas, Black Friday, or the Olympics. The supplier benefits because the lower retail price attracts new customers to the product, which leads to larger sales and brand recognition. The retailer benefits because the lower retail price temporarily leads to increased sales and the supplier refunds up to the product's regular price. Traditionally, the supplier offers the vendor fund with the goal that the retailer promotes its product, but does not enforce this.

However, there are several issues with vendor funds. Bell and Drèze (2002) and Ailawadi et al. (1999) discuss the large prevalence of forward-buying. Forward-buying is the phenomenon whereby, during vendor fund periods, many retailers stockpile the product for later use and cash in the vendor fund discount. On average, Nijs et al. (2010) estimate that retailers only pass 41%

of the original discount through to the customers. At the same time, estimates of The Nielsen Company (2014) suggest that only one third of trade promotions break even. Inefficiency, low pass-through, and forward-buying counteract the supplier's intentions. In fact, Bell and Drèze (2002) report that only 16% of manufacturers believe vendor deals are profitable, yet The Nielsen Company (2014) reports that $1 trillion circulates in the trade promotion market every year. In the meanwhile, most supply chains still use intuition and heuristics to guide their promotion planning. In this research, we collaborate with Oracle Retail to grasp this opportunity and use operations management ideas to improve the vendor fund process.

In particular, this chapter examines current practice from clients of Oracle Retail. Currently, their retail clients are offered vendor funds with pass-through constraints. Pass-through constraints are part of a vendor fund contract, in which the supplier states that the vendor fund discount is only valid when the retailer passes a minimal fraction of the discount through to the customer during the vendor fund period. To the best of our knowledge, the pass-through constrained vendor fund has not been studied in the literature before. Motivated from practice, we study this type of vendor deal and show that it has great benefits for the supplier and the retailer in that it removes the forward-buying problem and leads to supply chain coordination. The main strength of the pass-through constraint is that it gives more power to the "weaker" partner in the supply chain; the supplier. Without a pass-through constraint, the supplier only has the vendor fund discount as an indirect influence on the retail price and demand. Adding a pass-through constraint to the vendor fund gives the supplier direct influence over the retail price and demand. First, we show that this leads to supply chain coordination, as the supplier wants to maximize the supply chain's profit and share it with the retailer through the vendor fund discount. Second, the retailer might forward-buy and use the vendor fund discount for later sales, but the supplier can adjust the discount to this situation without losing profit.

Including pass-through constraints poses new directions for research in all promotion planning phases, either directly or indirectly (see summary in Figure 4.1). First, we analyze the retailer's vendor fund selection problem: whether to accept or reject vendor fund offers and how to price accordingly during the vendor fund period. In reality, the retailer's vendor fund selection problem is complex. Retailers continuously receive outside offers from other suppliers and only want to select a limited number of vendor deals for a variety of reasons. For example, some suppliers will only offer vendor funds if their competitors are not selected. Retailers may also feel that accepting too many vendor funds constrains their future marketing options. In particular, we model the case where vendor funds constrain promotional pricing. The operational approach can also be extended to the case where vendor funds constrain promotion vehicle scheduling (see Chapter 3 for details on promotion vehicle scheduling). In the end, we devise a practically-good and theoretically-good solution approach for the operational model that helps retailers decide which pass-through constrained vendor funds to select within their limits.

Second, we study the supplier's vendor fund offering problem: what discount to offer and what pass-through to require, accounting for the retailer's vendor fund selection problem. Because the real vendor fund selection problem is too complex to solve analytically, we abstract the problem and preserve its main characteristics. In this work, we model this vendor fund offering problem as a bilevel model and analyze it to uncover the strategic benefits (forward-buying elimination and supply chain coordination) of vendor funds with pass-through.

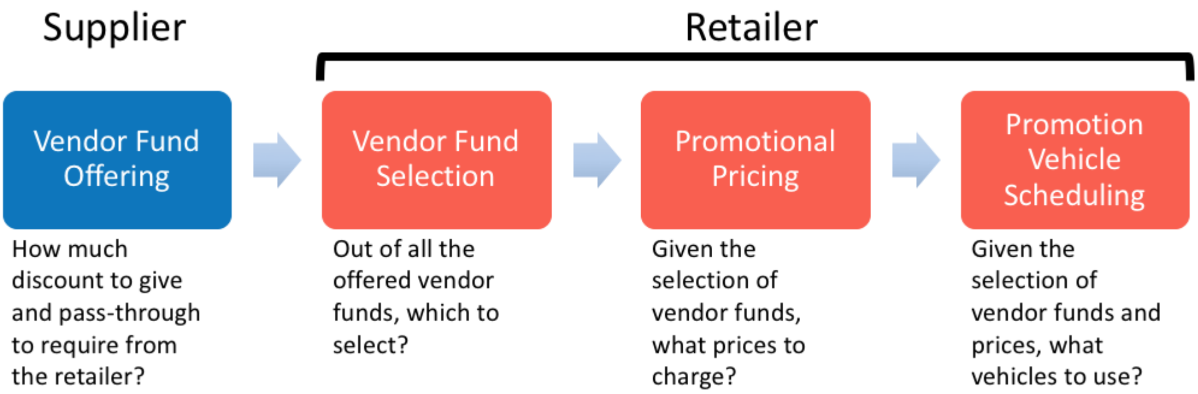


Figure 4.1: Stages of the promotion planning process and their problems associated with vendor funds

4.1.1 Contributions

In this chapter, we introduce and analyze the pass-through constrained vendor fund problem. In Section 4.2, we describe the vendor fund process and the framework of the vendor fund model. Our main contributions are threefold. First, we propose a new effective type of vendor fund (one with a pass-through constraint). We construct a bilevel model to analyze the pass-through constrained vendor fund. Second, we develop an intuitive model that solves the retailer's lower-level problem on vendor fund selection. Practically, we show that our approach makes near-optimal recommendations on which vendor funds to select. Third, we analyze the supplier's bilevel problem on vendor fund offering and selection to show why a pass-through constrained vendor fund is effective. In particular, we show that it resolves the forward-buying problem faced by suppliers while it also coordinates the supply chain on the short-term. In more detail, our contributions are as follows:

1. *Algorithm to solve the lower-level vendor fund problem:* In Section 4.3, we present the lower-level retailer problem on vendor fund selection. Retailers receive many offers from different suppliers and wish to decide which vendor funds to select while simultaneously adapting promotional pricing to the pass-through requirements set by the vendor funds. We model this problem as a quadratic integer optimization problem. Even though this problem is provably hard to solve, we use Lagrangian relaxation and linear optimization to devise a fast and intuitive algorithm with a provable optimality guarantee.

2. *Optimal vendor fund solution to the upper-level vendor fund problem:* In Section 4.4, we analyze the bilevel supplier problem on vendor fund offerings. In the upper level, the supplier's vendor fund offering problem is to determine what discount to give and what pass-through to demand from the retailer. Then in the lower level, the retailer's vendor fund selection problem is to decide whether to accept or reject the vendor fund and what promotional price to set subject to a pass-through constraint if the vendor fund is accepted. Now, instead of explicitly modeling all other vendor deals received by the retailer, we initially presume that the retailer has a deterministic outside option, and later show that our results still hold when this assumption is relaxed to a stochastic outside option.

3. *Elimination of forward-buying problem through pass-through constrained vendor fund:* We show that a supplier offering a vendor fund with pass-through does not have to fear the forward-buying problem. Under a vendor fund without pass-through, the vendor fund discount is the supplier's only instrument to tempt the retailer into a promotion. Including a pass-through constraint gives the supplier another lever to control the retail price, and hence the retailer's demand. In turn, this allows the supplier to use the vendor fund discount to differentiate between forward-buying and non-forward-buying retailers. Although the forward-buying problem has been studied in the marketing literature, creating a vendor fund that allows the supplier to disregard it completely has not been studied.

4. *Supply chain coordination during vendor fund period:* We show that the vendor fund also coordinates the supply chain in the short-term. Supply chain contracts aim to coordinate the supply chain long-term. However, supply chain contracts can become less efficient over

longer periods, because initial demand forecasts misalign more and more as time goes by. Hence, the pass-through constrained vendor fund is a tool to realign the supply chain in the most important periods of the year.

5. *Real-world performance:* Working together with Oracle Retail, we test our Iterative Lagrangian Relaxation (ILR) algorithm and the optimal vendor fund solution on real-world data from a large retailer. We show that the ILR algorithm achieves solutions within 5.25% of the optimal solution, while running within milliseconds. Analyzing the proposed optimal vendor fund, computational experiments show that it performs very well, and specifically we observe that the sensitivity of the expected profits with respect to misspecification of the outside option's distribution is limited. In Section 4.5, we describe the results of our computations.

4.1.2 Literature Review

In the marketing literature on vendor funds and trade deals, Blattberg et al. (1995), Ailawadi et al. (1999), and Bell and Drèze (2002) document the inefficient climate surrounding vendor funds. Blattberg et al. (1995) combines findings in the marketing literature to establish empirical conclusions about sales promotions. One of these empirical conclusions is that the retailer's pass-through is generally less than 100%. In a more recent study, Nijs et al. (2010) estimate that only 41% of the manufacturer's discount finally lands at the customer. Also Bell and Drèze (2002) state that many manufacturers find the vendor fund process to be inefficient. Murry and Heide (1998) and Poddar et al. (2013) observe that strong relationships between manufacturers and retailer employees is important in making sure that retailers comply and lower retail prices during vendor funds. However, Murry and Heide (1998) also suggest that economic incentives are more effective in ensuring compliance from the retailers. Part of the retailer's reluctance to comply is caused by the fact that lowering retail prices in any vendor fund heavily constrains the retailer's planning of sales promotions. An extensive survey of sales promotions can be found in Blattberg and Neslin (1990). On the whole, the marketing literature suggests that the vendor fund process can be improved through the right incentives. In fact, this is the goal of our work. To the best of our knowledge, vendor funds have not been widely studied in operations literature.

Specific to retailer promotion planning, Silva-Risso et al. (1999) acknowledge the difficulty in aligning trade promotions and sales promotions. They devise a decision support tool based on simulated annealing that suppliers and retailers can use together to schedule promotional pricing and promotion vehicles (features, displays). Recently, Cohen et al. (2017b) have developed a non-linear and discrete optimization formulation to plan promotional pricing. They propose an efficient and provably-good algorithm based on linearizing the original problem. Additionally, Ferreira et al. (2016) devise a promotion price recommendation tool that performs well in a field experiment. Their data-driven model captures demand accurately, as it is able to incorporate both parametric and nonparametric models. Caro and Gallien (2012) address the problem of creating optimal markdown promotion strategies in fashion retail. On the other hand, Özer and Zheng (2016) consider a behavioral model to compare markdown promotional strategies to everyday low pricing strategies. This work accounts for both inventory management and dynamic pricing decisions (in the case of markdowns). In contrast, this work focuses on non-markdown promotional planning, as in supermarkets for example. Sales promotion planning is also linked to dynamic pricing reviewed in Talluri and van Ryzin (2004).

There are a few more related works that investigate the operational issues surrounding vendor funds. Kim and Staelin (1999) consider a one-period off-invoice vendor fund model with two competing suppliers and retailers. In this type of vendor fund, suppliers only give a discount and retailers then decide how much of the discount to pass-through to the customer. As in this research, the paper assumes that customer demand is linear in the own price and the competitor's price, and can incorporate other promotion vehicles. Under this assumption, Kim and Staelin (1999) derive the optimal allowances that the suppliers should give as well as the optimal pass-through for the retailers. According to the optimal solution the supplier should offer discounts even though the retailer pockets large parts of it. We consider a different type of vendor fund that is used in practice, one with pass-through. We show it has substantial benefits over the off-invoice vendor fund considered in Kim and Staelin (1999). Furthermore, because we are interested in the forward-buying problem, we consider a two-period model instead of a one-period model.

Drèze and Bell (2003) address the forward-buying problem and propose scan-backs to improve the situation. Scan-backs are vendor funds in which retailers share their sales data and the supplier only reimburses sales that occurred during the vendor fund period. Drèze and Bell (2003) show that there exist scan-backs in which neither the supplier nor the retailer are worse off than in the original case without sharing. Yuan et al. (2013) provide an additional comparison between off-invoice vendor funds and scan-backs. Yuan et al. (2013) first analyze the two types of vendor funds under uncertain demand, after which the theory is tested with market experiments. Both theory and experiments show that off-invoices actually yield higher profits than scan-backs under demand uncertainty. To the best of our knowledge, we approach the problem from a new angle, as the vendor fund with pass-through has not been studied in the literature before. Not only do we find a combination of discount and pass-through that makes all supply chain partners better off, alike scan-backs in Drèze and Bell (2003), we also show that this type of vendor fund leads to short-term supply chain coordination. Another huge benefit under a pass-through constrained vendor fund is that the retailer does not need to share sales data.

Acknowledging the weaknesses of scan-backs, Poddar and Donthu (2013) suggest a virtual forward-buying model. As with our vendor fund with pass-through, the model allows retailers to forward-buy. Retailers can state the amount of units they want to buy at a discount, but instead of getting the discounts all at once, they are accrued over time. Practically speaking, the virtual forward-buying model can be viewed as a method to translate short-term vendor funds into a long-term supply chain contract. Poddar and Donthu (2013) use simulation to analyze the model and find that it reduces costs in the supply chain. In comparison, we prove analytically that the pass-through constrained vendor fund attains the maximum as it coordinates the supply chain.

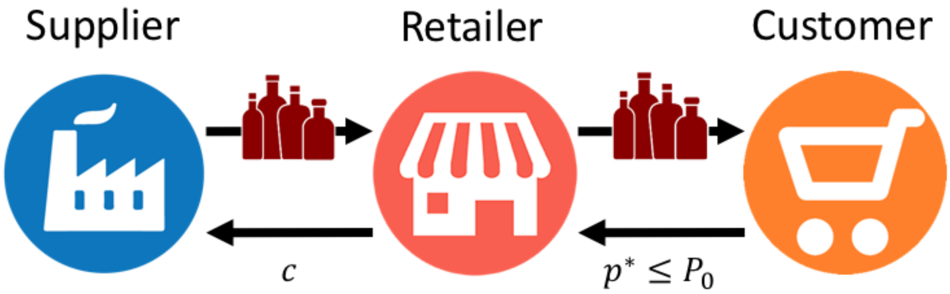
This work also relates to the large body of research on supply chain management. Recent research in this area has dealt with omnichannel revenue and inventory management (Acimovic and Graves 2015, Elmachtoub and Levi 2016, Harsha et al. 2016), sustainability (Gopalakrishnan et al. 2016, Khan et al. 2016, Calmon and Graves 2017), and behavioral operations management (Özer et al. 2011, Nagarajan and Shechter 2013) among others. Specifically, our work is connected to the field of supply chain contracts and coordination, on which Cachon (2003) gives an extensive review. The focus of this research area is on the development of contracts that coordinate suppliers and retailers to cooperate in a manner that is optimal for all supply chain partners. These contracts stipulate wholesale price as well as cost-sharing schemes, examples include buy-back (Pasternack 1985), quantity flexibility (Tsay 1999), channel rebates (Taylor 2002), linear price discount (Bernstein and Federgruen 2005), and revenue-sharing (Cachon and Lariviere 2005) schemes. Over long periods, actual demand can deviate from the initial forecast on which these contracts were built, which causes the contract to become sub-optimal. Promotion planning and pass-through constrained vendor funds realign the supply chain through temporary discounts from the supplier to the retailer and from the retailer to the customer. Another difference is that trade promotions bring about the forward-buying problem. Long-term supply chain contracts are relieved from this problem, because there is no incentive for retailers to forward-buy when the contract is not temporary. Often, the long-term supply chain contracts are set through bargaining and negotiation (Nagarajan and Sošić 2008), but vendor funds are short-term deals without negotiation.

Finally, in modeling the operational vendor fund selection problem we use optimization techniques. We model this as an integer quadratic program and see it is hard to solve. Our solution approach uses ideas from integer programming theory (Schrijver 1998, Wolsey and Nemhauser 1999, Bertsimas and Weismantel 2005) and Lagrangian relaxation theory and applications (Everett 1963, Fisher 1981).

4.2 Problem Description

In what follows, we introduce and formulate the problem of the supplier's offering and the retailer's selection of vendor funds. Suppliers regularly offer vendor funds to retailers with the goal of retailers promoting their products to customers. Figure 4.2 illustrates the vendor fund process. In Figure 4.2a either the supplier is not offering a vendor fund, or the retailer rejected the vendor fund. On the other hand, in Figure 4.2b the supplier offers a vendor fund and the retailer accepts it.

One part of the vendor fund is a discount (δ) on the wholesale unit price (c) of the supplier's product to the retailer. This means that the retailer will temporarily (e.g., for a week) be able to purchase the product at a discount that is $\delta \cdot 100\%$ of the regular retail price (P_0), if the retailer decides to accept the offer. In return, as the other part of the vendor fund, the retailer must meet particular requirements set by the supplier. Prior literature has considered scan-backs where, instead of receiving an up-front discount, the retailer only receives a reimbursement after product units are sold (see for example Drèze and Bell (2003)). For this the retailer has to share point-of-sales data with the supplier, but many retailers are unwilling to share this data. In this chapter, we are motivated by retail practice. From Oracle Retail clients we observe that some vendor funds require the retailer to pass a certain percentage of the discount through to their customers (γ) . Frequently, the supplier demands either that the average retail price over the vendor fund period is lowered by at least $\gamma \cdot 100\%$ of the regular retail price (P_0), or simply



(a) Without vendor fund: retailer purchases unit from supplier at wholesale price c and can determine promotional price p^*

(b) With vendor fund: retailer purchases unit from supplier at a δP_0 discount but has to pass-through discount by lowering promotional price by at least γP_0

Figure 4.2: Description of the vendor fund process

that the price is reduced during the entire vendor fund period. In this chapter, we will focus on the first type of pass-through constraint, but our methods and results are also easily extendable to the case where the retailer is required to discount during the whole vendor fund period.

Before formulating our model, we introduce the notation that will be used for parameters and decision variables in this chapter:

T: Time horizon, indexed by t,

K: Number of prices outside of the regular price, indexed by k,

N: Number of available items/suppliers, indexed by i,

F_i: Set containing indices of available vendor funds for item i, indexed by j,

w_i: Unit wholesale price charged by the supplier of item i,

c_i: Unit production cost for the supplier of item i,

$\mathcal{P}_i$: Price ladder $\mathcal{P} = \{P_{i0} > P_{i1} > \cdots > P_{iK}\}$ containing all unit retail prices for item i, P_{i0} is the regular price,

$d_{it}(P_{ik})$: Demand for item i at time t at price P_{ik},

L_i: Maximum number of price promotions for item i throughout the time horizon,

S_i: Minimum number of separating periods between price promotions of item i,

Q: Maximum number of vendor funds that the retailer can select throughout the time horizon,

τ_j: Set of time periods during which the supplier offers vendor fund j,

θ_j: Set of time periods during which the retailer will forward-buy from vendor fund j,

δ_j: Percentage of the regular price P_0 received by the retailer as a discount for every sale when vendor fund j is selected,

γ_j: Percentage of the regular price P_0 that the retailer has to at least pass through to the customer when vendor fund j is selected,

x_{ikt}: Binary variable indicating whether in time period t price k is used for item i,

z_j: Binary variable indicating whether vendor fund j is selected.

Additionally, if I is an integer, we use $[I]$ to denote the set of the first I integers, i.e., $[I] = \{1, \ldots, I\}$.

In this work, the supplier is interested in offering the right vendor fund, while the retailer is interested in selecting the right vendor funds and promotional prices. The two problems are inextricably linked. Suppliers offer vendor deals to encourage retailers to set the right price and hopefully increase the sales of their product. For their offer to have any effect, they have to ensure that, out of the many options available to the retailer, their vendor funds get selected. All in all, we model the supply chain's vendor fund problem as a bilevel optimization problem. At the upper level, the supplier decides which vendor funds to offer to maximize its profits subject to the lower level where a profit-maximizing retailer selects which vendor funds to select and which prices to set under a set of business rules. At a high level, this bilevel vendor fund problem can be formulated as follows:

$$\max_{0 \le \delta, \gamma \le 1} \pi_S(x^*, z^*, \delta, \gamma) \tag{4.1a}$$

$$\text{s.t. } (x^*, z^*) \in \arg\max_{x, z \in \{0,1\}} \pi_R(x, z, \delta, \gamma), \tag{4.1b}$$

where the following formulation provides more detail on π_S and π_R. In its full formulation, the upper-level problem represents the supplier of item 1 (without loss of generality) maximizing profit over a finite time horizon of T time periods by setting the discounts (δ) and pass-through

(γ) of each of its vendor funds in the set F_1. The upper-level problem is subject to the lower-level problem in which the retailer maximizes profit over T periods by selecting the promotional prices (x) of N items and vendor funds (z) from the sets F_i given by the N different suppliers. This formulation can be presented as follows:

$$\max_{0\leq\delta,\gamma\leq 1} \pi_S(\delta,\gamma) = \sum_{t=1}^{T}\sum_{k=0}^{K}(w_1 - c_1)d_{1t}(P_{1k})\,x^*_{1kt} - \sum_{j\in F_1} z^*_j \sum_{t\in\tau_j\cup\theta_j}\sum_{k=0}^{K}\delta_j P_{10} d_{1t}(P_{1k})\,x^*_{1kt} \tag{4.2a}$$

$$\text{s.t. } (x^*,z^*) \in \arg\max_{x,z\in\{0,1\}} \pi_R(x,z) = \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=0}^{K}(P_{ik} - w_i)\,d_{it}(P_{ik})\,x_{ikt} \tag{4.2b}$$

$$+ \sum_{i=1}^{N}\sum_{j\in F_i} z_j \sum_{t\in\tau_j\cup\theta_j}\sum_{k=0}^{K}\delta_j P_{i0} d_{it}(P_{ik})\,x_{ikt} \tag{4.2c}$$

$$\text{s.t. } \sum_{k=0}^{K} x_{ikt} = 1 \qquad \forall i\in[N], \forall t\in[T] \tag{4.2d}$$

$$\frac{1}{|\tau_j|}\sum_{t\in\tau_j}\sum_{k=0}^{K} P_{ik}x_{ikt} \leq (1-\gamma_j z_j)P_{i0} \qquad \forall i\in[N], j\in F_i \tag{4.2e}$$

$$\sum_{i=1}^{N}\sum_{j\in F_i} z_j \leq Q \tag{4.2f}$$

The lower-level problem (4.1b) is the vendor fund selection problem which addresses how the retailer should select among a multitude of different vendor funds and correspondingly plan their promotional pricing. The detailed formulation given in equations (4.2c-4.2f) is also referred to as (IQP). Our objective is to maximize the retailer's profits, captured by equation (4.2c), by selecting the retail price (x) and which vendor funds to accept (z). The first term of the total profit represents the usual sales profits. It accounts for the profits that are received over T time periods when vendor funds are not involved. The second part represents the bonus from selecting each vendor fund j. Namely, the retailer receives a discount of $\delta_j P_{i0}$ for each unit of item i that is sold in the vendor fund period (any unit sold in time periods τ_j) or if the unit was forward-bought (any unit sold in time periods θ_j).

With regards to the constraint set, constraint (4.2d) is a logical constraint that guarantees the model to select exactly one price from the price ladder in each time period. In our model, constraint (4.2e) captures a pass-through constraint that has been applied in practice; namely if vendor fund j is selected, the average price over the vendor fund period should be less than or equal to a $1-\gamma_j$ fraction of the associated item's regular price P_{i0}. Alternatively, some suppliers might require the retail price to be lowered during the entire vendor fund period. Our model can easily be extended to this case by introducing constraints

$$\sum_{k=0}^{K} P_{ik} x_{ikt} \leq (1 - \gamma_j z_j) P_{i0} \quad \forall i \in [N], \forall j \in F_i, \forall t \in \tau_j. \tag{4.3}$$

In addition, retailers put restrictions on the number of accepted vendor funds. This is modeled in constraint (4.2f) with a maximum of Q vendor funds that can be selected. There could be many reasons: there are suppliers who demand it in the contract, there are operational costs involved with implementing sales promotions, or there is a feeling of becoming too constrained in marketing options.

In this formulation, we do not explicitly mention the business rules that retailers impose on the promotional pricing schedule. However, the solution approaches that we develop in this work can also account for the most common business rules. For example, a retailer may want to limit the number of price promotions to preserve a product's image and avoid customers getting accustomed to promotions. If the retailer wants to ensure that the number of promotions for item i over the whole time horizon is constrained to be at most the limit L_i, then we can use constraint (4.4),

$$\sum_{t=1}^{T} \sum_{k=1}^{K} x_{ikt} \leq L_i \qquad \forall i \in [N]. \tag{4.4}$$

Another example, most retailers separate their promotions over time to mitigate the stockpiling effect. This is the effect whereby customers stockpile promoted products, and thereby purchase less of the product in subsequent time periods. If a retailer wants to ensure that when a promotion occurs in time period t for item i, another one cannot occur until at least S_i separating periods later, then we can use constraint (4.5),

$$\sum_{s=t}^{t+S_i} \sum_{k=1}^{K} x_{iks} \leq 1 \qquad \forall i \in [N], \forall t \in [T - S_i]. \tag{4.5}$$

The upper-level problem (4.1a) is the vendor fund offering problem which describes how the supplier should select the discount and pass-through parameters of the vendor fund. The objective of the supplier in equation (4.2a) is to maximize profit by setting the percentage discount (δ) and pass-through (γ), while taking into account the retailer's decisions on the optimal retail price (x^*) and acceptance ($z^* = 1$) or rejection ($z^* = 0$) of the vendor fund. The first part represents the profit, while the second term accounts for the additional cost due to giving a discount.

In the following two sections, we lay out how we propose to solve the retailer's vendor fund selection problem and the supplier's vendor fund offering problem. First, we address the retailer's lower-level problem. Unfortunately, this problem is hard to solve. In fact, we prove that the problem is NP-hard. Nevertheless, several observations lead us to properties of the formulation that help us to devise a solution approach. This algorithm solves a Lagrangian relaxation of the original problem by iteratively solving for different sets of decision variables. The performance of this algorithm is proven to be bounded and practically near-optimal.

Second, we address the supplier's upper-level problem. Clearly, since the upper-level problem depends on the solution of the lower-level problem and the lower-level problem cannot be solved with a closed-form solution, we have to develop a different approach. We observe that under particular assumptions on the retailer's behavior we can find a solution to the lower-level problem that allows us to solve the upper-level problem in closed-form and obtain the optimal vendor fund. This closed-form solution gives us interesting insights into the effectiveness of the optimal pass-through constrained vendor fund. In particular, we show that it prevents retailers from forward-buying and coordinates the supply chain in the short term.

Finally, even in the case where the retailer actually solves a more complex problem, we propose to use the optimal vendor fund that was constructed under the assumption that the retailer solves a simpler problem. To test its effectiveness, we run extensive computational experiments where we include this optimal vendor fund as one of the vendor funds that the retailer can choose from.

4.3 Solving the Lower-Level Problem

In this section, we analyze how the retailer solves the lower-level problem described by equations (4.2c-4.2f). Due to the model's complexity, we devise an algorithm that approximates the original problem through the use of particular characteristics of the model. Specifically, we show that the problem's hardness originates from the pass-through constraint. Therefore, we use Lagrangian relaxation to move the pass-through constraint into the objective. After establishing properties of this problem, we devise an algorithm that iteratively solves linear programs to obtain an approximate solution. Afterwards, we prove an analytical guarantee on the optimality ratio of this algorithm.

4.3.1 Lower-Level Problem Complexity and Properties

As mentioned previously, we consider the following lower-level problem that was initially presented in equations (4.2c-4.2f), which we will also refer to as the integer quadratic problem formulation:

$$\max_{x,z\in\{0,1\}} \pi_R(x,z) = \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=0}^{K}(P_{ik}-w_i)\,d_{it}\,(P_{ik})\,x_{ikt} + \sum_{i=1}^{N}\sum_{j\in F_i} z_j \sum_{t\in\tau_j\cup\theta_j}\sum_{k=0}^{K}\delta_j P_{i0} d_{it}\,(P_{ik})\,x_{ikt} \tag{4.6a}$$

$$\text{s.t.} \quad \sum_{k=0}^{K} x_{ikt} = 1 \qquad \forall i\in[N], \forall t\in[T] \tag{4.6b}$$

$$\frac{1}{|\tau_j|}\sum_{t\in\tau_j}\sum_{k=0}^{K} P_{ik}x_{ikt} \le (1-\gamma_j z_j)P_{i0} \qquad \forall i\in[N], j\in F_i \tag{4.6c}$$

$$\sum_{i=1}^{N}\sum_{j\in F_i} z_j \le Q \tag{4.6d}$$

The model's decision variables are integer and its quadratic objective is neither concave nor convex. Thus, the problem is hard to solve, both practically and theoretically. In fact, Theorem 4.3.1 shows that the problem is NP-hard through a reduction from the maximal independent set problem.

Theorem 4.3.1. The lower-level problem formulated in problem (4.6) is NP-hard.

Proof. See Appendix C.1.1. □

Still, the problem possesses a nice structure, which we will exploit to construct our approximation method. Particularly, we use Lagrangian relaxation to remove the pass-through constraint and penalize it in the objective. After that, we are left with a more difficult objective that is still not concave or convex, but we observe that the resulting problem can be approximated by iteratively solving two linear problems (LPs) instead of integer problems (IPs). In the next section, we formalize the algorithm and prove Theorem 4.3.3, which gives a theoretical guarantee on the absolute gap between solutions of the Iterative Lagrangian Relaxation (ILR) algorithm and the optimal solution.

To construct our algorithm, we first discuss several observations and problem reformulations. First of all, we recognize that the set of pricing constraints (4.6b) has the consecutive ones property and is therefore integral. This property is also maintained when including the commonly used business rules described by constraints (4.4) and (4.5). This means that solving a linear objective in x_{ikt} over these constraints, where x_{ikt} can take any value between 0 and 1, will yield solutions where all x_{ikt} are either 0 or 1. Similarly, we note that the set of feasible solutions to constraint (4.6d) is also integral, because of the consecutive ones property. Therefore, solving a linear objective in z_j over this constraint, where z_j is allowed to be any value between 0 and 1, will also yield solutions where every z_j is either 0 or 1. In both cases this means that, instead of having to use slower IP solvers, we can use faster LP solvers.

These two observations lead us to conclude that the pass-through constraint (4.6c) is the main source of difficulty. Note that the pass-through constraint is also essential in proving the NP-hardness of problem (4.6). A commonly used method to deal with complicating constraints is Lagrangian relaxation, described in Fisher (1981). We remove the pass-through constraint and penalize it in the objective. For this, we rewrite the pass-through constraint as follows, for all $i \in [N]$, $j \in F_i$,

$$\frac{1}{|\tau_j|}\sum_{t\in\tau_j}\sum_{k=0}^{K} P_{ik}x_{ikt} \le (1-\gamma_j z_j)P_{i0} \Leftrightarrow \sum_{t\in\tau_j}\left[\sum_{k=0}^{K}(P_{ik}x_{ikt}) + \gamma_j P_{i0} z_j - P_{i0}\right] \le 0, \tag{4.7}$$

where the equivalence follows from simple algebraic manipulation. The Lagrangian relaxation formulation is obtained by removing constraint (4.6c) from problem (4.6) and penalizing it in the objective. Particularly, we add a new term to the objective that is the multiplication of the expression in constraint (4.7) and the penalty parameter $\lambda_{ij} \le 0$. Because the model is a maximization problem, it will avoid the case where this added term is negative, and hence avoid the case where constraint (4.7) is positive and not satisfied. This Lagrangian relaxation is given as follows:

$$\max_{x,z\in\{0,1\}} \pi_R(x,z|\lambda) = \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=0}^{K}(P_{ik}-w_i)\,d_{it}(P_{ik})\,x_{ikt} + \sum_{i=1}^{N}\sum_{j\in F_i} z_j \sum_{t\in\tau_j\cup\theta_j}\sum_{k=0}^{K}\delta_j P_{i0} d_{it}(P_{ik})\,x_{ikt}$$
$$+ \sum_{i=1}^{N}\sum_{j\in F_i}\lambda_{ij}\sum_{t\in\tau_j}\left[\sum_{k=0}^{K}(P_{ik}x_{ikt}) + \gamma_j P_{i0} z_j - P_{i0}\right]. \tag{4.8a}$$

$$\text{s.t.} \quad \sum_{k=0}^{K} x_{ikt} = 1 \qquad \forall i\in[N], \forall t\in[T] \tag{4.8b}$$

$$\sum_{i=1}^{N}\sum_{j\in F_i} z_j \le Q \tag{4.8c}$$

Even though we have penalized the constraint, we are still left with quadratic terms $x_{ikt}\cdot z_j$

in the objective. However, as mentioned before, if we fix z and λ, maximizing $\pi_R(x, z|\lambda)$ over constraint (4.8b) can be solved within milliseconds using LP solvers. We note that we can add other constraints as long as they do not break the total unimodularity of the constraint set. For example, we can add the commonly required constraints (4.4) and (4.5) as they retain the total unimodular structure. For any fixed $\lambda \leq 0$ and fixed z, this problem in x is formulated as follows:

$$\max_{0 \leq x \leq 1} \pi_R(x|z, \lambda) = \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{k=0}^{K} (P_{ik} - w_i)\, d_{it}(P_{ik})\, x_{ikt} + \sum_{i=1}^{N} \sum_{j \in F_i} z_j \sum_{t \in \tau_j \cup \theta_j} \sum_{k=0}^{K} \delta_j P_{i0} d_{it}(P_{ik})\, x_{ikt}$$
$$+ \sum_{i=1}^{N} \sum_{j \in F_i} \lambda_{ij} \sum_{t \in \tau_j} \left[\sum_{k=0}^{K} (P_{ik} x_{ikt}) + \gamma_j P_{i0} z_j - P_{i0} \right]. \tag{4.9a}$$

$$\text{s.t.} \quad \sum_{k=0}^{K} x_{ikt} = 1 \qquad \forall i \in [N], \forall t \in [T] \tag{4.9b}$$

Also, if x and λ are fixed, maximizing $\pi_R(x, z|\lambda)$ over constraint (4.8c) can be solved as an LP. For any fixed $\lambda \leq 0$ and fixed x, the problem in z is formulated as follows:

$$\max_{0 \leq z \leq 1} \pi_R(z|x, \lambda) = \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{k=0}^{K} (P_{ik} - w_i)\, d_{it}(P_{ik})\, x_{ikt} + \sum_{i=1}^{N} \sum_{j \in F_i} z_j \sum_{t \in \tau_j \cup \theta_j} \sum_{k=0}^{K} \delta_j P_{i0} d_{it}(P_{ik})\, x_{ikt}$$
$$+ \sum_{i=1}^{N} \sum_{j \in F_i} \lambda_{ij} \sum_{t \in \tau_j} \left[\sum_{k=0}^{K} (P_{ik} x_{ikt}) + \gamma_j P_{i0} z_j - P_{i0} \right]. \tag{4.10a}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} \sum_{j \in F_i} z_j \leq Q \tag{4.10b}$$

4.3.2 Iterative Lagrangian Relaxation (ILR) Algorithm

Altogether, these observations lead us to propose an algorithm that initializes some $\lambda^{(1)}$, iteratively solves an LP for z and one for x, then updates $\lambda^{(1)}$ to $\lambda^{(2)}$ using a subgradient method, and repeats the procedure for a fixed number of iterations. Due to weak duality, minimizing problem (4.8) over λ should lead to a good solution to problem (4.6). More formally, our Iterative Lagrangian Relaxation (ILR) algorithm is described in Algorithm 4.3.2.

Algorithm 4.3.2. (Iterative Lagrangian Relaxation Algorithm)

1. Initialize the penalty parameters $\lambda^{(1)}$, either randomly, or by setting them equal to a constant. Initialize the pricing strategy such that no promotions are used: $x_{i0t}^{(0)} = 1$ and $x_{ikt}^{(0)} = 0$ for all $i \in [N]$, $k \in [K]$, $t \in [T]$.

2. Approximate the optimal solution of the Lagrangian relaxation by searching over λ using a subgradient method. Firstly, for the currently fixed λ we approximate the problem by separately solving problems (4.10) and (4.9) once. Then, we compute the objective's subgradient with respect to λ, after which we take a step in the direction of the subgradient. For $r = 1, \ldots, R$:

   (a) Solve problem (4.10) with $\lambda = \lambda^{(r)}$ and the pricing strategy fixed to $x = x^{(r-1)}$ to find $z^{(r)}$,

   (b) Solve problem (4.9) with $\lambda = \lambda^{(r)}$ and the vendor funds fixed to $z = z^{(r)}$ to find $x^{(r)}$.

   (c) Compute the subgradient of the objective with respect to λ; denoted by $g(x^{(r)}, z^{(r)})$. This is equivalent to computing the error in the pass-through constraint of each item i and vendor fund j: $g_{ij}(x^{(r)}, z^{(r)}) = \sum_{t \in \tau_j} \left[\sum_{k=0}^{K} (P_{ik} x_{ikt}^{(r)}) + \gamma_j P_{i0} z_j^{(r)} - P_{i0} \right]$. Once all errors are non-positive, we know from equation (4.7) that we have found a feasible solution for the pass-through constraint. In this case, if the objective of this new feasible solution is less than the objective of the old best feasible solution, we replace the old solution and objective with the new solution and objective.

   (d) Update the penalty parameters λ by using a subgradient method. For each vendor fund j, we set $\lambda_{ij}^{(r+1)} = \min\{\lambda_{ij}^{(r)} - \eta_r g_{ij}(x^{(r)}, z^{(r)}), 0\}$, where the user can specify the step-size η_r. In our computations we use the following commonly used step-size (Fisher 1981),

$$\eta_r = \frac{\pi_R(x^{(r)}, z^{(r)} | \lambda^{(r)}) - \max_{r' \leq r} \left\{ \pi_R(x^{(r')}, z^{(r')}) \right\}}{\sqrt{r} ||g(x^{(r)}, z^{(r)})||_2^2}.$$

3. If we have found a feasible solution after R steps, we terminate the algorithm with the best feasible solution that was found. If no feasible solution was found, then we create a feasible solution in two steps and terminate afterwards:

   (a) Remove all vendor funds $j \in F$ for which the pass-through constraint was violated (i.e., $g_{ij}(x^{(R)}, z^{(R)}) > 0$) by setting $z_j = 0$.

   (b) Solve problem (4.9) with $\lambda = \lambda^{(R)}$ and the vendor fund selection fixed to the previously constructed z to find the final x.

In Section 4.5.1, we will test the practical performance of the ILR algorithm by running computations. Next, we present Theorem 4.3.3 that gives a bound on the difference between the best feasible solution of the ILR algorithm and the optimal solution to problem (4.6).

Theorem 4.3.3. Let (x^*, z^*) be the optimal solution to problem (4.6) and let (x^{ILR}, z^{ILR}) denote the best feasible solution found by the ILR algorithm for λ^{ILR}. Before running the ILR algorithm,

$$\pi_R(x^*, z^*) - \pi_R(x^{ILR}, z^{ILR}) \leq D(0),$$

and after running the ILR algorithm,

$$\pi_R(x^*, z^*) - \pi_R(x^{ILR}, z^{ILR}) \leq D(\lambda^{ILR}),$$

where $D(\lambda) = \max_{G \subseteq \cup_{i=1}^{N} F_i : |G| \leq Q} \sum_{i=1}^{N} \sum_{j \in F_i \cap G} \left[\sum_{t \in \tau_j \cup \theta_j} \delta_j P_{i0} d_{it}(P_{iK}) + \sum_{t \in \tau_j} \lambda_{ij} \gamma_j P_{i0} \right]$.

Proof. See Appendix C.1.2. □

We assess the computational performance of the ILR algorithm and the tightness of this bound in Section 4.5.1. First, we provide intuition on Theorem 4.3.3 and its proof. In the proof, we construct a guarantee on how well the ILR algorithm approximates problem (4.8), after which we establish a bound on how well the ILR algorithm approximates problem (4.6) generally. Theorem 4.3.3 gives both an a-priori and a-posteriori bound on the algorithm's performance. Before running the algorithm, we do not know what λ will give the optimal solution. To obtain an a-priori bound we maximize the a-posteriori bound over λ. This happens when $\lambda = 0$, because $\lambda \leq 0$ and multiplied by positive parameters in the bound.

At each iteration of the algorithm, we maximize a Lagrangian version of the profit, $\pi_R(x, z|\lambda)$, for a given penalty parameter $\lambda \leq 0$. At iteration r, step 2a starts with the previous set of promotional prices, x_{r-1}, and selects the best vendor funds, z_r. Given these best vendor funds, step 2b chooses the best promotional prices, x_r. Interestingly, in this last step, the Lagrangian profit for the newly selected vendor funds and price promotions, $\pi_R(x_r, z_r|\lambda)$, must be at least the Lagrangian profit for the newly selected vendor funds and the optimal price promotions, $\pi_R(x^*, z_r|\lambda)$. Hence, instead of comparing the gap between (x_r, z_r) and (x^*, z^*), we can compare between (x^*, z_r) and (x^*, z^*). This allows us to ignore any errors we make in pricing. As a consequence, we can focus on the largest possible error we can make by selecting the wrong vendor funds compared to the optimal solution. This maximal difference is given by $D(\lambda)$. We compute $D(\lambda)$ by selecting a set G of Q vendor funds, where each of the funds achieves their maximum possible benefit, $\sum_{t\in\tau_j\cup\theta_j}\delta_j P_{i0}d_{it}(P_{iK})+\sum_{t\in\tau_j}\lambda_{ij}\gamma_j P_{i0}$. Summing the benefits of the Q vendor funds with the largest positive maximum possible benefits gives us the largest possible error we could make in selecting the wrong vendor funds.

4.4 Solving the Upper-Level Problem

In this section, we analyze how the supplier solves the upper-level problem described by equation (4.2a) accounting for the retailer's solution to the lower-level problem described by equations (4.2c-4.2f). In the previous section, we observed that the integer quadratic problem formulation of the lower-level problem turns out to be NP-hard and does not allow a closed-form solution. Nevertheless, under mild assumptions that maintain the key characteristics of vendor funds, we are able to provide an efficient frontier between those vendor funds that the retailer will accept and reject when comparing to an outside option. With this solution to the lower-level problem, we are able to characterize the solution to the upper-level problem in closed-form as well. As a result, we can identify the optimal vendor fund discount and pass-through under additional assumptions on the retailer's behavior. In the following sections, this allows us to gather analytical and computational insights into why the pass-through constrained vendor fund works well.

4.4.1 Upper-Level Problem Tractability and Formulation

For the upper-level problem to be tractable, we make an observation on the amount of information the supplier needs to account for when deciding on the optimal vendor fund parameters. Specifically, we assume that the supplier is deciding on a single vendor fund offer. Then, the only decisions by the retailer that affect the supplier's decisions are whether the retailer accepts the vendor fund and what price the retailer sets during the vendor fund period.

First, the supplier would like the vendor fund to be accepted, as this gives an opportunity for additional profit. In the lower-level problem described by equations (4.2c-4.2f), the driver behind which vendor funds get selected is constraint (4.2f). Only a limited number of vendor funds can be selected, and hence, the retailer will only select the supplier's vendor fund if the offered vendor fund will lead to a larger profit gain for the retailer than an outside option gives (i.e., the retailer's next-best vendor fund). In the following, we let (x^*, z^*) denote the optimal solution when the supplier's vendor fund is selected, and (x^{**}, z^{**}) denotes the optimal solution in which this is not the case. Given this notation, the retailer will select the supplier's vendor fund if $\pi_R(x^*, z^*) \geq \pi_R(x^{**}, z^{**})$, or equivalently,

$$\sum_{t\in\tau_j\cup\theta_j} \sum_{k=0}^{K} (P_{1k} - w_1 + \delta_1 P_{10})\, d_{1t}\,(P_{1k})\, x^*_{1kt} \geq \overline{\pi} \tag{4.11}$$
$$= \sum_{i=2}^{N}\sum_{t=1}^{T}\sum_{k=0}^{K} (P_{ik} - w_i)\, d_{it}\,(P_{ik})\, x^{**}_{ikt} + \sum_{i=2}^{N}\sum_{j\in F_i} z^{**}_j \sum_{t\in\tau_j\cup\theta_j}\sum_{k=0}^{K} \delta_j P_{i0} d_{it}\,(P_{ik})\, x^{**}_{ikt}$$
$$+ \sum_{t=1}^{T}\sum_{k=0}^{K} (P_{1k} - w_1)\, d_{1t}\,(P_{1k})\, x^{**}_{1kt}$$
$$- \sum_{i=2}^{N}\sum_{t=1}^{T}\sum_{k=0}^{K} (P_{ik} - w_i)\, d_{it}\,(P_{ik})\, x^{*}_{ikt} - \sum_{i=2}^{N}\sum_{j\in F_i} z^{*}_j \sum_{t\in\tau_j\cup\theta_j}\sum_{k=0}^{K} \delta_j P_{i0} d_{it}\,(P_{ik})\, x^{*}_{ikt}$$
$$- \sum_{t\notin\tau_j\cup\theta_j}\sum_{k=0}^{K} (P_{1k} - w_1)\, d_{1t}\,(P_{1k})\, x^{*}_{1kt},$$

where $\overline{\pi}$ represents the opportunity profit from choosing the outside option over the supplier's vendor fund. Assuming that the supplier and retailer have full information, as is common in the supply chain contract literature, the supplier can approximately compute $\overline{\pi}$. Nonetheless, we later extend the model to the case where $\overline{\pi}$ is a random variable to the supplier.

Second, the supplier would like the retailer to charge a price that creates the largest additional profit gain. In the lower-level problem described by equations (4.2c-4.2f), the decisions for each item and their respective vendor funds are independent when constraint (4.2f) is eliminated. In fact, the pricing decisions for the supplier's item during the vendor fund period are made independently of the pricing decisions for other items as well as the decisions made outside of the vendor fund period. Hence, if the supplier's vendor fund is selected with certainty, then the supplier can decide on the vendor fund offer accounting only for how the retailer prices during the vendor fund period. This certainty is achieved by adding inequality (4.11) as a constraint to the upper-level problem.

Next to these observations on how the supplier views the retailer's decision, we make two additional assumptions. First, we assume that the vendor fund lasts for a single time period τ, and subsequently the retailer forward-buys θ time periods. This can be a special holiday or important events (Christmas, Black Friday, Olympics), in which many suppliers offer short-term deals. Additionally, we assume that the demand function of the supplier's item is constant over this short period of time and linear in the price. This means that the demand function is given by $d(p) = a - bp$, where p is the price charged by the retailer, and a and b are the parameters of the linear demand curve. Linear demand is often assumed in the marketing literature on vendor deals (Kim and Staelin 1999), and several recent papers in the operations literature (Besbes and Zeevi 2015, Cohen et al. 2018c) show evidence of its efficiency. In the end, these observation and assumptions lead us to the following reformulation of the bilevel vendor fund problem, where the supplier guarantees that the retailer accepts (for notational convenience, we drop the subscript of the supplier's item and vendor fund):

$$\max_{0\leq\delta,\gamma\leq 1} \overline{\pi}_S(\delta,\gamma) = \sum_{t=\tau}^{\tau+\theta}\sum_{k=0}^{K}(w - c - \delta P_0)\,(a - bP_k)\,x^*_{kt} \tag{4.12a}$$

$$\text{s.t.} \sum_{t=\tau}^{\tau+\theta}\sum_{k=0}^{K}(P_k - w + \delta P_0)\,(a - bP_k)\,x^*_{kt} \geq \overline{\pi} \tag{4.12b}$$

$$x^* \in \arg\max_{x\in\{0,1\}} \overline{\pi}_R(x) = \sum_{t=\tau}^{\tau+\theta}\sum_{k=0}^{K}(P_k - w + \delta P_0)\,(a - bP_k)\,x_{kt} \tag{4.12c}$$

$$\text{s.t.} \sum_{k=0}^{K} x_{kt} = 1 \qquad \forall t \in \{\tau, \ldots, \tau+\theta\} \tag{4.12d}$$

$$\sum_{k=0}^{K} P_k x_{k\tau} \leq (1-\gamma)P_0 \tag{4.12e}$$

In this model, the supplier maximizes profits by setting the vendor fund parameters (δ, γ) such that the retailer prefers the vendor fund over the outside option in equation (4.12b) and the retailer optimally sets the retail price subject to a pass-through constraint in equations (4.12c-4.12e). The model is built for the case that the retailer will accept the supplier's vendor fund. However, depending on the outside offer, it is possible that the supplier cannot profit from offering a vendor fund. Only then will the supplier offer a suboptimal vendor fund that the retailer will not accept. We note that, when $\gamma = 0$, the above models can capture the setting where the supplier cannot or does not demand pass-through.

4.4.2 Retailer's Efficient Vendor Fund Frontier

In order to solve the upper-level problem, we first analyze the lower-level problem described by equations (4.12c-4.12e). Only in the vendor fund period is the retailer's price constrained by the pass-through constraint. We note that the pass-through constraint is tight, because otherwise the supplier could have demanded a larger pass-through without affecting the retailer's price optimization. In fact, the supplier could have set the pass-through percentage γ such that the pass-through price $(1-\gamma)P_0$ equals the unconstrained optimizer $\frac{a+bw-b\delta P_0}{2b}$. This means that the retailer sets the price during the vendor fund equal to $(1-\gamma)P_0$ as long as the supplier sets γ such that it is in the price ladder, $(1-\gamma)P_0 \in \mathcal{P}$, and the following condition holds,

$$\frac{a+bw-b\delta P_0}{2b} \geq (1-\gamma)P_0 \Leftrightarrow \gamma \geq 1 - \frac{a+bw-b\delta P_0}{2bP_0}. \tag{4.13}$$

Outside the vendor fund there are θ periods in which the retailer forward-buys. For strategic purposes, most retailers often impose business rules, such as those described in constraints (4.4) and (4.5), that cause the price to return to the original retail price P_0.

Substituting the optimal pricing scheme into constraint (4.12b) we can establish the efficient frontier between those vendor funds that the retailer accepts and those that are not selected over the outside option. Proposition 4.4.1 characterizes this efficient frontier.

Proposition 4.4.1. The supplier's vendor fund (δ, γ) is accepted by the retailer if

$$\delta \geq \begin{cases} \dfrac{w}{P_0} + \dfrac{\overline{\pi} + \overline{\gamma} P_0(a - b(1-\overline{\gamma})P_0)}{P_0(a(\theta+1) - b(\theta+1-\overline{\gamma})P_0)} - 1 & \text{if } 0 \leq \gamma \leq \overline{\gamma} \\ \dfrac{w}{P_0} + \dfrac{\overline{\pi} + \gamma P_0(a - b(1-\gamma)P_0)}{P_0(a(\theta+1) - b(\theta+1-\gamma)P_0)} - 1 & \text{if } \overline{\gamma} \leq \gamma \leq 1 \end{cases},$$

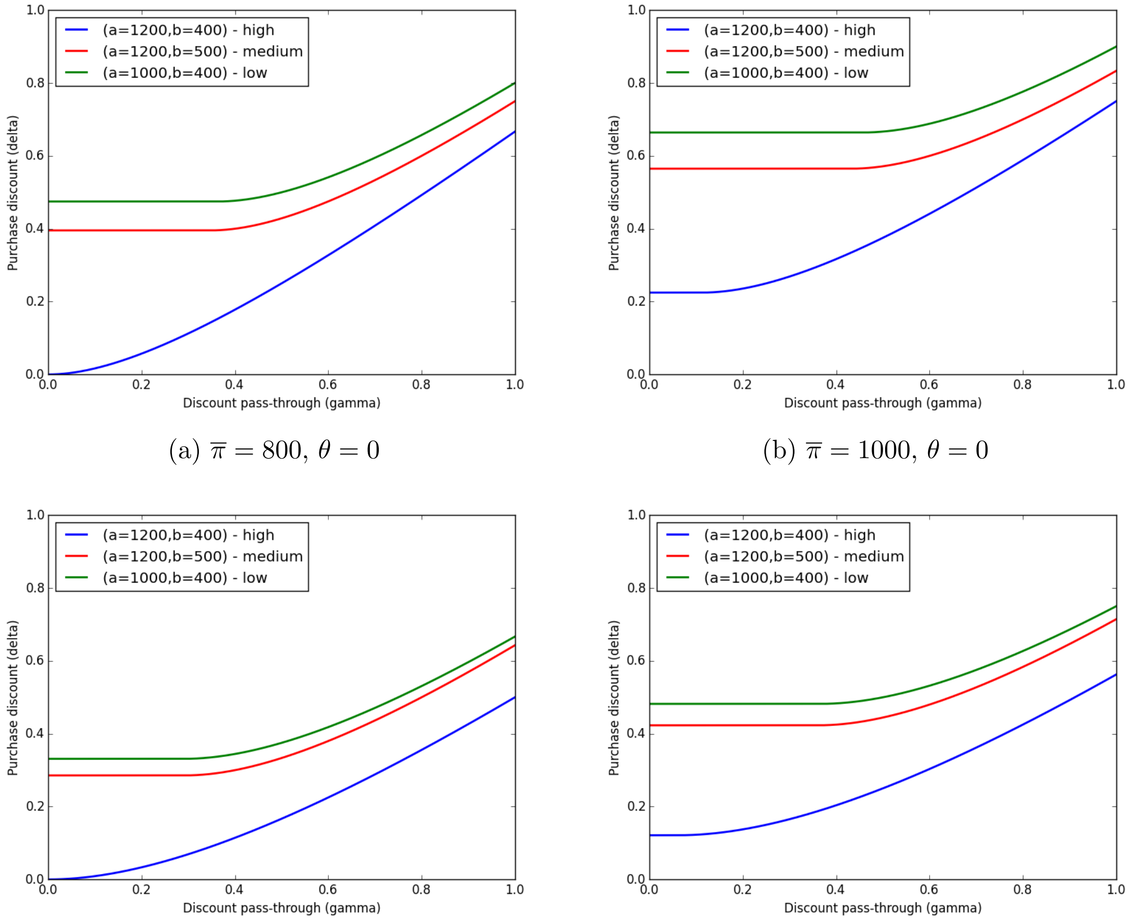
where

$$\overline{\gamma} = \frac{\sqrt{b\overline{\pi} + (a-bP_0)^2(\theta+\theta^2)} - (a-bP_0)(\theta+1)}{bP_0}.$$

Proof. See Appendix C.2.1. □

We illustrate the efficient frontier in Figure 4.3. In these graphs, a vendor fund with its parameters δ and γ above the curve should be accepted and under the curve should be rejected by the retailer. Each of the four graphs represents a different setting of the outside option profit $\overline{\pi}$ or the opportunity to forward-buy by setting $\theta = 0$ or $\theta = 1$. In all cases, the time horizon is set to $T = 2$, the regular price is set to $P_0 = 2$, and the unit cost is set to $w = 1$ (changing the difference between these parameters shifts the curves upwards or downwards). Additionally, in every graph, the 3 curves correspond to different demand curves for the supplier's product, their intercepts and slopes are given in the legend.

First of all, we observe that the larger the demand intercept a and the smaller the demand slope b, the lower the curve lies. A large intercept and small slope represents a case wherein



(a) $\overline{\pi} = 800,\ \theta = 0$

(b) $\overline{\pi} = 1000,\ \theta = 0$

(c) $\overline{\pi} = 800,\ \theta = 1$

(d) $\overline{\pi} = 1000,\ \theta = 1$

Figure 4.3: Efficient frontier between accepted (above the curve) and rejected (below the curve) vendor funds for different levels of the outside profit $\overline{\pi}$ and forward-buying period θ

demand is high regardless of price. In other words, if demand is higher, then the retailer is willing to accept a larger variety of vendor funds. From all graphs, we note that the curve shifts most when the intercept decreases, whereas increasing the slope has a smaller effect.

Figures 4.3a and 4.3b present efficient frontiers under no-forward buying ($\theta = 0$), where the profit $\overline{\pi}$ that can be gained from the outside option is 800 and 1000 respectively. We see that the frontier shifts upward when the outside option's profit increases. As the outside option becomes more lucrative, the supplier has to offer a larger discount or demand a smaller pass-through. Similarly, for the case of a forward-buying retailer ($\theta = 1$), Figures 4.3c and 4.3d illustrate that a more profitable outside option shifts the efficient frontier upward. However, the magnitude of this shift is smaller. The forward-buying retailer gains more out of the supplier's vendor fund, because the purchase discount is spread out over multiple periods. As a consequence, the supplier does not need to increase the purchase discount as much as for a non-forward-buying retailer. This also explains why the efficient frontier is generally lower in the case of forward-buying.

Finally, we notice that the retailer is sometimes willing to accept vendor funds in which more pass-through is demanded than discount is given. This observation seems counterintuitive when you consider that the purchase discount and pass-through parameters δ and γ both represent percentages of the regular price P_0.

4.4.3 Supplier's Optimal Vendor Fund Solution

With the solution to the lower-level problem given by the efficient frontier of Proposition 4.4.1, we are now able to reformulate the upper-level problem and solve for the optimal vendor fund that the supplier should offer. Returning to the original problem, we can now replace constraint (4.12b) with the expressions in Proposition 4.4.1, and replace constraints (4.12c-4.12e) by the discussed pricing policy. Altogether, the upper-level problem becomes the following:

$$\max_{0 \leq \delta, \gamma \leq 1} \ (w - c - \delta P_0)\,(a - b(1-\gamma)P_0) + \theta(w - c - \delta P_0)\,(a - bP_0) \tag{4.14a}$$

$$\text{s.t. } \delta \geq \frac{w}{P_0} + \frac{\overline{\pi} + \gamma P_0(a - b(1-\gamma)P_0)}{P_0(a(\theta+1) - b(\theta+1-\gamma)P_0)} - 1 \tag{4.14b}$$

$$\gamma \geq \frac{\sqrt{b\overline{\pi} + (a - bP_0)^2(\theta + \theta^2)} - (a - bP_0)(\theta + 1)}{bP_0} \tag{4.14c}$$

These problems can be reformulated as one-dimensional problems in γ. In Lemma 4.4.2 we show that the efficient frontier in constraint (4.14b) is always tight, thereby allowing us to write

δ as a function of γ.

Lemma 4.4.2. The constraint corresponding to the efficient frontier (4.14b) holds with equality in the supplier's optimal vendor fund (δ^*, γ^*).

Proof. See Appendix C.2.2. □

Applying this lemma, we rewrite the model in terms of only the pass-through γ:

$$\max_{0\le\gamma\le 1} \ \pi_S(\gamma) = ((1-\gamma)P_0 - c)(a - b(1-\gamma)P_0) + \theta(P_0 - c)(a - bP_0) - \overline{\pi} \tag{4.15a}$$

$$\text{s.t. } \gamma \ge \frac{\sqrt{b\overline{\pi} + (a-bP_0)^2(\theta+\theta^2)} - (a-bP_0)(\theta+1)}{bP_0} \tag{4.15b}$$

Finally, Proposition 4.4.3 outlines the optimal vendor fund and prices when the vendor fund is selected, as well as the corresponding profits.

Proposition 4.4.3. For any instance of the bilevel supplier problem (4.12), the optimal supplier's vendor fund and the optimal retailer's promotion price are given by:

$$(\delta^*, \gamma^*) = \left(\frac{w}{P_0} + \frac{2\overline{\pi} + (P_0 - \frac{a}{2b} - \frac{c}{2})(a-bc)}{P_0(a - bc + 2\theta(a-bP_0))} - 1, 1 - \frac{a+bc}{2bP_0}\right) \text{ and } (1-\gamma^*)P_0 = \frac{a+bc}{2b}.$$

Resulting, the supplier's vendor fund profit is $\overline{\pi}_S^* = \dfrac{(a-bc)^2}{4b} + \theta(P_0 - c)(a - bP_0) - \overline{\pi}$ and the retailer's vendor fund profit is $\overline{\pi}_R^* = \overline{\pi}$.

Proof. See Appendix C.2.3. □

Proposition 4.4.3 shows that both with forward-buying ($\theta > 0$) and without forward-buying ($\theta = 0$), the supplier will require the same pass-through and the retailer will hence charge the same promotion price. Only the optimal discount differs depending on the degree of forward-buying. The discount under no forward-buying is larger than the discount under forward-buying. When the retailer buys forward, the retailer is effectively getting an extra discount on the forward-bought units. If the supplier offers a vendor fund acceptable to a non-forward-buying retailer, then a forward-buying retailer will surely accept the offer and make extra profit. Hence, the supplier can improve profits by slightly reducing the discount (i.e., increase the wholesale promotion price), because the forward-buying retailer will still accept.

Next, in Proposition 4.4.4 we show that the forward-buying problem is eliminated by the pass-through constraint.

Proposition 4.4.4. Pass-through constrained vendor funds eliminate the forward-buying problem.

Proof. See Appendix C.2.4. □

When imposing a pass-through constraint, Proposition 4.4.3 shows that the supplier can obtain the exact same profit whether the retailer is forward-buying or not. However, in this proposition we also check that the forward-buying problem is not resolved by a vendor fund without a pass-through constraint. To model the case of no pass-through constraint, we set $\gamma = 0$ in the original problem (4.12). Solving this model without pass-through shows us that the optimal supplier's revenue differs between the two cases. Therefore, we observe that suppliers need to impose the pass-through constraint to not be impacted by forward-buying behavior of the retailer.

In what follows, Proposition 4.4.5 shows that vendor funds can be used to coordinate supply chains on the short-term.

Proposition 4.4.5. Pass-through constrained vendor funds coordinate the supply chain during the vendor fund period.

Proof. See Appendix C.2.5. □

We achieve supply chain coordination when the supplier and retailer together earn the maximum profit that a monopolist could earn in a centralized supply chain. In the proposition, we show that the optimal vendor fund profit of the centralized supply chain equals $\overline{\pi}^*_{SC} = \frac{(a-bc)^2}{4b} + \theta(P_0 - c)(a - bP_0)$, which is the sum of the supplier's optimal vendor fund profit $\overline{\pi}^*_S$ and the retailer's optimal vendor fund profit $\overline{\pi}^*_R$. Hence, the vendor fund coordinates the supply chain for the short-term (i.e., during the vendor fund).

The major insight from these propositions is that the pass-through constraint is crucial. The supplier wants to increase profit by increasing sales to the retailer, which can be done by increasing the retailer's sales. Without the pass-through constraint, the supplier needs to use the vendor fund discount to achieve two goals: minimize forward-buying and maximize retail sales. Offering a larger discount encourages the retailer to lower the retail price, which leads to larger sales and consequently profits for the supplier. Though, offering larger discounts also encourages more forward-buying from the retailer. The pass-through constraint gives the supplier control over the retail price, so as to increase sales and profits. At the same time, the discount can then be used solely to split profits with the retailer. The supplier can adjust the discount to whether the retailer is forward-buying or not. Hence, the supplier can get the same profits in either case.

4.4.4 Stochastic Outside Option

In reality, the supplier does not know which competing vendor funds are available to the retailer or how the retailer selects among these. Instead, the supplier can view the profit of the outside option as a random variable. The supplier assumes that $\overline{\pi} \sim F(\cdot)$, where $F(\cdot)$ is a cumulative distribution function, while the retailer can know the actual value of the outside option.

Due to a lack of information, the supplier maximizes its expected profits, taking the expectation with respect to the retailer preferring the vendor fund over the outside option. Under a deterministic outside option, the supplier can always ensure that the retailer accepts the vendor fund offer. In the stochastic case, the outside option is uncertain, so there is a probability that the retailer will not accept. Hence, the expected profit consists of two terms, one term being the profit under a vendor fund multiplied by the probability of the vendor fund being accepted, the other being the regular profit without a vendor fund multiplied by the probability of the vendor fund being rejected. Altogether, knowing the pricing policy of the retailer, the supplier's expected vendor fund profit in the stochastic problem becomes the following:

$$\begin{aligned}
\max_{0\leq\delta,\gamma\leq 1} \mathbb{E}\left[\pi_S(\delta,\gamma)\right] &= \Big((w-c-\delta P_0)\left(a-b(1-\gamma)P_0\right)+\theta(w-c-\delta P_0)\left(a-bP_0\right)\Big)\cdot\mathbb{P}\Big(\overline{\pi}\leq\overline{\pi}_R\Big) \\
&+\Big((w-c)\left(a-b\frac{a+bw}{2b}\right)+\theta(w-c)\left(a-bP_0\right)\Big)\cdot\mathbb{P}\Big(\overline{\pi}\leq\overline{\pi}_R\Big) \\
&=\Big((w-c-\delta P_0)\left(a-b(1-\gamma)P_0\right)-\theta\delta P_0\left(a-bP_0\right)-\frac{1}{2}(w-c)\left(a-bw\right)\Big)\cdot F\Big(\overline{\pi}_R\Big) \\
&+\frac{1}{2}(w-c)\left(a-bw\right)+\theta(w-c)\left(a-bP_0\right)
\end{aligned} \tag{4.16}$$

where $\overline{\pi}_R = ((1-\gamma)P_0 - w + \delta P_0)\left(a - b(1-\gamma)P_0\right) + \theta\left(P_0 - w + \delta P_0\right)\left(a - bP_0\right)$ is the retailer's optimal vendor fund profit. In the original problem (4.12), the supplier knew the outside option of the retailer and maximized the deterministic profit while enticing the retailer to accept the vendor fund. Here, the supplier does not know the outside option of the retailer and maximizes the expected profit, in which the uncertainty comes from the retailer's vendor fund selection described by the probabilities. The first term of the expected profit function (4.16) is the profit when the vendor fund is accepted multiplied by the probability of acceptance. The second term is the profit when the vendor fund is rejected multiplied by the probability of rejection. When the vendor fund is rejected, the supplier's profit is determined by the retailer's optimal promotional price under no vendor fund.

In Lemma 4.4.6 we characterize the stationary points of the model. Whether these stationary points are the global maxima depends on the specific distribution that the outside option takes.

Lemma 4.4.6. For any instance of the stochastic bilevel supplier problem (4.16), its stationary point $(\widetilde{\delta}, \widetilde{\gamma})$ is given by:

$$\widetilde{\delta} \text{ s.t. } (w - c - \widetilde{\delta} P_0)(a(\theta+1) - b(\theta+1-\widetilde{\gamma})P_0) - \frac{1}{2}(w-c)(a(2\theta+1) - b(2\theta P_0 + w)) = \frac{F\left(\overline{\pi}_R\right)}{f\left(\overline{\pi}_R\right)}$$

and $\widetilde{\gamma} = 1 - \dfrac{a}{2bP_0}$.

Proof. See Appendix C.2.6. □

Note that in these stationary points, the pass-through (and therefore also the promotion price) remains the same after changing to a stochastic outside option. On the other hand, the optimal discount is dependent on the distribution of the outside option. For some distributions, the optimal discount can be found in closed form. In addition, whether these stationary points are global maxima depends on the distribution. When the profit of the outside option is uniformly distributed, $\overline{\pi} \sim U[\overline{\pi}_L, \overline{\pi}_U]$, the vendor funds are optimal. Often, the supplier has limited information on the outside option, such as its minimum $\overline{\pi}_L$ or maximum $\overline{\pi}_U$, in which case it is natural to assume a uniform distribution. In Proposition 4.4.7 we summarize the optimal vendor fund and promotion price for the uniform model, as well as the optimal profit for the supplier and retailer.

Proposition 4.4.7. For any instance of the uniform bilevel supplier problem (4.16), the optimal supplier's vendor fund and the optimal retailer's promotion price are given by:

$$(\delta^*, \gamma^*) = \left(\frac{w}{2P_0} + \frac{2\overline{\pi}_L + (2P_0 - \frac{a}{2b} - \frac{c}{2})(a-bc) + b\left(w^2 - c^2\right) + 2\theta P_0(a - bP_0)}{2P_0\left(a - bc + 2\theta(a - bP_0)\right)} - 1, 1 - \frac{a+bc}{2bP_0} \right)$$

and $(1-\gamma^*)P_0 = \dfrac{a+bc}{2b}$.

In both cases, when the retailer accepts the vendor fund (if $\overline{\pi} \leq \overline{\pi}_R^*$), the supplier's profit is

$$\pi_S^* = \frac{1}{2}\left(\frac{(a-bc)^2}{4b} + \theta(P_0 - c)(a - bP_0) - \overline{\pi}_L + \frac{1}{2}w(a(2\theta+1) - b(2P_0 + w + c)) + c(bc - \theta(a - bP_0)) \right),$$

and the retailer's profit is

$$\overline{\pi}_R^* = \frac{1}{2}\left(\frac{(a-bc)^2}{4b} + \theta(P_0 - c)(a - bP_0) + \overline{\pi}_L - \frac{1}{2}w(a(2\theta+1) - b(2P_0 + w + c)) - c(bc - \theta(a - bP_0))\right).$$

Proof. See Appendix C.2.7. □

We observe that the optimal pass-through and promotion price under a uniform outside option are the same as under a deterministic outside option. As in the deterministic case, the optimal discount is still smaller when the retailer forward-buys. However, the optimal discount decreases when the outside option's lower bound decreases, even when the outside option's mean is kept constant. From this we infer that the optimal discount decreases as the variance increases. Thus, the supplier offers a more conservative vendor fund when the uncertainty increases.

Similar to under a deterministic outside option, the uniform vendor fund allows the supplier to forget about forward-buying, extending Proposition 4.4.4, and coordinates the supply chain, extending Proposition 4.4.5. The forward-buying problem is eliminated by the pass-through constraint, because the supplier's profit is independent of whether the retailer is forward-buying or not. Additionally, adding up the optimal supplier's and retailer's profit shows that the supply chain is coordinated when the retailer accepts the vendor fund. If the retailer rejects the vendor fund, then the supply chain is not coordinated, because the retailer can obtain more profit from the outside option.

4.5 Computational Experiments

4.5.1 Performance of the ILR Algorithm

After describing the algorithm and bound, we now run computational experiments to test the performance of the algorithm. We compare the ILR algorithm (Section 4.3.2) to the optimal solution, which is computed using the formulation in problem (4.6). Through Oracle Retail, we have access to data from a large supermarket client. We use this dataset to guide the parameter settings of our computational experiments.

We assume different settings with $T = 13, 26$ time periods (a quarter year and half year), $K = 5, 10, 15, 20$ different promotional prices, and $J = 5, 10, 15$ vendor funds. It is common for supermarkets to make their decisions on a rolling horizon of a quarter year or half year.

Also, products have at most 20 promotional prices and 15 vendor deals is on the larger size for a period of 26 weeks. These settings are motivated by practice.

In our computations, we focus on two products in the coffee category of the retailer: one house brand and one premium brand. For these products, we normalize the regular price to $P_0 = 1$ and the cost to $c_t = 0$. The promotional prices are drawn from the dataset and normalized to this range. The demand functions $d_t(p)$ are assumed to be linear in the price and constant over time, i.e., $d_t(p) = a - bp$. For the house brand product $a = 958$ and $b = 909$, and for the premium brand product $a = 938$ and $b = 888$. Each vendor fund j is randomly generated by drawing δ_j and γ_j uniformly on $[0, 0.5]$, and drawing τ_j uniformly at random on $\{\{1,2\},\ldots,\{T-1,T\}\}$. Furthermore, Q is drawn uniformly at random from $\{1,\ldots,J\}$, L is uniform on $\{1,\ldots,T\}$, and S is uniform on $\{1,\ldots,T/L\}$. We initialize the ILR algorithm with $\lambda_j^{(1)} = -1$ and $R = 100$ iterations. The computations were run using a standard desktop computer with an Intel Core i5-4690K @ 3.5GHz CPU and 8 GB RAM. The formulation in problem (4.6) and the ILR algorithm were programmed using Julia/JuMP (Dunning et al. 2017) and solved with the use of Gurobi 6.5.1.

In Table 4.1 we present the average performance of the ILR algorithm and the bound on this performance given in Theorem 4.3.3. The performance ratio is defined by the best-found profit of the ILR algorithm divided by the optimal profit found through problem (4.6). For each setting of J, K, and T, the table shows the average performance ratios over 1000 random instances for both products.

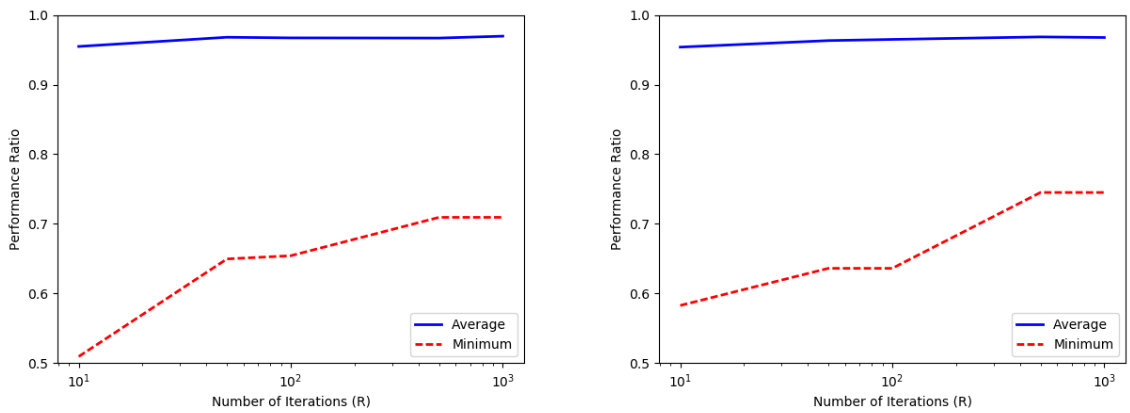
Table 4.1: Performance of the ILR algorithm and bound from Theorem 4.3.3

			$T = 13$				$T = 26$			
			K				K			
			5	10	15	20	5	10	15	20
		5	0.9847	0.9848	0.9804	0.9793	0.9920	0.9902	0.9897	0.9884
ILR	J	10	0.9716	0.9671	0.9636	0.9623	0.9838	0.9798	0.9776	0.9767
		15	0.9555	0.9516	0.9475	0.9485	0.9732	0.9690	0.9689	0.9649
		5	0.8820	0.8600	0.8413	0.8345	0.9349	0.9141	0.9070	0.8996
Bound	J	10	0.7948	0.7498	0.7314	0.7167	0.8746	0.8394	0.8234	0.8150
		15	0.7221	0.6742	0.6537	0.6467	0.8160	0.7751	0.7606	0.7505

First, we observe that the ILR algorithm is near-optimal, i.e., within 5.25% of the optimal in the worst case. Especially in the more realistic case where the number of vendor funds (J) is smaller, the algorithm performs consistently within 2% of the optimal. A change in the number of promotional prices (K) does not seem to have a considerable effect on the algorithm's performance. Counterintuitively, the algorithm actually performs better as the number of time periods (T) increases. In general, retailers are satisfied with these small optimality losses in return for quick and interpretable solutions.

With regards to the lower bound on optimality, we see that the bound deteriorates more rapidly when the number of vendor funds (J) increases than when the number of promotional prices (K) increases. As discussed in Section 4.3.2, the bound looks at the worst error that can be made in choosing the wrong vendor funds. Clearly, when the number of vendor funds grows, this error is expected to worsen. There is a gap between the actual performance of the algorithm and the bound on this performance, but still the bound gives a good indication that the ILR algorithm performs well, on average above 65% for just one quarter and above 75% for two quarters.

Finally, before running the ILR algorithm, the user has to specify the number of iterations to run (R). In Figure 4.4 we analyze the sensitivity of the performance ratio to this parameter. Figure 4.4a shows the average performance and bound in the setting where $J = 10$, $K = 10$, and $T = 13$, while Figure 4.4b shows the results in the largest setting $J = 15$, $K = 20$, and $T = 26$.



(a) $J = 10$, $K = 10$, $T = 13$

(b) $J = 15$, $K = 20$, $T = 26$

Figure 4.4: Average and worst-case performance of the ILR algorithm when varying the number of iterations (R) of the ILR algorithm

In both Figures 4.4a and 4.4b we observe that the average performance of the ILR algorithm is near-optimal for any number of iterations (R). We only see a slight improvement between 10 and 50 iterations. It is more interesting that the worst-case performance ratio improves significantly as we increase the number of iterations (R). Initially, for 10 iterations the worst solution is within between 40-50% of optimal, but for 1000 iterations the worst solution is at least within 30% of the optimal solution. Regardless, the average performance of the algorithm is near-optimal and it solves within milliseconds.

4.5.2 Performance of the Optimal Vendor Fund

In what follows, we run computations to analyze the sensitivity of the optimal expected profit with respect to misspecifying the distribution. First, using Proposition 4.4.7 we construct optimal vendor funds under uniform outside options with the same mean ($\mathbb{E}[\overline{\pi}] = \mu_{\overline{\pi}} = 150$) but different spread around the mean (± 0, ± 50, ± 100, ± 150). However, we assume that the outside option is actually beta or gamma distributed, and we replace the expression of the uniform cumulative distribution function in the expected profit function (4.16) by that of the beta or gamma distribution. With this, we compute the expected profit of the uniform vendor funds when the true distribution of the outside option is beta or gamma. Second, we compute the expected profit of the truly optimal vendor funds under these distributions numerically.

In Table 4.2 we present the resulting performance under misspecification of the outside option's distribution. Realistically, retailers are forward-buying, so we focus on the performance of vendor funds under forward-buying. The performance is measured by the expected profit of the row's uniform vendor fund divided by the expected profit of the column's optimal vendor fund. For every row, the beta distribution is stretched to the same interval as the uniform outside option. The other model parameters are estimated from Oracle Retail client data: $P_0 = 1$, $w = 0.9$, $c = 0$, $\theta = 1$, $a = 958$, and $b = 909$.

Table 4.2: Performance of uniform vendor funds relative to optimal vendor funds when the outside option has a beta or gamma distribution

	Beta			Gamma		
Vendor fund under	$(0.5, 0.5)$	$(2, 2)$	$(1, \mu_{\overline{\pi}})$	$(2, \mu_{\overline{\pi}}/2)$	$(\mu_{\overline{\pi}}/2, 2)$	$(\mu_{\overline{\pi}}, 1)$
Deterministic $\overline{\pi}$ $(= \mu_{\overline{\pi}})$	0.8677	0.8677	0.7844	0.8435	0.9656	0.9636
Uniform $\overline{\pi}$ $(\mu_{\overline{\pi}} - 50, \mu_{\overline{\pi}} + 50)$	0.9913	0.9919	0.8783	0.9333	0.9692	0.9376
Uniform $\overline{\pi}$ $(\mu_{\overline{\pi}} - 100, \mu_{\overline{\pi}} + 100)$	0.9871	0.9873	0.9493	0.9871	0.8541	0.8273
Uniform $\overline{\pi}$ $(\mu_{\overline{\pi}} - 150, \mu_{\overline{\pi}} + 150)$	0.9836	0.9841	0.9914	0.9987	0.8362	0.8250

From Table 4.2 we observe that the misspecification loss is at most 22%. This happens when the proposed vendor fund is built with a deterministic outside option in mind but the true distribution is gamma with parameters $(1, \mu_{\overline{\pi}})$. This is the same as the exponential distribution with parameter $\mu_{\overline{\pi}}$. In this case we can expect the error to be larger, because the exponential distribution has a large variance and the corresponding deterministic vendor fund ignores this. Overall, the performance ratios increase significantly by assuming a small spread, but become worse as the spread is increased too much. The results corresponding to $\overline{\pi}$ being uniform on $[\mu_{\overline{\pi}} - 50, \mu_{\overline{\pi}} + 50]$ show that the loss from misspecifying the distribution as uniform with a small spread is limited. Overall, these results suggest that a vendor fund offering supplier can assume the outside option to be uniform without incurring significant losses.

4.6 Conclusions

This chapter discusses the impact of vendor funds on the promotion planning process of suppliers and retailers in the presence of pass-through constraints. Vendor funds are fundamental to the retail industry. The trade promotion market is large, even though suppliers and retailers currently do not deem it a success. In this chapter, we propose the pass-through constrained vendor fund that has been used in practice, but not yet studied in the academic literature. To assess its strengths, we model promotion planning with vendor funds as a bilevel optimization model. We start by solving the real-world lower-level problem of the retailer. Then, as this lower-level problem is complex, we abstract out its main characteristics to be able to solve and analyze the entire bilevel problem of the supplier.

At first, we study the lower-level model and give retailers a tool that recommends the optimal selection of vendor funds. Retailers are often faced with many different vendor funds from different suppliers, and a better selection of these vendor funds can significantly improve profits. Even though the model is a hard integer quadratic optimization model, we develop an algorithm that quickly finds near-optimal solutions, both in theory and practice.

Afterwards, we analyze the upper-level model to examine how suppliers should set their discounts and pass-through constraints, while accounting for a retailer who can decide to reject the vendor fund, but in case of acceptance has to set the promotional price in accordance with the pass-through constraint. We show that the pass-through constrained vendor fund eliminates the forward-buying problem; the effect by which a retailer stockpiles the supplier's product when a vendor fund happens. Additionally, we show that vendor funds are effective in coordinating the supply chain on the short-term. These results extend to a setting where the outside option available to the retailer is stochastic.

Chapter 5

The Impact of Data Analytics on Promotion Planning

5.1 Introduction

This chapter describes the process of applying a promotion optimization model in order to establish its value for retailers. In past work, it was shown that optimizing promotions has the potential to significantly impact the profit of retailers on the order of 3-9% (see, e.g., Cohen et al. 2018a, 2017b). In this chapter, our goal is to establish the impact of promotions in practice by testing our promotion optimization model with an Oracle Retail Global Business Unit (RGBU) client. Additionally, the description of this process can act as a case study on how this model can be applied to an entire range of retailers, from mid-market retailers, with an annual revenue below \$1 billion, to tier-1 retailers, with an annual revenue exceeding \$5 billion, in all verticals (e.g., electronics, fashion, groceries, hard-lines).

In recent years, based on our experience with Oracle Retail clients we have observed that retailers, particularly supermarket chains, have become more and more interested in ways to efficiently manage sales promotions. Promotions are a prevailing tool in the grocery industry, as they can serve several purposes such as increasing store traffic, introducing new items, boosting the sales of a particular brand, capturing additional customer segments, and retaliating competitors who are promoting their own items. The main tool is the classical price promotion, which is a temporary reduction in the retail price. According to a study by A.C. Nielsen in 2009, 42.8% of grocery sales in the US were made during sales promotions. It is well known that the supermarket industry is usually characterized by low profit margins (of the order of 2%

or less) and therefore, improving the promotion planning process can have a significant impact on the bottom line.

The key research questions in this chapter are: (1) What are the practical steps required when implementing an optimization promotion tool at a retailer? (2) How much money does a retailer leave on the table by using the currently implemented prices relative to using a more scientific and data-driven approach to determine the optimal promotion prices? Most of the promotion planning tools currently available in the industry are based on running different "what-if" scenarios via simulation. These techniques do not really optimize promotions, i.e., determining when to offer the promotions, what price discounts to use, and for which products. Our goal is to apply an optimization model that can maximize retail profits by setting the right promotions at the right time. Importantly, the model captures consumer behavior through the demand function that is calibrated with real data. Additionally, it is flexible enough to account for the business rules that are imposed by the retailer.

In a recent paper (Cohen et al. 2017b), the authors mined two years of sales and promotions data from a grocery retailer. Based on the data, they were able to develop various demand models that capture price and promotion effects as well as general consumer behavior. In addition, these models provided a high forecasting accuracy out-of-sample. Next, the authors developed an optimization model that maximizes the expected profits characterized by these demand models. Since the optimization formulation was a non-linear integer program, the authors proposed an approximation solution approach based on linear programming. This method is easy to implement, very scalable and has provided a near-optimal solution for the tested real-world instances. In addition, tests on historical data suggested that the promotion optimization tool increases profits by 3-10% just by optimizing the promotion schedule. Note that if implementing the promotions recommended by this model does not require additional fixed costs (this seems reasonable as we only vary prices), then even a 3% increase in profits for a retailer with annual profits of $100 million translates into a $3 million profit increase. As we previously discussed, profit margins in this industry are small and therefore, a 3% profit improvement is significant.

Encouraged by the results in Cohen et al. (2017b), we approached a tier-1 client of Oracle Retail to set up a controlled pilot in order to assess the impact of such a promotion optimization tool. In this work, we describe the process of designing a pilot, and in particular, the various challenges when applying this data-driven tool to a real-world setting. We present the whole process: from data collection and analysis to our promotion recommendations for the pilot retailer. We conclude the chapter by discussing the potential broader impact on Oracle Retail, their clients, and beyond.

5.1.1 Contributions

This chapter has several practical contributions to the field of retail operations. First, we conduct a beginning-to-end process in collaboration with a retailer: from the data collection to the future promotion recommendations for the upcoming selling season. We discuss in detail the challenges we faced, and how we tackled them. The first part provides a detailed data analysis of the historical data from 2012, 2013, and 2014. This task includes the product selection (Section 5.3.1) as well as the store clustering step (Section 5.3.2). We then use the sales data from 2012 and 2013 to estimate the different demand models for the three treatment items we chose (Section 5.4.1). Next, we test and validate our demand prediction out-of-sample by using the sales data from 2014. Our demand models yield a high out-of-sample prediction accuracy: adjusted R^2 of 0.89 for the category of interest, and low MdAPE (Median Absolute Percentage Error). The following step is to solve the promotion optimization problem in order to generate the future promotion decisions (Section 5.4.2). Comparing the promotions suggested by our model to the promotions implemented by the retailer in 2014 shows a profit increase of 9.94%, while maintaining the same level of revenue (Section 5.4.3). This convinced the retailer to set up a field experiment for 16 weeks in 2015. As the pilot retailer said: "After several discussions and the data analysis that the team of researchers performed, we realized that a lot of money is left on the table." Unfortunately, we could not follow-up with the pilot retailer regarding the outcome of our promotion recommendations as they went through a major acquisition. Nonetheless, we conclude the chapter by showing our concrete promotion recommendations, and discussing the broader impact of the promotion planning tool developed in this line of research (Section 5.5). We believe that the methodology and tool presented in this chapter are generalizable, and thus not restricted to the particular retailer considered in this work.

5.1.2 Literature Review

This chapter is closely related to the retail operations literature, a topic that received significant attention by both academics and practitioners. In recent years, many companies aim to develop efficient data-driven decision making tools. A typical retailer needs to decide a large number of operational decisions at any given point in time, such as 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). The same claim can be made for both assortment planning (see the survey in Kök et al. (2008) and the references therein), and pricing decisions (see, e.g., Phillips 2005, Cohen et al. 2017b).

More precisely, this work is related to the stream of literature on sales promotions. We refer the reader to Blattberg and Neslin (1990), and to the references therein for a comprehensive review. The topic of sales promotions was extensively studied in the marketing community, but has mainly focused on modeling and estimating dynamic sales models which can be used to derive managerial insights (see, e.g., Cooper et al. 1999, Foekens et al. 1998). For example, Foekens et al. (1998) study econometric models based on scanner data to examine the dynamic effects of sales promotions. Finally, note that several prescriptive works in the marketing community study the impact of retail coupons in the context of sales promotions (see, for example Heilman et al. 2002). Optimizing 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).

This work deals with optimizing promotion planning in a retail environment. The recent paper in Cohen et al. (2017b) introduces and studies an optimization formulation with a demand model estimated from data as input. The authors propose an efficient algorithm to solve the price promotion optimization problem, based on discretely linearizing the objective function and solving a linear program. The authors then show that their approximation yields a near-optimal solution (in the vast majority of practical instances) that runs in milliseconds, and can easily be implemented by retailers. The paper in Cohen et al. (2017b) considers the case of a single item, and the extension for multiple items can be found in Cohen et al. (2017a). In this work, we study the various practical challenges in applying such an optimization tool to a retailer. We show how to estimate the demand model and report the design of the pilot experiment done in collaboration with a retailer.

Finally, several works discuss field experiments on pricing decisions implemented at retailers. A classical successful example is the implementation at the fashion retail chain Zara (see Caro and Gallien 2012). In their work, the authors report the results of a controlled field experiment conducted in all Belgian and Irish stores during the 2008 fall-winter season. They assess that the new process has increased clearance revenues by approximately 6%. An additional recent work can be found in Ferreira et al. (2018) in which the authors collaborated with Rue La La, a flash sales fashion online retailer. The authors propose a non-parametric prediction model to predict future demand of new products, and develop an efficient solution for the price optimization problem. They estimate a revenue increase for the test group by approximately 9.7%. Pekgün et al. (2013) describe a collaboration with the Carlson Rezidor Hotel Group. In this study, the authors show that demand forecasting and dynamic revenue optimization consistently increased revenue by 2-4% in participating hotels relative to non-participating hotels. Our work shares a similar philosophy and presents a general data-driven promotion optimization tool applied to a retailer in the farm and ranch supply industry.

5.2 Business Problem

Over the last few years, retailers have become increasingly interested in developing ways to improve their promotion planning process. In this section, we (1) present the general context in which retailers plan their promotions, (2) discuss how this specifically relates to the pilot retailer, and (3) lay out the roadmap of the pilot design process.

5.2.1 General Context

As we discussed, promotions are an important tool for retailers to generate extra sales, increase store traffic, introduce new products, create brand loyalty, and price discriminate. An average tier-1 retail client runs weekly promotions in 1,000 stores, with roughly 200 categories each containing 50 to 600 items. Overall, an average store with 40,000 SKUs (Stock Keeping Units) could promote up to 2,000 SKUs every week. Most promotions consist of combinations of price promotions (temporary reductions in price), as well as promotion vehicles (communication methods to promote products) such as flyers, TV commercials, and end-cap displays. While scheduling promotions, retailers also need to satisfy various business rules and vendor funds from suppliers. In this context, the retailer's challenge is to set the right price discount and use the right promotion vehicles for the right set of products at the right time. Very often, retailers decide their promotion planning based on historical performance, past experience, and intuition. Our goal is to show that tools based on data-analytics and optimization for promotion planning can significantly increase profits, and reduce the time spent in deciding the future promotions. Due to the usually small profit margins and the large sales volume of the retail industry, a small improvement in the weekly profits can have a large impact on the retailer overall profitability.

5.2.2 Pilot Retailer

In this work, we describe the design of a pilot experiment with a tier-1 client of Oracle Retail. The pilot retailer has been in the retail business for over 50 years, supplying outdoor equipment for farms and ranches. The pilot retailer has hundreds of different stores, for the most part located in the Midwest of the United States. To help their customers enjoy the rural lifestyle, their product offerings have increased over the years to now include lawn, garden, farm and ranch supplies, livestock feed, animal health, pet food and supplies, hardware, plumbing, electrical, automotive, toys, housewares, and work clothing.

These products are continuously promoted, and naturally, the retailer faces many of the same promotion planning challenges that other retailers encounter. The pilot retailer plans its promotions at the chain-level, meaning that the promotion schedule is determined centrally for all 120,000 SKUs in all stores. The promotions consist of a mix of price discounts and advertisements through marketing vehicles; including flyers, TV and radio commercials, and online promotions in their website. This process is time-consuming, costly, and could perhaps sacrifice a substantial amount of money left on the table. Hence, we focus on improving price promotions through a price promotion optimization model. We believe that the most significant impact can be made by improving the planning of promotion discounts. The natural next step would be to address the promotion vehicle scheduling problem (more details on the promotion vehicle scheduling problem can be found in Chapter 3).

Since the pilot retailer manages its promotions centrally, sales promotions are typically supported by the entire supply chain. Consequently, there is a lead-time of roughly six weeks to plan the future promotions. Fortunately, the supply chain and promotion planning are tightly integrated, which has the benefit that stock-out situations occur rarely. However, this could lead to higher inventory costs, because inventory control has to account for the larger volumes of sales induced during promotions. Interestingly, this cost is generally much smaller relative to the revenue boost obtained from a promotion. For the three treatment items proposed for the pilot study (see Section 5.3.1), we observed that on average, a promotion leads to four times as many sales and increases the revenue by a factor of two. As a result, this confirms the intuition that the overhead inventory and logistics cost is largely compensated by the large increase in revenue from a promotion.

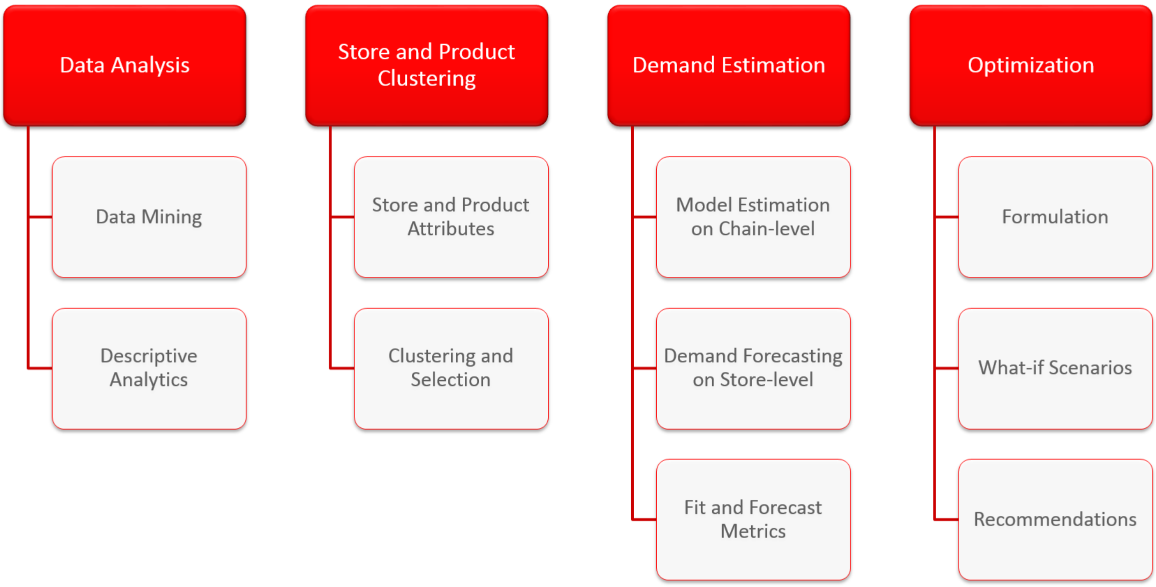


Figure 5.1: Flowchart describing the (sub)stages of the pilot process

5.2.3 Roadmap

In the following sections, we present the application of our model to the pilot retailer and the design of our pilot. First, we describe the pilot retailer's data and how we selected the products and stores for our pilot. Next, we estimate and test our demand forecasting model out-of-sample. We then discuss the promotion optimization problem applied to the pilot retailer. We calibrate our model using historical data from 2012 and 2013, and backtest it using the data from 2014. Finally, we present our recommended promotions for the retailer during 16 weeks of 2015 (June to September). As illustrated in Figure 5.1, the phases of this promotion recommendation process can be categorized into the following: (1) Data Analysis, (2) Store and Product Clustering, (3) Demand Estimation, and (4) Optimization. In each of these stages, we used different software tools. The main tools we used were: Oracle SQL and R for data management, clustering, and demand estimation; Python and Gurobi for optimization; and Microsoft Excel for building the user interface of our promotion planning tool.

5.3 Data Analysis

The pilot retailer provided us with data from 157 stores. The dataset spans a period of 153 weeks between January 2012 and December 2014, which we split into a training set of the first 104 weeks (2012-2013), and a testing set of the final 49 weeks (in 2014). For each week-store-product combination, we have data on the sales, price and promotions of the products in the stores during that week. The dataset also contains data on the relevant holidays, the total selling square footage in the store, the categorization/brand/size of the different products, and several other product attributes.

As for any large dataset, we encountered several challenges in analyzing the data. The first challenge is that the data we received is incomplete. For some product categories, certain week and store combinations were missing, promotion data was sometimes limited, and several product features were not included. Ultimately, we ended up gathering some of the product feature data manually when selecting treatment and control products.

The second challenge is that the pilot retailer carries mostly farming and outdoor equipment products, which are usually referred to as hard-lines. Customer purchasing behavior of these products could be quite different compared to grocery items, and hence, the demand models used to fit supermarket data in Cohen et al. (2017b) cannot be used directly. In both applications, demand still depends on location, price, and of course, promotions. However, one of the main differences is the strong stockpiling effect (also called pantry loading effect) observed in non-perishable grocery items. This effect might not exist for certain product categories at the pilot retailer, and we will let the data inform us on this. A second key difference is that the pilot retailer's sales can be seasonal and influenced by weather conditions. For example, at the onset of spring, many farmers need to buy supplies causing the pilot retailer's sales to peak. Finally, in comparison to the supermarket industry, outdoor and farming products are generally purchased less frequently. This is an important issue because a low rate of sales can make it more difficult to estimate demand with good accuracy. Nonetheless, in this study, we aggregate our data and extend our demand model to include a multitude of demand factors, which makes it possible to apply our estimation methods to any retailer's products.

In what follows, we describe how we select our treatment and control products and stores to remedy the aforementioned problems, and to design good conditions for our pilot study. In a pilot study, the promotion recommendations should be implemented for the treatment products in the treatment stores. The pilot retailer is responsible for setting the promotions of the treatment products in the control stores and the control products in both the treatment and control stores. This way, by selecting the most appropriate products and stores, we are able to control for structural product and store differences when assessing the impact of our promotion policies.

5.3.1 Product Selection

In this section, we discuss the product selection process. Our goal is to select several appropriate products for our pilot. Naturally, we want to select products for which the pilot retailer is interested in improved promotion recommendations. In addition, for stable results, we want to select products with a very high prediction accuracy in demand forecasting. After mining several product categories, we decided to pick the oil category. This decision was supported by at least four reasons: (1) this category was close to grocery products, in which we had some good previous experience and as a result, behavioral effects such as pantry loading were observed in the data, (2) the feedback from the sales management team was in favor of using this category, (3) the data for this category was cleaner than most other categories with some frequent past promotions, and (4) the demand prediction accuracy was very high.

The oil category contains 137 SKUs, out of which only 22 SKUs have incomplete pricing and promotion data. The remaining 115 SKUs form a representative and clean dataset that constitutes over 99% of the entire oil category in terms of sales and revenue (see Table 5.1).

Table 5.1: Sales and revenue for the entire oil category during 2012-2014

	2012	2013	2014
Sales (Units)	1,243,897	1,455,319	1,495,591
Revenue ($)	12,521,783	12,560,407	12,598,077

Table 5.1 shows that the total sales and revenue are relatively stable within the 3 years of data, with the exception of an increase in sales from 2012 to 2013. Note that we aggregate the data across all the stores, which causes the rate of sales to become more stable. We perform this aggregation because promotions are planned at the chain-level, and the same promotions are applied in each pilot treatment store.

Products in the oil category are frequently promoted, and depict limited seasonality effects. Additionally, products in this category exhibit the stockpiling effect (i.e., during promotions, consumers purchase larger quantities for future consumption). As an illustration, Figure 5.2 presents the volume of sales and the corresponding price levels versus time for one of the treatment products in the oil category during the 49 weeks of 2014.

One can see that during this time period, the retailer used four different price levels: a regular price of 3.59 and three promotional prices (1.99, 2.09 and 2.29). This time series illustrates the aforementioned characteristics: limited seasonality, frequent promotions, and the presence of

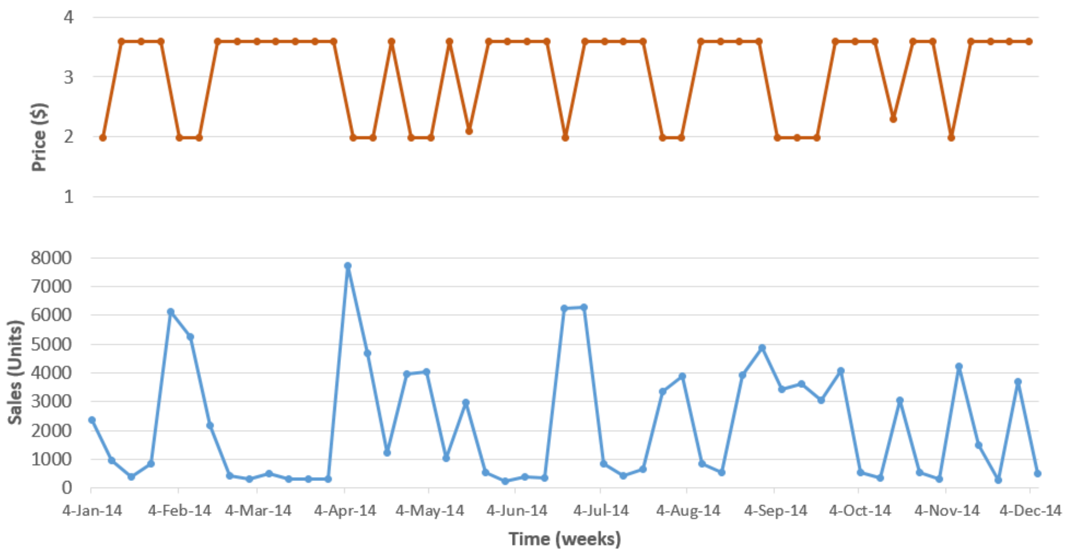


Figure 5.2: Time series of prices and sales for one treatment product in the engine oil subcategory during 2014

Figure 5.3: Example of features for a product in the oil category

the stockpiling effect. More precisely, when two (or more) consecutive promotions are used, the subsequent promotion may yield a lower boost in sales relative to the initial promotion. Examples can be seen at weeks 5 and 6, weeks 14 and 15, and so forth.

Now that we decided to focus on the oil category, our next step is to select the treatment products for our pilot. We will do so based on the products' features. Figure 5.3 shows some of the most important features of engine oils. Sometimes, the retailer does not store the data on all the features of a product or the data might be stored in textual form. In this case, we either collect the data manually or by scanning the text.

Within the oil category, we decided to focus on engine oil products as this is the largest subcategory with regards to the number of products and total sales. From this subcategory, we pick three different treatment products for which we will plan the future price promotions. The control products are simply the other products in the engine oil subcategory. Selecting more

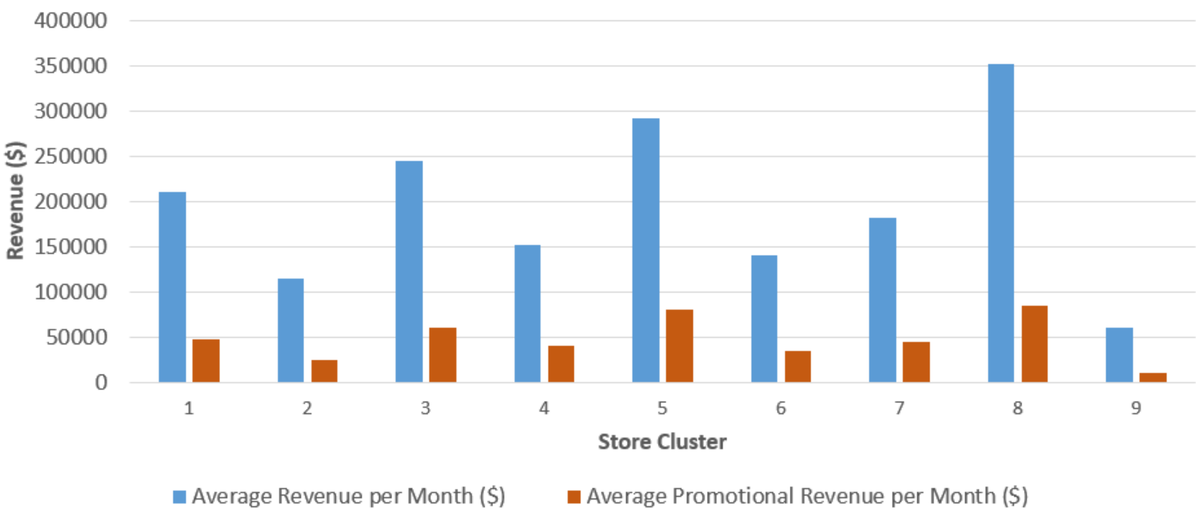


Figure 5.4: Average revenue and promotional revenue in the engine oil subcategory in the different store clusters during 2012-2014

than one product for the pilot will allow us to control for some external factors, and to convey that the impact of our tool is not restricted to a single specific product. We select our three treatment products to be different grades and oil types of the same engine oil brand. Selecting different grades and types of oil ensures that the products are not competing with each other, and prevents any cross-price effects between these products. We carefully selected these three products while consulting the management team of the pilot retailer, and ensuring that the demand forecast accuracy was high.

5.3.2 Store Clustering

Next, we select the pilot stores such that the treatment and control stores are as similar as possible. Similarity between the stores can reduce some uncontrolled external variations, and lead to a more accurate assessment of the impact of our promotion recommendations. In order to extend the scope of our pilot, we select multiple treatment and control stores. Selecting multiple stores also helps to limit the possibility of randomly picking an outlier store. To identify similar stores, we cluster the stores using *Kernel K-means* (Dhillon et al. 2005). We run this clustering method by using several attributes, such as store revenues, store promotional revenues, number of products sold in the store, products sold under promotions, and store square footage. The algorithm found 9 store clusters, for which Figure 5.4 presents the average monthly revenue and promotional revenue in each cluster.

The chart shows a large difference in average monthly revenue between the different clusters, but the average monthly promotional revenue is quite stable. Part of the difference can be

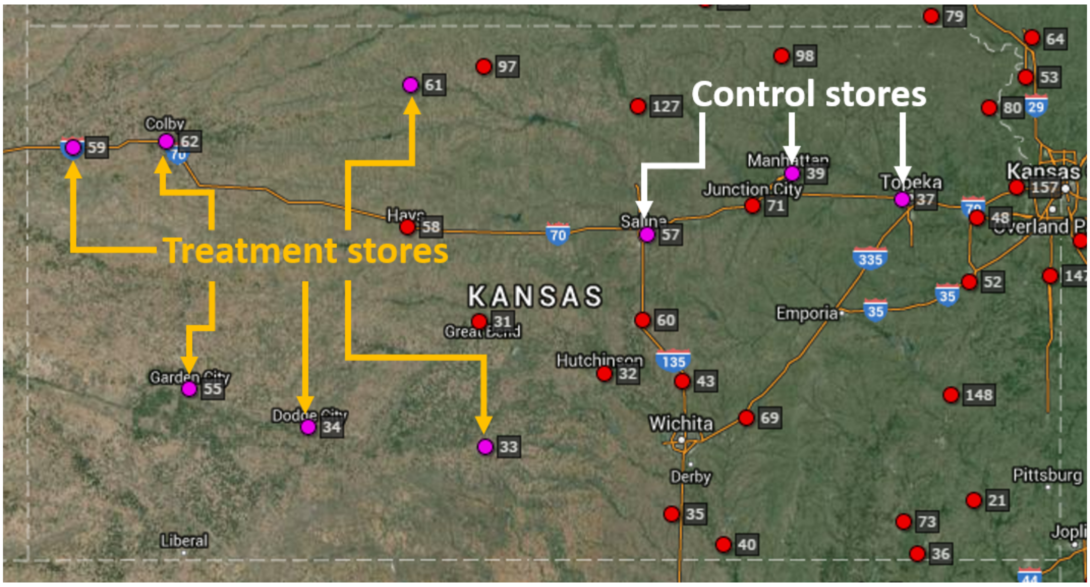


Figure 5.5: Map plotting the treatment and control stores in Kansas

explained by the fact that certain clusters are larger than others. Particularly, most clusters contain between 21 and 28 stores, the exceptions are cluster 6 with 7 stores, cluster 8 with 10 stores, and cluster 9 with 5 stores. In order to increase the robustness of our results, we want to pick a cluster with both large average revenues and promotional revenues. Even though cluster 8 is the largest in terms of revenues, it contains only 10 stores. This implies that each store in cluster 8 is not representative of the retailer's average store (since it is much larger than most stores). For that reason, we decided to pick cluster 5, which is sizable in revenues (second largest cluster) and contains 21 stores.

Having selected a cluster with 21 stores, there are some additional requirements in order to choose the treatment and control stores. In particular, it is important to pick stores from the same state, because this allows us to avoid dealing with inter-state differences such as tax regulations. Also, we need to ensure that the treatment stores are not located too close to any control store, because this could expose customers to both control and treatment stores. Customers close to both a treatment and a control store could decide to go to the lowest price store leading to a systemic bias in the results (i.e., unwanted inflation of treatment or control store sales). Under these extra conditions, we selected 3 control stores and 6 treatment stores in the state of Kansas highlighted with arrows in Figure 5.5. We choose Kansas as most of the 21 stores from cluster 5 are located in Kansas.

The map shows that the stores are all well within the state boundaries of Kansas and that the distance between the pilot and the control stores is large enough to avoid the double-exposure phenomenon described above. Table 5.2 reports some additional information about each of the 9 selected stores: in the upper part of the table, we show the sales units, revenue, promotion sales units and revenue for the 6 treatment stores, and in the lower part we show these statistics for the 3 control stores.

Table 5.2: Average monthly sales, revenue, promotional sales, and revenue in the 9 pilot stores in Kansas during 2012-2014 (top: treatment stores, bottom: control stores)

Store	Sales (Units)	Revenue ($)	Promotional Sales (Units)	Promotional Revenue ($)
Store1	16,936	250,419	3,058	78,176
Store2	23,589	308,703	3,607	83,841
Store3	26,446	347,434	3,664	88,965
Store4	16,401	235,320	2,910	79,952
Store5	21,989	237,626	3,040	66,138
Store6	20,726	251,453	2,796	66,425
Store7	41,846	274,781	4,704	78,036
Store8	38,804	277,743	4,188	75,271
Store9	42,443	338,921	6,173	108,605

Interestingly, Table 5.2 shows that even though the 9 stores come from the same cluster, there are significant differences in sales units and in revenue between the treatment and control stores. On the other hand, the variation in sales within the treatment stores and within the control stores are more limited. Since we account for store differences and compare the average impact across all treatment stores with the average impact across all control stores, this should still allow us to obtain robust results.

5.4 Solution Approach

Having selected the treatment products and treatment stores, we next present a concrete instance of the promotion planning problem for the pilot retailer. The objective is to maximize the total profit by deciding when to promote a product, and which price discount to set. In this section, we estimate a demand forecasting model, and use it as an input to the promotion optimization problem. Initially, the model was proposed and developed for a single item in Cohen et al. (2017b), and later extended to a multiple item setting in Cohen et al. (2017a). The current work focuses on applying the promotion optimization tool to the pilot retailer as well as measuring its potential impact. Note that a very similar process and methodology can be applied to any retailer.

5.4.1 Demand Estimation

The first part of the Promotion Optimization Problem (POP) is to understand how pricing and promotions affect customer behavior, specifically the demand-price curve. Since the pilot retailer sets the promotions at the chain-level, we also design our promotions for all the treatment stores at the same time. In order to do so, the POP needs accurate forecasts of the chain-level demand for each product. In what follows, we present our demand model which accounts for the current and past prices, the promotion and seasonality effects. We then discuss the estimation procedure, and the out-of-sample prediction performance of our model.

In Cohen et al. (2017b), supermarket demand forecasting models have been proposed with a high accuracy in terms of out-of-sample prediction. Similar models can be used for farming supplies and other hard-lines for the pilot retailer. However, compared to other contexts, seasonality plays a large role in the pilot retailer's industry. In particular, the demand can be significantly lower during the winter than during the spring or the summer when farmers need to buy supplies. In addition, the pilot retailer notices large demand spikes during certain national holidays. The estimated demand forecasting model we used for product i at time t is given by:

$$\log(\hat{d}_{it}) = \hat{\alpha}_i + \hat{\beta}_i^0 \log(p_{it}) + \hat{\beta}_i^1 \log(p_{i,t-1}) + \hat{\delta} \cdot t + \sum_m \hat{\eta}_m x_{mt} + \sum_h \hat{\theta}_h y_{ht}, \tag{5.1}$$

where $\hat{d}_{it}$ is the expected demand in the treatment stores for treatment product i at time t, p_{it} is the price of treatment product i at time t, x_{mt} indicates whether time t falls in month m (1 if yes, 0 if not), and y_{ht} indicates whether holiday h falls in period t (1 if yes, 0 if not). Initially, we estimate a linear regression model with all the observable variables that could have an impact on demand. Through a stepwise selection process, based on the significance of the variables and minimizing the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC), we arrive at the final demand forecasting model in (5.1). During this process, we have tried multiple models with some non-linear transformations of the variables. In our out-of-sample tests, we observed that the log-log model in (5.1), where we take the logarithm of demand and price provides the best fit. This validates the popularity of the SCANPRO model which is often used in practice and in the marketing literature (Wittink et al. 1988).

The parameter $\hat{\alpha}_i$ is an intercept that captures the product effect over all the treatment stores. We include the logarithm of the current price in order to capture the current price-

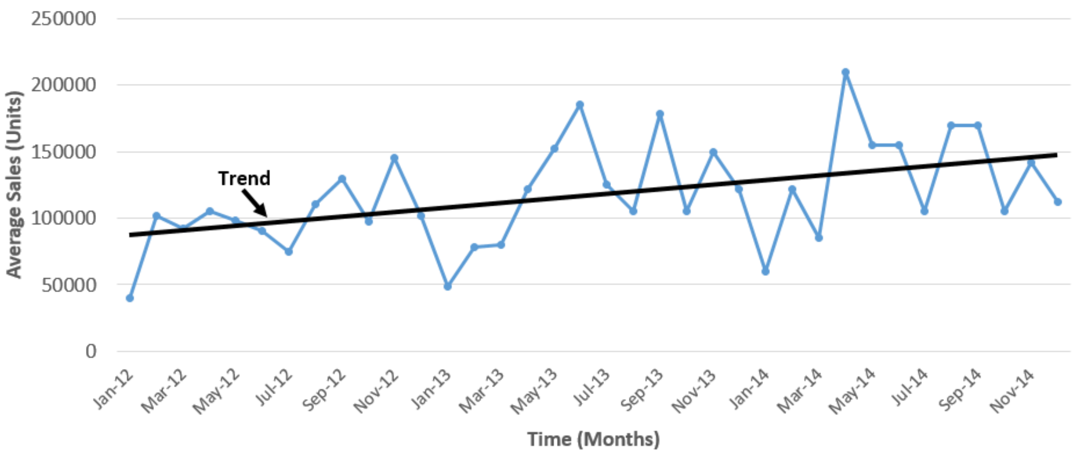


Figure 5.6: Time series and trendline of average monthly sales in the engine oil subcategory in the 9 pilot stores

elasticity through $\hat{\beta}_i^0$. The parameter $\hat{\beta}_i^1$ captures the effect of the price in the past period on the current demand. The effect of the price in the past period relates to the stockpiling effect: the phenomenon in which promoting an item causes the customer to purchase larger quantities towards future consumption. Initially, we capture the stockpiling effect by including the past prices of multiple periods back. In the end, we only include the prices of M_i past periods. The memory, M_i, represents the number of past prices of product i that affect the current demand of product i. The memory of each product is not an input and is estimated from the data through the iterative selection process. For each of the three treatment products, we found a memory of one period, i.e., $M_i = 1$. Outside of the product, price and stockpiling effects, our demand model also controls for: the demand trend (parameter $\hat{\delta}$), monthly seasonality in demand (parameters $\hat{\eta}_m$), and holiday-boost effect (parameters $\hat{\theta}_h$).

As we saw from the sales units and revenues of the entire oil category, there is a small but steady increase in the total sales. Figure 5.6 supports this observation by showing that average monthly sales trend upward over time. For this reason, we include the parameter $\hat{\delta}$ in our model to capture the trend in sales.

In order to control for seasonality effects, we include a factor that captures the base monthly sales. Figure 5.7 presents the average monthly sales over the three years of data. One can see that there is some considerable variation in the sales between the months. Specifically, the winter months sell less on average, while the spring and summer months sell well beyond average. A partial explanation for the first period could be that it corresponds to the North American planting season when the engine oil of planting machines needs to be refreshed, whereas the

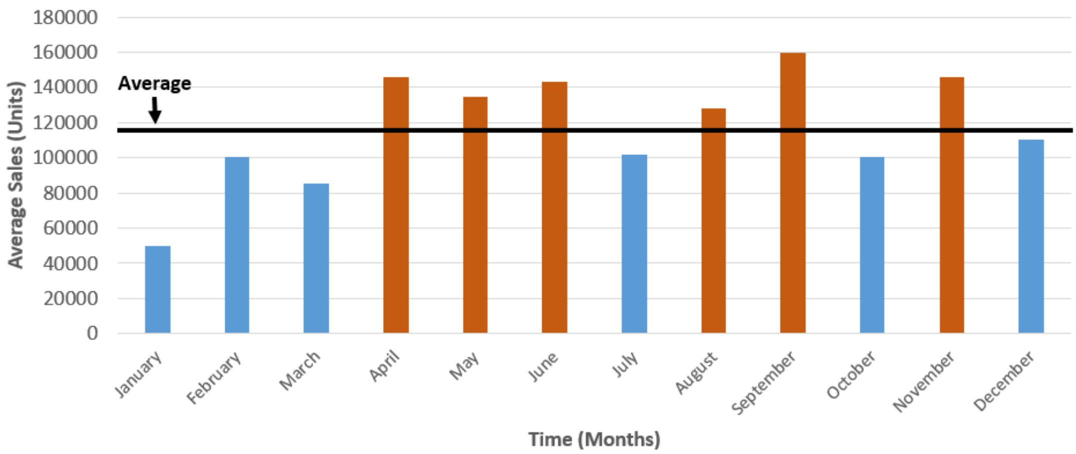


Figure 5.7: Average monthly sales in the engine oil subcategory in the 9 pilot stores during 2012-2014

second season corresponds to the North American harvesting season during which harvesting machines are used. Other explanations could be the fact that the weather improves so that people use bicycles, motorcycles and other recreational machines more often.

Finally, we observed some significant variations in oil demand around the holidays. Figure 5.8 shows the average sales during the week in which a holiday fell. During the weeks of Father's day and Thanksgiving day, we notice the largest demand spikes. The spike on Father's day could be explained by the fact that this holiday falls towards the latter half of the spring (June in the United States), when the weather has improved and fathers may be interested in recreational activities such as riding a motorcycle. Thanksgiving day seems to be a period in which cars are refreshed for the winter. On the other hand, the demand on Christmas and New Year are relatively low, possibly due to store closure, the winter season (cold weather), and potential stockpiling from Thanksgiving.

In general, our demand forecasting model can capture cross-item effects, which can manifest themselves in two ways: cannibalization and halo. Namely, promoting an item causes either a reduction in sales for its substitutes (cannibalization), or an increase in sales for its complements (halo). Thus, our demand forecasting model can include cross-item effects (modeled as dependence on prices of other products), and the stockpiling effect (modeled as dependence on past prices). Formally, we can write our model as a general time-dependent demand function for product i at time t, denoted by $d_t^i(p_t^i, p_{t-1}^i, \dots, p_{t-M_i}^i, \mathbf{p_t^{-i}})$, that explicitly depends on the current self price and up to M_i self past prices $p_t^i, p_{t-1}^i, \dots, p_{t-M_i}^i$, as well as on some (or all) cross current prices $\mathbf{p_t^{-i}}$. In the final model, the cross-item effects for the 3 treatment products were statistically insignificant. With our selection of products, we deliberately aimed to keep

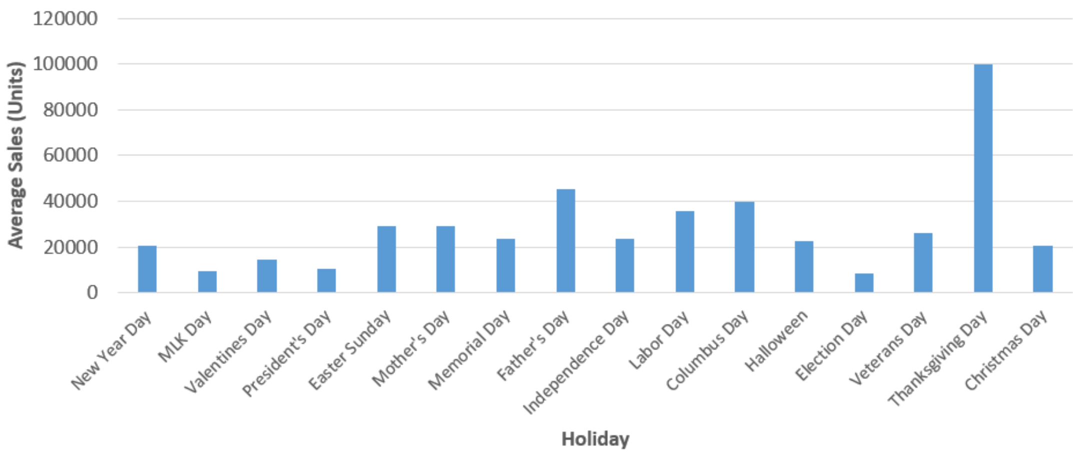


Figure 5.8: Average weekly sales in the engine oil subcategory in the 9 pilot stores during holidays in 2012-2014

cross-item effects minimal. Otherwise, the results could be obfuscated, as promotions for the treatment products could significantly impact the profits of the other treatment and control products.

We next present the estimation results of the demand model in (5.1). We split the data into two parts: a set of training data from the first 104 weeks (2012-2013) and a set of testing data from the final 49 weeks (2014). The parameters of model (5.1) are estimated by using the ordinary least squares regression method on the training set. The estimated parameters are reported in Tables 5.3 and 5.4. In Table 5.3, we present the product-dependent parameter estimates for the 3 treatment products. We observe that the base sales of product 1 are smaller relative to products 2 and 3, but the estimates are still relatively close. The estimated price elasticities have all a similar magnitude (between -5.678 and -5.981), but the past price effect is stronger for products 2 and 3.

Table 5.3: Estimated product parameters

Variable		Product1	Product2	Product3
Product	$\hat{\alpha}_i$	5.2978	6.3014	6.9543
Price	$\hat{\beta}_i^0$	-5.775	-5.981	-5.678
Past price	$\hat{\beta}_i^1$	0.8310	1.1004	1.1966

Table 5.4 shows the estimated monthly seasonality parameters. The first row corresponds to the small positive estimate for the trend in demand. The demand factor indicates that the demand increases by approximately 0.17% every week, or equivalently a yearly increase of 9.24%. The second part of the table shows the parameter estimates and demand factors for the monthly

sales. Three months (April, May and June) are left out of the table, because the parameter estimates of these three months were statistically insignificant. The negative estimates for the nine other months show that their estimated sales are lower than the estimated sales of the three base months. Especially, the demand factors of December, January, and February show that the sales in the winter months are 36%, 38%, and 41% lower when compared to the spring months. This confirms our earlier intuition that the winter period can admit lower sales, whereas the spring has the most sales during the year. Finally, the third part of the table shows the estimates for the holiday factors. Having corrected for the trend in demand and monthly base demand, the only significant impact on sales can be seen during New Year, Martin Luther King day and Christmas. Compared to the other holidays, such as Father's day and Thanksgiving, these three holidays lead to lower sales. In the week of New Year, the demand drops by 32% and in the week of Christmas it decreases by 29%. We can attribute the large decrease in the week of Christmas to the closure of most stores during Christmas day.

Table 5.4: Estimated seasonality parameters

Variable		Estimate	Demand Factor
Trend	$\hat{\delta}$	0.0017	1.0017
January	$\hat{\eta}_{jan}$	-0.4751	0.6218
February	$\hat{\eta}_{feb}$	-0.5275	0.5901
March	$\hat{\eta}_{mar}$	-0.1753	0.8392
July	$\hat{\eta}_{jul}$	-0.1319	0.8764
August	$\hat{\eta}_{aug}$	-0.156	0.8556
September	$\hat{\eta}_{sep}$	-0.1445	0.8655
October	$\hat{\eta}_{oct}$	-0.1736	0.8406
November	$\hat{\eta}_{nov}$	-0.3305	0.7186
December	$\hat{\eta}_{dec}$	-0.4434	0.6419
New Year Day	$\hat{\theta}_{nyd}$	-0.3834	0.6815
MLK Day	$\hat{\theta}_{mlk}$	-0.0973	0.9073
Christmas Day	$\hat{\theta}_{chr}$	-0.3408	0.7112

Next, we present the accuracy of our demand forecasting model in (5.1). We use the training set to fit our model and then, apply it on the testing set. The predicted and actual demand during the testing period can then be used to calculate the out-of-sample forecasting metrics. In this work, we consider both the R^2 and MdAPE (Median Absolute Percentage Error), which is defined below.

The out-of-sample forecasting metrics are presented in Table 5.5. In this case, the in-sample R^2 for the entire oil category is 0.92, while the out-of-sample R^2 is 0.89. Note that this is considered as a very good prediction accuracy in the retail industry, especially for products with a less stable sales rate such as the pilot retailers' products. In addition, the fact that the in-sample and out-of-sample R^2 are close together indicates that there is no sign of overfitting and that the model generalizes well. Similar results are observed for the prediction accuracy at the brand level and at the individual product level. Next to the out-of-sample R^2, the table also shows the out-of-sample MdAPE metric. The MdAPE is calculated as the median of the absolute percentage error in weekly demand, that is:

$$MdAPE = \text{median} \left\{ \left| \frac{\hat{d}_{it} - d_{it}}{d_{it}} \right| \right\},$$

where the median is taken over all the time periods during the testing period (2014). We observe in Table 5.5 that the MdAPE is less than 30% in every setting. Based on the pilot retailer's experience, these are good forecasting performance metrics. Note that the MdAPE improves by 5 percentage points from the entire category to the specific treatment brand. This shows that the treatment product forecasts are significantly better relative to some of the other products.

Table 5.5: Forecasting metrics on the testing set (2014)

Forecasting Metric	Oil Category	Treatment Brand	Product1	Product2	Product3
R^2	0.89	0.90	0.82	0.86	0.93
MdAPE	0.2994	0.2400	0.2777	0.2018	0.2544

In Figure 5.9, we present a comparison of the actual and predicted sales for one of the treatment products during the testing period (2014). One can see that the predictions follow the same pattern as the actual sales, often with a similar magnitude. Only in some of the highest selling periods, our model under-predicts. Nevertheless, this difference is still small in relative terms with an MdAPE of 26.9% during all the promotion periods. Overall, we believe that the demand model we proposed provides a high out-of-sample prediction accuracy.

5.4.2 Promotion Optimization

In this section, we discuss how we generate the recommended promotion policy by solving the Promotion Optimization Problem (POP), that we formulate below. As input to the POP, we use the demand forecasting model estimated in Section 5.4.1. The algorithms we are using to solve the problem are based on linear programming, and are developed in Cohen et al. (2017b) and in Cohen et al. (2017a). Since the algorithm is based on linear programming, our promotion planning tool is efficient and very scalable. This allows us to run and test many "what-if"

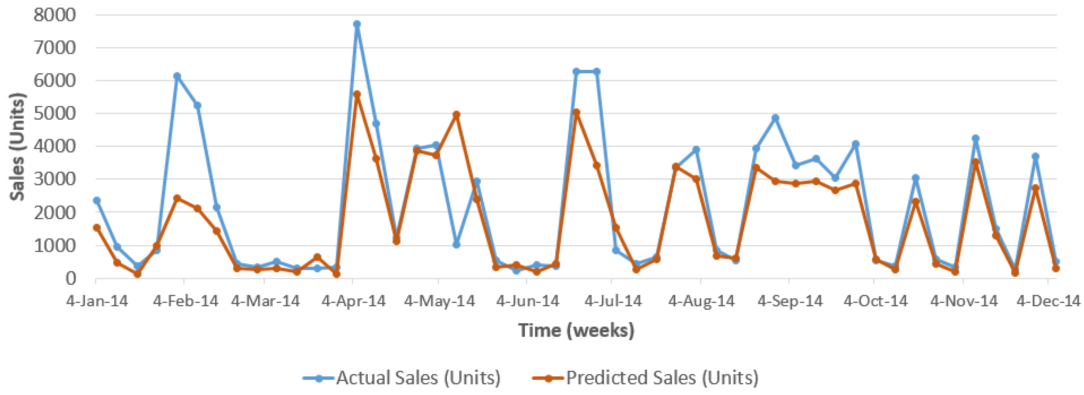


Figure 5.9: Time series of actual and predicted sales for one treatment product during the testing period in 2014

scenarios in order to reach a promotion policy which is robust to variations in the demand parameters.

Our main goal is to show that the expected profits could be increased significantly by implementing the recommended promotion policy from solving the POP relative to the current promotions used by the retailer. The promotion policy provides recommendations to the retailer in terms of: (1) which products should be promoted in which stores; (2) when to schedule the promotions; and (3) what should be the promotions depths.

In what follows, we present the POP formulation. We also briefly discuss some of the key technical results that allow us to solve the POP efficiently and provide intuitive pricing solutions to retailers (as we mentioned, more details can be found in Cohen et al. 2017a,b). The objective is to maximize the total profits during the upcoming selling season (e.g., a quarter of 13 weeks or a period of 16 weeks), whereas the decision variables are for each time period, which products to promote and what prices to set (i.e., the promotion depths). We also incorporate various important real-world business requirements that should be satisfied (examples of such requirements are presented below). We first introduce some notation:

- N - Number of products in the category.
- T - Number of weeks in the horizon.
- L_i - Limitation on the number of times we are allowed to promote product i.
- S_i - Number of separating periods (restricting the separation between two successive promotions) for product i.

- $\mathcal{Q} = \{q^0 > q^1 > \cdots > q^k > \ldots > q^K\}$ - Price ladder, i.e., the discrete set of admissible prices.
- q^0 - Regular (non-promoted) price, which is the maximum price in the price ladder.
- q^K - Minimum price in the price ladder.
- c_t^i - Unit cost of product i at time t.

We assume without loss of generality that all the products have the same regular price q^0, as one can normalize it for each item, and assume conveniently that $q^0 = 1$. In order to simplify the exposition, we also assume that all the products have the same time-independent price ladder (one can easily extend to different price ladders at the expense of a more cumbersome notation). The decision variables are the prices set at each time period for each product denoted by $p_t^i \in \mathcal{Q}$. Since we are considering a set of discrete prices only (motivated by the business requirement of a finite price ladder), one can rewrite the price p_t^i at time t as follows:

$$p_t^i = \sum_{k=0}^{K} q^k \gamma_t^{ik},$$

where γ_t^{ik} is a binary variable that is equal to 1 if the price q^k is selected from the price ladder at time t for product i, and 0 otherwise. This way, the decision variables are now the set of binary variables γ_t^{ik}; $\forall t = 1, \ldots, T$; $\forall i = 1, \ldots, N$ and $\forall k = 0, \ldots, K$, for a total of $NT(K+1)$ variables (for a typical Oracle Retail client, N is around 250, K is about 20 and T is 13, so that the number of binary variables amounts to 65,000). In addition, we require the following constraint to ensure that exactly a single price is selected:

$$\sum_{k=0}^{K} \gamma_t^{ik} = 1; \qquad \forall t.$$

Very often, several business requirements are imposed on the prices. The first example is a limitation on the number of promotions for each product (denoted by L_i) so as to preserve the image of the store/brand. This can be captured by the following constraint:

$$\sum_{t=1}^{T} \sum_{k=1}^{K} \gamma_t^{ik} \leq L_i.$$

A second example is a rule on separating two successive promotions by some minimal time period, denoted by S_i for product i:

$$\sum_{\tau=t}^{t+S_i} \sum_{k=1}^{K} \gamma_\tau^{ik} \leq 1 \qquad \forall t.$$

As mentioned above, the objective of the POP is to maximize the total profits over a finite planning horizon. The prices are set under the assumption that unit costs are fixed and known, which is a reasonable assumption for many retailers. However, future demand and sales are unknown to the retailer and need to be forecasted. As discussed in Section 5.4.1, we estimate realistic demand models tailored to the specific context of the retailer by using historical data. In this case, we managed to obtain a demand model with a high out-of-sample prediction accuracy. Recall that in its general form, our demand model for product i at time t can be written as follows: $d_t^i\big(p_t^i, p_{t-1}^i, \ldots, p_{t-M_i}^i, \mathbf{p_t^{-i}}\big)$. We next present the optimization formulation:

$$\begin{aligned}
\max_{\gamma_t^{ik}} \quad & \sum_{i=1}^{N} \sum_{t=1}^{T} (p_t^i - c_t^i) d_t^i(p_t^i, p_{t-1}^i, \ldots, p_{t-M_i}^i, \mathbf{p_t^{-i}}) \\
\text{s.t.} \quad & p_t^i = \sum_{k=0}^{K} q^k \gamma_t^{ik} \\
& \sum_{t=1}^{T} \sum_{k=1}^{K} \gamma_t^{ik} \leq L_i \qquad \forall i \\
& \sum_{\tau=t}^{t+S_i} \sum_{k=1}^{K} \gamma_\tau^{ki} \leq 1 \qquad \forall i, t \\
& \sum_{k=0}^{K} \gamma_t^{ik} = 1 \qquad \forall i, t \\
& \gamma_t^{ik} \in \{0, 1\} \qquad \forall i, t, k
\end{aligned} \tag{Multi-POP}$$

As we mentioned, the binary decision variable γ_t^{ik} is equal to 1 if the price of item i at time t is selected to be q^k from the price ladder, and 0 otherwise. In the formulation (Multi-POP) above, all the constraints follow from the two aforementioned business rules. One can naturally include some additional business rules, depending on the specific requirements and needs of the retailer.

The resulting formulation is a Non-Linear Integer Programming problem, which is proven to be NP-hard in Cohen et al. (2018a). However, as we mentioned, one can use the solution approach developed in Cohen et al. (2017b) for the case of a single item (as well as the extension for multiple items presented in Cohen et al. 2017a). These solution approaches are based on a linear approximation to the objective that allows us to solve the problem efficiently as a linear program (LP), by showing the integrality of the integer programming formulation.

5.4.3 Empirical Validation

In order to convince the relevant executives and decision makers of the benefits of using our promotion planning tool, we next demonstrate the value of optimized promotion planning on historical data. To this end, we use the demand forecasting model estimated in Section 5.4.1 and plug it into the (Multi-POP) formulation. Furthermore, we set the parameters of the (Multi-POP) formulation, such as the price ladder, the limit on the number of promotions, and the number of separating periods equal to the values that the pilot retailer used in the testing period. Optimizing this (Multi-POP) instance yields the optimal promotion plan for 2014 (i.e., a time horizon of 49 weeks), which we can then compare to the actual promotion policy that was implemented in 2014 by the pilot retailer. Altogether, this backtest provides an empirical validation on historical data of how much an optimized promotion planning could have improved the profits in 2014. Table 5.6 shows the potential improvement in total profits, revenues and sales for 2014 when comparing the implemented promotion plan to our optimized promotion policy.

Table 5.6: Key Performance Indicators (KPI) of the treatment products in all stores during 2014

Promotions:	Actual	Actual	Optimized	Improvement
Demand:	Actual	Forecasted	Forecasted	
KPI 2014	(a)	(b)	(c)	$\frac{(c)-(b)}{(b)}$
Profit ($)	$169,190.53	$157,535.49	$173,193.78	+ 9.94%
Revenue ($)	$1,478,905.89	$1,319,001.47	$1,334,419.66	+1.17%
Sales (Units)	690,414	612,265	612,138	-0.02%

The first column (a) reports the profit, revenue, and sales that were actually attained in 2014. For the second column (b), we use the historical prices to compute the profit, revenue and sales units, but instead of using the actual realized demand we use our demand forecasting model. The differences between columns (a) and (b) indicate the aggregate error in our demand prediction model in 2014. Note that the difference between the yearly actual and predicted sales is approximately 12%, which is small compared to the 24% MdAPE describing the brand's weekly error. Since the values in Table 5.6 are aggregated over the three treatment products and the entire year, the forecast error (for one product in a given week) is also quite small in relative

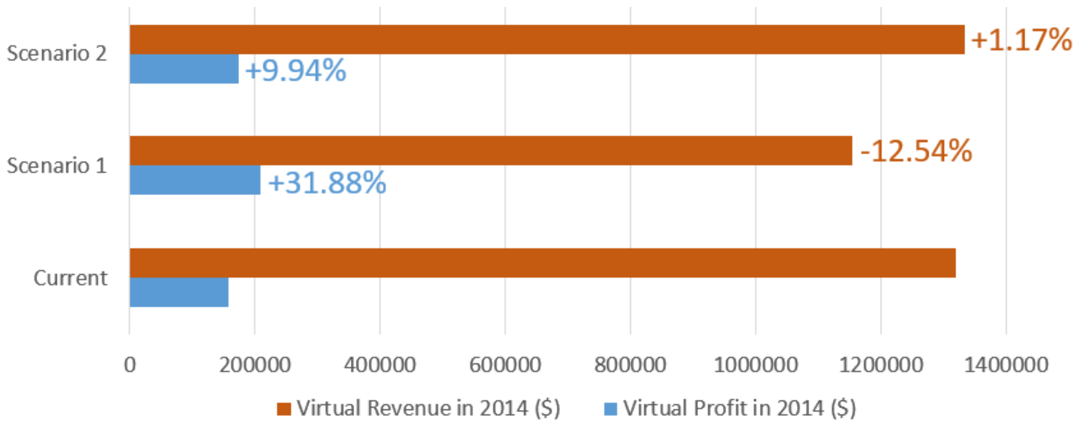


Figure 5.10: Bar chart showing the rescaled revenues and profits of current practice and two optimization scenarios with percentage change from current practice

terms. The third column (c) contains the resulting KPIs of our optimization model. To ensure a fair comparison, we compare columns (b) and (c) and hence, use the same demand forecasting model for both promotion policies. The final column shows that our optimized promotion policy could lead to nearly 10% additional profits, around 1% extra revenue, and an equal amount of sales units. This backtest clearly suggests that optimizing the retailer's promotion planning using our tool could have a significant impact on the profits, while maintaining the level of revenues.

In addition to optimizing the promotion policy for 2014, we also run multiple "what-if" scenarios. Since the method used to solve the POP runs very fast (within milliseconds for instances with hundreds of items), it allows us to rerun the model for a variety of parameter settings. Our promotion policy reported in Table 5.6 was designed under the pilot retailer's business rule that the yearly revenues should not decrease from last year. In Figure 5.10, we compare the profits and revenues when this business rule is not imposed (Scenario 1) to the case where this business rule is imposed (Scenario 2). In addition, the bottom bars show the revenue and profit of the actual promotion plan from column (b). In the end, our discussions with the managers conveyed the importance of this business rule, as a 12.54% loss in yearly revenues (Scenario 1) could be too risky and not well accepted.

These "what-if" scenarios can also be used to assess the robustness of our solution. In particular, we performed several "what-if" scenarios by varying the estimated demand parameters within their confidence intervals, as well as the input parameters related to some of the business rules (e.g., the limitation on the number of promotions). From this sensitivity analysis, we observed that the promotion policies are quite robust with respect to the expected profits.

5.5 Business Impact

In this section, we quantify the business impact of our promotion planning tool. Following the conclusive empirical validation presented in Section 5.4.3, the pilot retailer gave us the green light to design a field experiment. We next describe the details of the experiment as well as our promotion recommendations. We then discuss the potential broader impact for Oracle Retail's clients.

The pilot retailer agreed to start the live pilot on Tuesday, June 2, 2015 and end it on Tuesday, September 22, 2015 (for a total duration of 16 weeks). Consequently, we used all the past data in order to recalibrate our demand models. Adding more recent data (from both 2014 and the beginning of 2015) allowed us to estimate a more robust demand forecasting model. Though, we note that the estimated parameters are similar to the ones presented in Section 5.4.1. Next, we solve the promotion optimization problem for the 3 treatment items in the 6 treatment stores when using a time horizon of 16 weeks. In order to reach a robust solution, we also performed several "what-if" scenarios by perturbing the estimated demand parameters.

During our interactions with the pilot retailer, we were informed of special business requirements. For example, special promotional events had to be scheduled during the week of Father's day (June 21) and Labor day (September 7). The price ladder starts at the \$3.59 regular price of the three products, and decreases by 10 cents each time. In addition, the number of promotional weeks for the pilot was limited to 6. Finally, promotions are allowed to last for one or two weeks only and always begin on a Tuesday morning.

The promotion optimization tool suggested to schedule a promotional price of \$2.09 (i.e., a 41.7% discount) during the following weeks: June 16 to June 23, June 30 to July 7, July 21 to July 28, August 11 to August 18, September 1 to September 8, and September 15 to September 22. We make the following observations. First, note that both special events (Father's day and Labor day) are naturally included. Second, the promotions suggested by our tool are relatively spaced out, with a 2-3 weeks interval between successive promotions. Third, the optimization model suggests to use all six allowed promotions. Fourth, note that a constant discount was set for all the promotions. This has the added benefit that the recommended promotion plan is easy to implement for the retailer and store managers.

In order to accurately assess the pilot impact, our suggested promotions are implemented for the 3 treatment products in all 6 treatment stores. For all other engine oil products in these 6 stores, the retailer is responsible for setting its own promotion schedule. In addition, the retailer is also responsible to set the promotions for all the products in the 3 control stores (including the 3 treatment products). This allows us to isolate and control for the store and product effects, when inferring the impact of our approach. Unfortunately, we could not follow-up with the pilot retailer regarding the outcome of our promotion recommendations as they went through a major acquisition.

In this chapter, we discussed in detail the application of our promotion model and tool to a specific retailer. As we mentioned, this tool emerged from a broader collaboration between industry and academia. The goal of this collaboration was to create a general data-driven tool which can be applied to any retailer, works for a general demand model, can capture a wide range of business rules, and is directly calibrated from data. We see the work presented in this chapter to be only the beginning of improving retail promotions by using data analytics and promotion optimization tools.

5.6 Conclusions

This chapter presents the application of our promotion planning model to a client of Oracle Retail, and the design of a pilot to test the impact of optimized promotion planning. In retail, an improved promotion planning can lead to significant improvements in total profits. Applying our promotion optimization tool to the pilot retailer brought several practical challenges. In this chapter, we discussed those challenges and the ways we addressed them.

For the pilot design, we obtained three years of sales data (between 2012 and 2014). Our first step was to select the right products and stores for our pilot. By splitting this dataset into a two year long train set, and one year long test set, we are able to empirically validate our model on historical data. Certain categories of products missed sales or promotion information. We selected our pilot treatment and control products from the oil category, because its data was complete and our prediction models yielded accurate forecasts for this category. The pilot retailer has more than a hundred stores, and we observed large differences between these stores (in terms of sales volume, revenues, square footage etc.). In order to reduce the variation in our results, we clustered the retailer's stores and selected similar stores for our treatment and control. Altogether, we established for which products and in which stores the optimal promotion plan should be implemented.

Our promotion planning tool uses data analytics and optimization techniques to first estimate a demand forecasting model, and then solve a promotion optimization formulation. The main goal is to set the right promotions at the right time so as to maximize profits. The demand forecasting model captures several important factors, such as brand, price, and promotion effects. In contrast to other retail industries, the demand for farming and ranch supplies is highly seasonal, leading us to include demand trend, monthly seasonal effects and holiday demand factors. We estimate the demand forecasting model on the training set, and validate it on the testing set. The out-of-sample R^2 and MdAPE are on the order of 0.90 and 0.24 respectively, suggesting that the prediction accuracy is high. The promotion optimization problem is solved efficiently using an approximation based on linear programming while satisfying some practical retail business rules. We empirically validate our tool using historical parameters for the business rules, and optimizing the promotions for the testing period (i.e., 49 weeks in 2014). This backtest shows that the expected profits in the testing period could have been improved by 10%, while maintaining the same revenue level. This convinced the pilot retailer to setup a 16 week field experiment for which we present our promotion recommendations. Finally, we discuss the generality of our model and the broader potential impact it can have on retail promotion planning.

Chapter 6

Conclusions

In this thesis, we study how analytics can improve decision-making in promotional pricing and advertising. We introduce new practical problems, present mathematical formulations, develop new algorithmic methods, and assess their theoretical and practical performance. Each of these problems are motivated by collaborations with the online advertising and grocery retail industries.

Firstly, we analyze the online advertising portfolio optimization problem. In the online space, we wish to target the right advertisement to the right customer. We model this problem as a Multi-Armed Bandit (MAB) problem with periodic budgets. We propose to solve this problem with an Optimistic-Robust Learning algorithm that uses ideas from Upper Confidence Bound algorithms and robust optimization. Theoretically, the expected regret of this algorithm is on the order of $O(\ln(T)^2)$, which is close to the bound for regular MAB problems. Practically, we show that it performs strongly against benchmark algorithms, as we decrease regret by at least 10-20% compared to these benchmarks.

Secondly, we analyze the promotion vehicle scheduling problem. Brick-and-mortar retailers want to schedule the right promotion vehicles (e.g., flyers, commercials) for the right items at the right time. This problem is modeled as a non-linear integer optimization problem that is shown to be NP-hard. Nevertheless, we propose a greedy algorithm and an approximate integer program to solve the problem. Both theoretically and practically, we show that the algorithm quickly finds near-optimal solutions. These solutions also have a significant practical impact as they indicate profit improvements on the order of 2-9%.

Thirdly, we analyze the vendor fund problem. Suppliers want retailers to promote their products to the customer, and hence, suppliers offer vendor funds that entice retailers to promote. We model this problem as a bilevel optimization problem. To solve the lower-level problem we use linear programming relaxations, after which the upper-level problem can be solved with ideas from game theory. We show that vendor funds with pass-through constraint mitigate forward-buying by the retailer and coordinate supply chains on the short-term.

In the end, we consider a pilot study on the potential impact of analytics on promotional pricing. We collect data from a large retailer, and apply our tools to their data. We show that this can lead to a profit improvement of roughly 10% on the available historical data. We also present our recommendations based on solving the model for the future pilot.

In general, our models indicate that advertisers and retailers can increase their profits by 2-20% by using a data-driven approach. These improvements show the significant impact that analytics can have in promotion planning. In fact, we see similar opportunities for the use of analytics emerging in other areas. At the same time, the abundance of data and the increased speed at which it is collected, can enable the development of algorithms that can learn personalized preferences in a short period of time. Ultimately, customizing decisions to the individual through the use of analytics is a goal to strive for as it can be a tremendous benefit to everyone's life.

Appendix A

Appendix of Chapter 2

A.1 Proofs of Section 2.5

A.1.1 Proof of Lemma 2.5.3

For notational convenience, we will omit the condition that $\tau_{jt} \geq L_{ijt}$, and presume it given. For $x^{\Pi}_{it} = 0$ and $x^{\Pi}_{jt} = 1$ where $1 \leq i < j \leq n$ to hold, we need that $\frac{\overline{r}_{it}+\Gamma_{it}}{\overline{c}_{it}+\Lambda+\Delta_{it}} \leq \frac{\overline{r}_{jt}+\Gamma_{jt}}{\overline{c}_{jt}+\Lambda+\Delta_{jt}}$, because of the structure of the algorithm. This means that we can bound the desired probability by

$$\mathbb{P}\left(x^{\Pi}_{it} = 0 \cap x^{\Pi}_{jt} = 1\right) \leq \mathbb{P}\left(\frac{\overline{r}_{it}+\Gamma_{it}}{\overline{c}_{it}+\Lambda+\Delta_{it}} \leq \frac{\overline{r}_{jt}+\Gamma_{jt}}{\overline{c}_{jt}+\Lambda+\Delta_{jt}}\right)$$

Note that the expression for Γ_{it} means we can write

$$\frac{\overline{r}_{it}+\Gamma_{it}}{\overline{c}_{it}+\Lambda+\Delta_{it}} = \frac{\overline{r}_{it} + \frac{\overline{r}_{it}}{\overline{c}_{it}+\Lambda}\Delta_{it} + \frac{(\eta+\Lambda+1)(\overline{c}_{it}+\Lambda+\Delta_{it})}{(\eta+\Lambda)(\overline{c}_{it}+\Lambda)}\sqrt{\frac{\ln(t)}{\tau_{it}}}}{\overline{c}_{it}+\Lambda+\Delta_{it}} = \frac{\overline{r}_{it}}{\overline{c}_{it}+\Lambda} + \frac{(\eta+\Lambda+1)\sqrt{\frac{\ln(t)}{\tau_{it}}}}{(\eta+\Lambda)(\overline{c}_{it}+\Lambda)},$$

which implies that we want to bound

$$\mathbb{P}\left(\frac{\overline{r}_{it}+\Gamma_{it}}{\overline{c}_{it}+\Lambda+\Delta_{it}} \leq \frac{\overline{r}_{jt}+\Gamma_{jt}}{\overline{c}_{jt}+\Lambda+\Delta_{jt}}\right) = \mathbb{P}\left(\frac{\overline{r}_{it}}{\overline{c}_{it}+\Lambda} + \frac{(\eta+\Lambda+1)\sqrt{\frac{\ln(t)}{\tau_{it}}}}{(\eta+\Lambda)(\overline{c}_{it}+\Lambda)} \leq \frac{\overline{r}_{jt}}{\overline{c}_{jt}+\Lambda} + \frac{(\eta+\Lambda+1)\sqrt{\frac{\ln(t)}{\tau_{jt}}}}{(\eta+\Lambda)(\overline{c}_{jt}+\Lambda)}\right).$$

Now, whenever the event $\frac{\overline{r}_{it}}{\overline{c}_{it}+\Lambda} + \frac{(\eta+\Lambda+1)\sqrt{\frac{\ln(t)}{\tau_{it}}}}{(\eta+\Lambda)(\overline{c}_{it}+\Lambda)} \leq \frac{\overline{r}_{jt}}{\overline{c}_{jt}+\Lambda} + \frac{(\eta+\Lambda+1)\sqrt{\frac{\ln(t)}{\tau_{jt}}}}{(\eta+\Lambda)(\overline{c}_{jt}+\Lambda)}$ occurs, at least one of the three following events must occur: (i) $\frac{\overline{r}_{it}}{\overline{c}_{it}+\Lambda} \leq \frac{\mu^r_i}{\mu^c_i+\Lambda} - \frac{(\eta+\Lambda+1)\sqrt{\frac{\ln(t)}{\tau_{it}}}}{(\eta+\Lambda)(\overline{c}_{it}+\Lambda)}$, (ii) $\frac{\mu^r_j}{\mu^c_j+\Lambda} + \frac{(\eta+\Lambda+1)\sqrt{\frac{\ln(t)}{\tau_{jt}}}}{(\eta+\Lambda)(\overline{c}_{jt}+\Lambda)} \leq \frac{\overline{r}_{jt}}{\overline{c}_{jt}+\Lambda}$, or (iii) $\frac{\mu^r_i}{\mu^c_i+\Lambda} < \frac{\mu^r_j}{\mu^c_j+\Lambda} + \frac{2(\eta+\Lambda+1)\sqrt{\frac{\ln(t)}{\tau_{jt}}}}{(\eta+\Lambda)(\overline{c}_{jt}+\Lambda)}$. This follows from the following derivation, which shows that if the original event is to occur and events (i) and (ii) do not hold, then event (iii) must occur,

$$\frac{\mu_i^r}{\mu_i^c+\Lambda} < \frac{\overline{r}_{it}}{\overline{c}_{it}+\Lambda} + \frac{(\eta+\Lambda+1)\sqrt{\frac{\ln(t)}{\tau_{it}}}}{(\eta+\Lambda)(\overline{c}_{it}+\Lambda)} \leq \frac{\overline{r}_{jt}}{\overline{c}_{jt}+\Lambda} + \frac{(\eta+\Lambda+1)\sqrt{\frac{\ln(t)}{\tau_{jt}}}}{(\eta+\Lambda)(\overline{c}_{jt}+\Lambda)} < \frac{\mu_j^r}{\mu_j^c+\Lambda} + \frac{2(\eta+\Lambda+1)\sqrt{\frac{\ln(t)}{\tau_{jt}}}}{(\eta+\Lambda)(\overline{c}_{jt}+\Lambda)},$$

where the first inequality holds when (i) is not satisfied, the second inequality is the original event we analyze, and the third inequality holds when (ii) is not satisfied. Therefore, we can now bound the probability of events (i), (ii), and (iii) separately. For event (i) to occur we need that $\overline{r}_{it}-\mu_i^r \leq -\sqrt{\frac{\ln(t)}{\tau_{it}}}$ and/or $\overline{c}_{it}-\mu_i^c \geq \sqrt{\frac{\ln(t)}{\tau_{it}}}$, because otherwise event (i) is contradicted as follows,

$$\begin{aligned}\frac{\overline{r}_{it}}{\overline{c}_{it}+\Lambda} - \frac{\mu_i^r}{\mu_i^c+\Lambda} &= \frac{\overline{r}_{it}(\mu_i^c+\Lambda)-\mu_i^r(\overline{c}_{it}+\Lambda)}{(\mu_i^c+\Lambda)(\overline{c}_{it}+\Lambda)} = \frac{\overline{r}_{it}(\mu_i^c+\Lambda)-\mu_i^r(\mu_i^c+\Lambda)+\mu_i^r(\mu_i^c+\Lambda)-\mu_i^r(\overline{c}_{it}+\Lambda)}{(\mu_i^c+\Lambda)(\overline{c}_{it}+\Lambda)} \\ &= \frac{(\mu_i^c+\Lambda)(\overline{r}_{it}-\mu_i^r)-\mu_i^r(\overline{c}_{it}-\mu_i^c)}{(\mu_i^c+\Lambda)(\overline{c}_{it}+\Lambda)} > \frac{-(\mu_i^c+\Lambda)\sqrt{\frac{\ln(t)}{\tau_{it}}}-\mu_i^r\sqrt{\frac{\ln(t)}{\tau_{it}}}}{(\mu_i^c+\Lambda)(\overline{c}_{it}+\Lambda)} \\ &= \frac{-\sqrt{\frac{\ln(t)}{\tau_{it}}}}{\overline{c}_{it}+\Lambda} + \frac{-\mu_i^r\sqrt{\frac{\ln(t)}{\tau_{it}}}}{(\mu_i^c+\Lambda)(\overline{c}_{it}+\Lambda)} \geq \frac{-\sqrt{\frac{\ln(t)}{\tau_{it}}}}{\overline{c}_{it}+\Lambda} + \frac{-\sqrt{\frac{\ln(t)}{\tau_{it}}}}{(\eta+\Lambda)(\overline{c}_{it}+\Lambda)} = -\frac{(\eta+\Lambda+1)\sqrt{\frac{\ln(t)}{\tau_{it}}}}{(\eta+\Lambda)(\overline{c}_{it}+\Lambda)}.\end{aligned}$$

Thus, we can bound the probability of event (i) using Lemma 2.5.2,

$$\mathbb{P}\left(\frac{\overline{r}_{it}}{\overline{c}_{it}+\Lambda} \leq \frac{\mu_i^r}{\mu_i^c+\Lambda} - \frac{(\eta+\Lambda+1)\sqrt{\frac{\ln(t)}{\tau_{it}}}}{(\eta+\Lambda)(\overline{c}_{it}+\Lambda)}\right) \leq \mathbb{P}\left(\overline{r}_{it}-\mu_i^r \leq -\sqrt{\frac{\ln(t)}{\tau_{it}}}\right) + \mathbb{P}\left(\overline{c}_{it}-\mu_i^c \geq \sqrt{\frac{\ln(t)}{\tau_{it}}}\right) \leq \frac{2}{t^2}.$$

Similarly, whenever event (ii) holds either $\overline{r}_{jt}-\mu_j^r \geq \sqrt{\frac{\ln(t)}{\tau_{jt}}}$ and/or $\overline{c}_{jt}-\mu_j^c \leq -\sqrt{\frac{\ln(t)}{\tau_{jt}}}$ must hold, because otherwise event (ii) is contradicted as follows,

$$\begin{aligned}\frac{\overline{r}_{jt}}{\overline{c}_{jt}+\Lambda} - \frac{\mu_j^r}{\mu_j^c+\Lambda} &= \frac{\overline{r}_{jt}(\mu_j^c+\Lambda)-\mu_j^r(\overline{c}_{jt}+\Lambda)}{(\mu_j^c+\Lambda)(\overline{c}_{jt}+\Lambda)} = \frac{\overline{r}_{jt}(\mu_j^c+\Lambda)-\mu_j^r(\mu_j^c+\Lambda)+\mu_j^r(\mu_j^c+\Lambda)-\mu_j^r(\overline{c}_{jt}+\Lambda)}{(\mu_j^c+\Lambda)(\overline{c}_{jt}+\Lambda)} \\ &= \frac{(\mu_j^c+\Lambda)(\overline{r}_{jt}-\mu_j^r)-\mu_j^r(\overline{c}_{jt}-\mu_j^c)}{(\mu_j^c+\Lambda)(\overline{c}_{jt}+\Lambda)} < \frac{(\mu_j^c+\Lambda)\sqrt{\frac{\ln(t)}{\tau_{jt}}}+\mu_j^r\sqrt{\frac{\ln(t)}{\tau_{jt}}}}{(\mu_j^c+\Lambda)(\overline{c}_{jt}+\Lambda)} \\ &= \frac{\sqrt{\frac{\ln(t)}{\tau_{jt}}}}{\overline{c}_{jt}+\Lambda} + \frac{\mu_j^r\sqrt{\frac{\ln(t)}{\tau_{jt}}}}{(\mu_j^c+\Lambda)(\overline{c}_{jt}+\Lambda)} \leq \frac{\sqrt{\frac{\ln(t)}{\tau_{jt}}}}{\overline{c}_{jt}+\Lambda} + \frac{\sqrt{\frac{\ln(t)}{\tau_{jt}}}}{(\eta+\Lambda)(\overline{c}_{jt}+\Lambda)} = \frac{(\eta+\Lambda+1)\sqrt{\frac{\ln(t)}{\tau_{jt}}}}{(\eta+\Lambda)(\overline{c}_{jt}+\Lambda)}.\end{aligned}$$

Again, we use Lemma 2.5.2 to bound the probability of event (ii),

$$\mathbb{P}\left(\frac{\mu_j^r}{\mu_j^c+\Lambda} + \frac{(\eta+\Lambda+1)\sqrt{\frac{\ln(t)}{\tau_{jt}}}}{(\eta+\Lambda)(\overline{c}_{jt}+\Lambda)} \leq \frac{\overline{r}_{jt}}{\overline{c}_{jt}+\Lambda}\right) \leq \mathbb{P}\left(\overline{r}_{jt}-\mu_j^r \geq \sqrt{\frac{\ln(t)}{\tau_{jt}}}\right) + \mathbb{P}\left(\overline{c}_{jt}-\mu_j^c \leq -\sqrt{\frac{\ln(t)}{\tau_{jt}}}\right) \leq \frac{2}{t^2}.$$

However, event (iii) can not happen when $\tau_{jt} \geq \left(\frac{2\eta+2\Lambda+2}{\left(\frac{\mu_i^r}{\mu_i^c+\Lambda}-\frac{\mu_j^r}{\mu_j^c+\Lambda}\right)(\eta+\Lambda)(\overline{c}_{jt}+\Lambda)}\right)^2 \ln(t) = L_{ijt}$, because

$$\begin{aligned}
\frac{\mu_i^r}{\mu_i^c+\Lambda} < \frac{\mu_j^r}{\mu_j^c+\Lambda} + \frac{2\left(\eta+\Lambda+1\right)\sqrt{\frac{\ln(t)}{\tau_{jt}}}}{(\eta+\Lambda)(\overline{c}_{jt}+\Lambda)} &\Leftrightarrow \left(\frac{\mu_i^r}{\mu_i^c+\Lambda} - \frac{\mu_j^r}{\mu_j^c+\Lambda}\right)(\eta+\Lambda)(\overline{c}_{jt}+\Lambda) < 2\left(\eta+\Lambda+1\right)\sqrt{\frac{\ln(t)}{\tau_{jt}}} \\
&\Leftrightarrow \left(\frac{\mu_i^r}{\mu_i^c+\Lambda} - \frac{\mu_j^r}{\mu_j^c+\Lambda}\right)^2 (\eta+\Lambda)^2(\overline{c}_{jt}+\Lambda)^2 < (2\eta+2\Lambda+2)^2 \frac{\ln(t)}{\tau_{jt}} \\
&\Leftrightarrow \tau_{jt} < \left(\frac{2\eta+2\Lambda+2}{\left(\frac{\mu_i^r}{\mu_i^c+\Lambda}-\frac{\mu_j^r}{\mu_j^c+\Lambda}\right)(\eta+\Lambda)(\overline{c}_{jt}+\Lambda)}\right)^2 \ln(t) = L_{ijt}.
\end{aligned}$$

Hence, the probability of picking arm j over the better arm $i < j$ is bounded by the following,

$$\begin{aligned}
\mathbb{P}(x_{it}^{\Pi} = 0 \cap x_{jt}^{\Pi} = 1 \cap \tau_{jt} \geq L_{ijt}) &\leq \mathbb{P}\left(\frac{\overline{r}_{it}+\Gamma_{it}}{\overline{c}_{it}+\Lambda+\Delta_{it}} \leq \frac{\overline{r}_{jt}+\Gamma_{jt}}{\overline{c}_{jt}+\Lambda+\Delta_{jt}} \cap \tau_{jt} \geq L_{ijt}\right) \\
&= \mathbb{P}\left(\frac{\overline{r}_{it}}{\overline{c}_{it}+\Lambda} + \frac{(\eta+\Lambda+1)\sqrt{\frac{\ln(t)}{\tau_{it}}}}{(\eta+\Lambda)(\overline{c}_{it}+\Lambda)} \leq \frac{\overline{r}_{jt}}{\overline{c}_{jt}+\Lambda} + \frac{(\eta+\Lambda+1)\sqrt{\frac{\ln(t)}{\tau_{jt}}}}{(\eta+\Lambda)(\overline{c}_{jt}+\Lambda)} \cap \tau_{jt} \geq L_{ijt}\right) \\
&\leq \mathbb{P}\left(\frac{\overline{r}_{it}}{\overline{c}_{it}+\Lambda} + \frac{(\eta+\Lambda+1)\sqrt{\frac{\ln(t)}{\tau_{it}}}}{(\eta+\Lambda)(\overline{c}_{it}+\Lambda)} \leq \frac{\mu_i^r}{\mu_i^c+\Lambda}\right) \\
&+ \mathbb{P}\left(\frac{\mu_j^r}{\mu_j^c+\Lambda} + \frac{(\eta+\Lambda+1)\sqrt{\frac{\ln(t)}{\tau_{jt}}}}{(\eta+\Lambda)(\overline{c}_{jt}+\Lambda)} \leq \frac{\overline{r}_{jt}}{\overline{c}_{jt}+\Lambda}\right) \\
&+ \mathbb{P}\left(\frac{\mu_i^r}{\mu_i^c+\Lambda} < \frac{\mu_j^r}{\mu_j^c+\Lambda} + \frac{2\left(\eta+\Lambda+1\right)\sqrt{\frac{\ln(t)}{\tau_{jt}}}}{(\eta+\Lambda)(\overline{c}_{jt}+\Lambda)} \cap \tau_{jt} \geq L_{ijt}\right) \\
&\leq \frac{2}{t^2} + \frac{2}{t^2} + 0 = \frac{4}{t^2}.
\end{aligned}$$

A.1.2 Proof of Lemma 2.5.4

For $x_{1t}^{\Pi} = \ldots = x_{jt}^{\Pi} = 1$ where $h < j \leq n$ to hold, we need that $\sum_{i=1}^{j}(\overline{c}_{it}+\Lambda+\Delta_{it}) \leq B$ because otherwise the algorithm cannot pick all the first j arms. This implies that we can bound the probability of interest by using the smallest possible j,

$$\mathbb{P}\left(x_{1t}^{\Pi} = \ldots = x_{jt}^{\Pi} = 1\right) \leq \mathbb{P}\left(\sum_{i=1}^{j}(\overline{c}_{it}+\Lambda+\Delta_{it}) \leq B\right) \leq \mathbb{P}\left(\sum_{i=1}^{h+1}(\overline{c}_{it}+\Lambda+\Delta_{it}) \leq B\right).$$

Recall that solving (RP_{Λ}) guarantees that $\sum_{i=1}^{h}(\mu_i^c+\Lambda) \leq B < \sum_{i=1}^{h+1}(\mu_i^c+\Lambda) - \epsilon$ for some $\epsilon > 0$ that can be viewed as the added cost of the $h+1$'th arm that disallows us from pulling that arm. We use this fact as well as the definition of Δ_{it} to obtain the following,

$$\begin{aligned}
\mathbb{P}\left(\sum_{i=1}^{h+1}(\overline{c}_{it}+\Lambda+\Delta_{it})\leq B\right) &\leq \mathbb{P}\left(\sum_{i=1}^{h+1}(\overline{c}_{it}+\Lambda+\Delta_{it})\leq \sum_{i=1}^{h+1}(\mu_i^c+\Lambda)-\epsilon\right)\\
&= \mathbb{P}\left(\sum_{i=1}^{h+1}(\overline{c}_{it}-\mu_i^c)\leq \sum_{i=1}^{h+1}\left(-\Delta_{it}-\frac{\epsilon}{h+1}\right)\right)\\
&= \mathbb{P}\left(\sum_{i=1}^{h+1}(\overline{c}_{it}-\mu_i^c)\leq -\sum_{i=1}^{h+1}\sqrt{\frac{\ln(t)}{\tau_{it}}}+\sum_{i=1}^{h+1}\left(\min\{\sqrt[4]{\frac{1}{\tau_{it}}},\Lambda\}-\frac{\epsilon}{h+1}\right)\right).
\end{aligned}$$

If $\frac{(h+1)^4}{\epsilon^4}\leq \max\{\frac{1}{\Lambda^4},\tau_{it}\}$ holds, then we have $\min\{\sqrt[4]{\frac{1}{\tau_{it}}},\Lambda\}\leq \frac{\epsilon}{h+1}$, which results in

$$\mathbb{P}\left(\sum_{i=1}^{h+1}(\overline{c}_{it}+\Lambda+\Delta_{it})\leq B\right)\leq \mathbb{P}\left(\sum_{i=1}^{h+1}(\mu_i^c-\overline{c}_{it})\geq \sum_{i=1}^{h+1}\sqrt{\frac{\ln(t)}{\tau_{it}}}\right).$$

By choosing $\epsilon=\sum_{i=1}^{h+1}(\mu_i^c+\Lambda)-B$, we have that the above inequality holds when for $i=1,\ldots,h+1$ we have $\frac{(h+1)^4}{(\sum_{i=1}^{h+1}(\mu_i^c+\Lambda)-B)^4}\leq \tau_{it}$. When this takes place, since the right-hand side is positive, we can take an approach similar to that used in Hoeffding's inequality. Let $\theta\geq 0$ and apply Markov's inequality,

$$\begin{aligned}
\mathbb{P}\left(\sum_{i=1}^{h+1}(\mu_i^c-\overline{c}_{it})\geq \sum_{i=1}^{h+1}\sqrt{\frac{\ln(t)}{\tau_{it}}}\right) &= \mathbb{P}\left(e^{\theta\sum_{i=1}^{h+1}(\mu_i^c-\overline{c}_{it})}\geq e^{\theta\sum_{i=1}^{h+1}\sqrt{\frac{\ln(t)}{\tau_{it}}}}\right)\\
&\leq e^{-\theta\sum_{i=1}^{h+1}\sqrt{\frac{\ln(t)}{\tau_{it}}}}\,\mathbb{E}\left[e^{\theta\sum_{i=1}^{h+1}(\mu_i^c-\overline{c}_{it})}\right].
\end{aligned}$$

Now, we use the independence of realized costs (c_{it}) between targets and periods,

$$\begin{aligned}
e^{-\theta\sum_{i=1}^{h+1}\sqrt{\frac{\ln(t)}{\tau_{it}}}}\,\mathbb{E}\left[e^{\theta\sum_{i=1}^{h+1}(\mu_i^c-\overline{c}_{it})}\right] &= e^{-\theta\sum_{i=1}^{h+1}\sqrt{\frac{\ln(t)}{\tau_{it}}}}\prod_{i=1}^{h+1}\prod_{s<t:x_{is}=1}\mathbb{E}\left[e^{\frac{\theta}{\tau_{it}}(\mu_i^c-c_{is})}\right]\\
&\leq e^{-\theta\sum_{i=1}^{h+1}\sqrt{\frac{\ln(t)}{\tau_{it}}}}\prod_{i=1}^{h+1}\prod_{s<t:x_{is}=1}e^{\frac{\theta^2}{8\tau_{it}^2}}=e^{-\theta\sum_{i=1}^{h+1}\sqrt{\frac{\ln(t)}{\tau_{it}}}+\frac{\theta^2}{8}\sum_{i=1}^{h+1}\frac{1}{\tau_{it}}},
\end{aligned}$$

where in the first inequality we use Hoeffding's lemma and the fact that $\mu_i^c-1\leq \mu_i^c-c_{is}\leq \mu_i^c$ for all arms i and time periods s. We can select $\theta=\frac{4\sum_{i=1}^{h+1}\sqrt{\frac{\ln(t)}{\tau_{it}}}}{\sum_{i=1}^{h+1}\frac{1}{\tau_{it}}}$ to minimize the bound, which yields,

$$\mathbb{P}\left(x_{1t}^{\Pi}=\ldots=x_{jt}^{\Pi}=1\right)\leq \mathbb{P}\left(\sum_{i=1}^{h+1}(\overline{c}_{it}+\Lambda+\Delta_{it})\leq B\right)\leq e^{-\frac{2\left(\sum_{i=1}^{h+1}\sqrt{\frac{\ln(t)}{\tau_{it}}}\right)^2}{\sum_{i=1}^{h+1}\frac{1}{\tau_{it}}}}\leq e^{-\frac{2\sum_{i=1}^{h+1}\frac{\ln(t)}{\tau_{it}}}{\sum_{i=1}^{h+1}\frac{1}{\tau_{it}}}}=\frac{1}{t^2}.$$

A.1.3 Proof of Lemma 2.5.5

First recall that the ORL algorithm picks at least one item at each time period. In fact, it is essential to do so, otherwise as t increases τ_{it}'s remain the same which causes Δ_{it}'s to increase. Moreover, as shown in Lemma 2.5.3 we have $\frac{\bar{r}_{it}+\Gamma_{it}}{\bar{c}_{it}+\Lambda+\Delta_{it}} = \frac{\bar{r}_{it}}{\bar{c}_{it}+\Lambda} + \frac{(\eta+\Lambda+1)\sqrt{\frac{\ln(t)}{\tau_{it}}}}{(\eta+\Lambda)(\bar{c}_{it}+\Lambda)}$, and therefore, if no item is picked at a time period t, the best item remains the same for period $t+1$. Since Δ_{it} of the best item increases at time period $t+1$, we can again not pick any item at period $t+1$ and the ORL algorithm falls into a vicious cycle. We can assume the single item picked by the algorithm is one of the optimal arms as the lemma assumes $x^{\Pi}_{h+1,t} = \cdots = x^{\Pi}_{nt} = 0$. That is, neither item i nor any item $j > h$ is picked by the algorithm while at least one optimal arm $k \leq h$ is selected.

The ORL algorithm could not pick arm i. This means that the total cost of the picked arms plus the cost of the last arm considered to be picked (not necessarily item i) was more than the budget B. Let X be the set of arms picked by the ORL as well as the arm that made the algorithm halt (let us call it arm k', where k' might or might not be the same as i). There exist three possibilities here, i) arm k' that caused the ORL to halt is suboptimal, i.e., $k' > h$, or ii) some arm l such that $i < l \leq h$ is picked or considered to be picked by the ORL (i.e., $\exists i < l \leq h, l \in X$), or, iii) all arms in X have indices less than i, or equivalently, $\forall l \in X, \frac{\mu^r_l}{\mu^c_l+\Lambda} \geq \frac{\mu^r_i}{\mu^c_i+\Lambda}$. Hence,

$$\begin{aligned}
\mathbb{P}\left(x^{\Pi}_{it} = x^{\Pi}_{h+1,t} = \ldots = x^{\Pi}_{nt} = 0\right) &= \mathbb{P}(x^{\Pi}_{it} = 0 \cap \forall j > h,\ x^{\Pi}_{jt} = 0 \cap \exists i < l \leq h,\ l \in X \text{ or } k' > h) \\
&\quad + \mathbb{P}(x^{\Pi}_{it} = 0 \cap \forall j > h.\ x^{\Pi}_{jt} = 0 \cap \nexists i < l \leq h,\ l \in X) \\
&\leq \mathbb{P}(x^{\Pi}_{it} = 0 \cap \exists l > i,\ x^{\Pi}_{lt} = 1) + \mathbb{P}(x^{\Pi}_{it} = 0 \cap X \subseteq \{1, \ldots, i\}) \\
&\leq \sum_{l=i+1}^{n} \mathbb{P}(x^{\Pi}_{it} = 0 \cap x^{\Pi}_{lt} = 1) + \mathbb{P}(x^{\Pi}_{it} = 0 \cap X \subseteq \{1, \ldots, i\}).
\end{aligned}$$

Note that k' is the arm that made the algorithm halt due to lack of budget, hence $x_{k't} = 0$; however, for the ease of communication in the above inequalities, we abused the notation and used $x_{k't} = 1$ to express that the ORL was about to pick k', if the budget constraint allowed. By Lemma 2.5.3, we have $\mathbb{P}(x^{\Pi}_{it} = 0 \cap x^{\Pi}_{lt} = 1 \cap \tau_{lt} \geq L_{ilt}) \leq \frac{4}{t^2}$ for any $l > i$. Therefore,

$$\mathbb{P}(x^{\Pi}_{it} = x^{\Pi}_{h+1,t} = \ldots = x^{\Pi}_{nt} = 0 \cap \forall l > i,\ \tau_{lt} \geq L_{ilt}) \leq \sum_{l=i+1}^{n} \frac{4}{t^2} + \mathbb{P}(x^{\Pi}_{it} = 0 \cap X \subseteq \{1, \ldots, i\}).$$

Now consider the case that $X \subseteq \{1, \ldots, i\}$. Since only optimal arms are in X, we have $|X| \leq h$. Again two possibilities are present here: i) there exists some $1 \leq l \leq i$ such that all items indexed from 1 to l are in X and nothing else, i.e., $X = \{1, 2, \ldots, l\}$, and, ii) there exists $l_1 < l_2 \leq i$ such that $l_1 \notin X$ and $l_2 \in X$. Let us denote the maximum index in X by $\max(X)$. Since $X \subseteq \{1, \ldots, i\}$ we have $\max(X) \leq i$. Hence,

$$\begin{aligned}\mathbb{P}(x_{it}^{\Pi} = 0 \cap X \subseteq \{1, \ldots, i\}) &= \mathbb{P}(x_{it}^{\Pi} = 0 \cap X \subseteq \{1, \ldots, i\} \cap \forall 1 \leq l \leq \max(X),\ l \in X) \\ &\quad + \mathbb{P}(x_{it}^{\Pi} = 0 \cap X \subseteq \{1, \ldots, i\} \cap \exists 1 \leq l < \max(X),\ l \notin X) \\ &\leq \sum_{m=1}^{i} \mathbb{P}(x_{it}^{\Pi} = 0 \cap X \subseteq \{1, \ldots, i\} \cap \forall 1 \leq l \leq m,\ l \in X) \\ &\quad + \sum_{m=1}^{i} \mathbb{P}(x_{it}^{\Pi} = 0 \cap X \subseteq \{1, \ldots, i\} \cap \exists 1 \leq l < m,\ l \notin X).\end{aligned}$$

In case ii), there exists an optimal item $l \notin X$ (i.e., $x_{lt}^{\Pi} = 0$) that has smaller index than at least one item X, in particular m. We thus have $\mathbb{P}(x_{it}^{\Pi} = 0 \cap X \subseteq \{1, \ldots, i\} \cap \exists 1 \leq l < m,\ l \notin X) \leq \sum_{l=1}^{m-1} \mathbb{P}(x_{it}^{\Pi} = 0 \cap x_{lt}^{\Pi} = 0 \cap x_{mt}^{\Pi} = 1)$. Consequently,

$$\begin{aligned}\mathbb{P}(x_{it}^{\Pi} = 0 \cap X \subseteq \{1, \ldots, i\}) &\leq \sum_{m=1}^{i} \mathbb{P}(x_{it}^{\Pi} = 0 \cap X \subseteq \{1, \ldots, i\} \cap \forall 1 \leq l \leq m,\ l \in X) \\ &\quad + \sum_{m=1}^{i} \sum_{l=1}^{m-1} \mathbb{P}(x_{it}^{\Pi} = 0 \cap x_{lt}^{\Pi} = 0 \cap x_{mt}^{\Pi} = 1) \\ &\leq \sum_{m=1}^{i} \mathbb{P}(x_{it}^{\Pi} = 0 \cap X \subseteq \{1, \ldots, i\} \cap \forall 1 \leq l \leq m,\ l \in X) \\ &\quad + \sum_{m=1}^{i} \sum_{l=1}^{m-1} \mathbb{P}(x_{lt}^{\Pi} = 0 \cap x_{mt}^{\Pi} = 1).\end{aligned}$$

By Lemma 2.5.3 we have, $\sum_{m=1}^{i} \sum_{l=1}^{m-1} \mathbb{P}(x_{lt} = 0 \cap x_{mt} = 1 \cap \tau_{mt} \geq L_{lmt}) \leq \sum_{m=1}^{i} \sum_{l=1}^{m-1} \frac{4}{t^2}$, thus,

$$\begin{aligned}\mathbb{P}(x_{it}^{\Pi} = x_{h+1,t}^{\Pi} = \ldots = x_{nt}^{\Pi} = 0 \cap \forall l > i,\ \tau_{lt} \geq L_{ilt} \cap \forall 1 \leq m \leq i,\ \tau_{mt} \geq \max_{1 \leq k < m} L_{kmt}) \leq \sum_{l=i+1}^{n} \frac{4}{t^2} \\ + \sum_{m=1}^{i} \sum_{k=1}^{m-1} \frac{4}{t^2} + \sum_{m=1}^{i} \mathbb{P}(x_{it}^{\Pi} = 0 \cap X \subseteq \{1, \ldots, i\} \cap \forall 1 \leq l \leq m,\ l \in X).\end{aligned}$$

The only case left to bound is when for some $m \leq i$ we have $X = \{1, 2, \ldots, m\}$. In what follows we upper bound the probability of this event. $X = \{1, 2, \ldots, m\}$ means that all the arms picked by the ORL algorithm including the arm that made it halt are indexed by $1, \dots, m$, that is, the sum of the estimated costs of these items is above the budget while the sum of their estimated costs except for the one that caused the halt is within the budget. Therefore,

$$
\begin{aligned}
\mathbb{P}(x_{it}^{\Pi} = 0 \cap X \subseteq \{1, \dots, i\} \cap \forall 1 \leq l \leq m,\ l \in X) &\leq \sum_{m=1}^{i} \mathbb{P}(X = \{1, \dots, m\}) \\
&\leq \sum_{m=1}^{i} \mathbb{P}\left(\sum_{l=1}^{m} (\bar{c}_{lt} + \Lambda + \Delta_{lt}) > B \cap \exists 1 \leq q \leq m. \sum_{l=1, l \neq q}^{m} (\bar{c}_{lt} + \Lambda + \Delta_{lt}) \leq B \right) \\
&\leq \sum_{m=1}^{i} \mathbb{P}\left(\sum_{l=1}^{m} (\bar{c}_{lt} + \Lambda + \Delta_{lt}) > B \right).
\end{aligned}
$$

By Lemma A.1.1, proven below, if $\forall 1 \leq l \leq m$ we have $\tau_{lt} \geq \max\{16 \ln(t)^2, 4\frac{\ln(t)}{\Lambda^2}\}$, then $\mathbb{P}(\sum_{l=1}^{m} (\bar{c}_{lt} + \Lambda + \Delta_{lt}) > B) \leq \frac{1}{t^2}$. Therefore, when $\forall l > i,\ \tau_{lt} \geq L_{ilt}$ and $\forall 1 \leq m \leq i,\ \tau_{mt} \geq \max_{1 \leq k \leq m}\{\max\{16 \ln(t)^2, 4\frac{\ln(t)}{\Lambda^2}\}, L_{kmt}\}$, we have the following nice property

$$
\mathbb{P}(x_{it}^{\Pi} = x_{h+1,t}^{\Pi} = \cdots = x_{nt}^{\Pi} = 0) \leq \sum_{l=i+1}^{n} \frac{4}{t^2} + \sum_{m=1}^{i} \sum_{l=1}^{m-1} \frac{4}{t^2} + \sum_{m=1}^{i} \frac{1}{t^2}.
$$

Lemma A.1.1. Let $\Delta_{it} = -\min\{\sqrt[4]{\frac{1}{\tau_{it}}}, \Lambda\} + \sqrt{\frac{\ln(t)}{\tau_{it}}}$, then

$$
\mathbb{P}\left(\sum_{i=1}^{h} (\bar{c}_{it} + \Lambda + \Delta_{it}) > B \cap \forall i \leq h, \tau_{it} \geq \max\{16 \ln(t)^2, 4\frac{\ln(t)}{\Lambda^2}\} \right) \leq \frac{1}{t^2}.
$$

Proof. We use the definition of Δ_{it} to obtain the following,

$$
\begin{aligned}
\mathbb{P}\left(\sum_{i=1}^{h} (\bar{c}_{it} + \Lambda + \Delta_{it}) > B \right) &\leq \mathbb{P}\left(\sum_{i=1}^{h} (\bar{c}_{it} + \Lambda + \Delta_{it}) > \sum_{i=1}^{h} (\mu_i^c + \Lambda) \right) = \mathbb{P}\left(\sum_{i=1}^{h} (\bar{c}_{it} - \mu_i^c) > \sum_{i=1}^{h} (-\Delta_{it}) \right) \\
&= \mathbb{P}\left(\sum_{i=1}^{h} (\bar{c}_{it} - \mu_i^c) > \sum_{i=1}^{h} \left(\min\{\sqrt[4]{\frac{1}{\tau_{it}}}, \Lambda\} - \sqrt{\frac{\ln(t)}{\tau_{it}}} \right) \right).
\end{aligned}
$$

Given that $\tau_{it} \geq \max\{16 \ln(t)^2, 4\frac{\ln(t)}{\Lambda^2}\}$ then we have $\min\{\sqrt[4]{\frac{1}{\tau_{it}}}, \Lambda\} - \sqrt{\frac{\ln(t)}{\tau_{it}}} > 0$, which allows us to take an approach similar to that used in Hoeffding's inequality. Let $\theta \geq 0$ and apply Markov's inequality,

$$
\begin{aligned}
\mathbb{P}\left(\sum_{i=1}^{h} (\bar{c}_{it} - \mu_i^c) > \sum_{i=1}^{h} (\min\{\sqrt[4]{\frac{1}{\tau_{it}}}, \Lambda\} - \sqrt{\frac{\ln(t)}{\tau_{it}}}) \right) &= \mathbb{P}\left(e^{\theta \sum_{i=1}^{h} (\bar{c}_{it} - \mu_i^c)} > e^{\theta \sum_{i=1}^{h} (\min\{\sqrt[4]{\frac{1}{\tau_{it}}}, \Lambda\} - \sqrt{\frac{\ln(t)}{\tau_{it}}})} \right) \\
&\leq e^{-\theta \sum_{i=1}^{h} \left(\min\{\sqrt[4]{\frac{1}{\tau_{it}}}, \Lambda\} - \sqrt{\frac{\ln(t)}{\tau_{it}}} \right)} \mathbb{E}\left[e^{\theta \sum_{i=1}^{h} (\bar{c}_{it} - \mu_i^c)} \right].
\end{aligned}
$$

Now, we use the independence of realized costs (c_{it}) between targets and periods,

$$e^{-\theta\sum_{i=1}^{h}(\min\{\sqrt[4]{\frac{1}{\tau_{it}}},\Lambda\}-\sqrt{\frac{\ln(t)}{\tau_{it}}})}\mathbb{E}\left[e^{\theta\sum_{i=1}^{h}(\overline{c}_{it}-\mu_i^c)}\right] = e^{-\theta\sum_{i=1}^{h}(\min\{\sqrt[4]{\frac{1}{\tau_{it}}},\Lambda\}-\sqrt{\frac{\ln(t)}{\tau_{it}}})}\prod_{i=1}^{h}\prod_{s<t:x_{is}=1}\mathbb{E}\left[e^{\frac{\theta}{\tau_{it}}(c_{is}-\mu_i^c)}\right]$$

$$\leq e^{-\theta\sum_{i=1}^{h}(\min\{\sqrt[4]{\frac{1}{\tau_{it}}},\Lambda\}-\sqrt{\frac{\ln(t)}{\tau_{it}}})}\prod_{i=1}^{h}\prod_{s<t:x_{is}=1}e^{\frac{\theta^2}{8\tau_{it}^2}}$$

$$= e^{-\theta\sum_{i=1}^{h}(\min\{\sqrt[4]{\frac{1}{\tau_{it}}},\Lambda\}-\sqrt{\frac{\ln(t)}{\tau_{it}}})+\frac{\theta^2}{8}\sum_{i=1}^{h}\frac{1}{\tau_{it}}},$$

where in the first inequality, we use Hoeffding's lemma and the fact that $-\mu_i^c \leq c_{is}-\mu_i^c \leq 1-\mu_i^c$ for all arms i and time periods s. We can select $\theta = \frac{4\sum_{i=1}^{h}\left(\min\{\sqrt[4]{\frac{1}{\tau_{it}}},\Lambda\}-\sqrt{\frac{\ln(t)}{\tau_{it}}}\right)}{\sum_{i=1}^{h}\frac{1}{\tau_{it}}}$ to minimize the bound, which yields,

$$\mathbb{P}\left(\sum_{i=1}^{h}(\overline{c}_{it}+\Lambda+\Delta_{it})>B\right) \leq e^{-\frac{2\left(\sum_{i=1}^{h}\left(\min\{\sqrt[4]{\frac{1}{\tau_{it}}},\Lambda\}-\sqrt{\frac{\ln(t)}{\tau_{it}}}\right)\right)^2}{\sum_{i=1}^{h}\frac{1}{\tau_{it}}}}$$

$$\leq e^{-\frac{2\sum_{i=1}^{h}\left(\min\{\sqrt[4]{\frac{1}{\tau_{it}}},\Lambda\}-\sqrt{\frac{\ln(t)}{\tau_{it}}}\right)^2}{\sum_{i=1}^{h}\frac{1}{\tau_{it}}}} \leq e^{-\frac{2\sum_{i=1}^{h}\frac{\ln(t)}{\tau_{it}}}{\sum_{i=1}^{h}\frac{1}{\tau_{it}}}} = \frac{1}{t^2},$$

where in the third inequality, we used the assumption that $\tau_{it} \geq \max\{16\ln(t)^2, 4\frac{\ln(t)}{\Lambda^2}\}$. Observe that if $\min\{\sqrt[4]{\frac{1}{\tau_{it}}},\Lambda\} = \Lambda$, then $\Lambda^2 - 2\Lambda\sqrt{\frac{\ln(t)}{\tau_{it}}} \geq 0$ when $\tau_{it} \geq 4\frac{\ln(t)}{\Lambda^2}$. Similarly, if $\min\{\sqrt[4]{\frac{1}{\tau_{it}}},\Lambda\} = \sqrt[4]{\frac{1}{\tau_{it}}}$, then $\sqrt{\frac{1}{\tau_{it}}} - 2\sqrt[4]{\frac{1}{\tau_{it}}}\sqrt{\frac{\ln(t)}{\tau_{it}}} \geq 0$ when $\tau_{it} \geq 16\ln(t)^2$. Hence,

$$\left(\min\{\sqrt[4]{\frac{1}{\tau_{it}}},\Lambda\}-\sqrt{\frac{\ln(t)}{\tau_{it}}}\right)^2 = \min\{\sqrt{\frac{1}{\tau_{it}}},\Lambda^2\} - 2\min\{\sqrt[4]{\frac{1}{\tau_{it}}},\Lambda\}\sqrt{\frac{\ln(t)}{\tau_{it}}} + \frac{\ln(t)}{\tau_{it}} \geq \frac{\ln(t)}{\tau_{it}}.$$

□

Appendix B

Appendix of Chapter 3

B.1 Proofs of Section 3.4

B.1.1 Proof of Lemma 3.4.6

To better understand our construction, we advise the reader to consult Figure B.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.
- 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 $(1 + \frac{1}{M}) \cdot M^{\Delta}$. 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

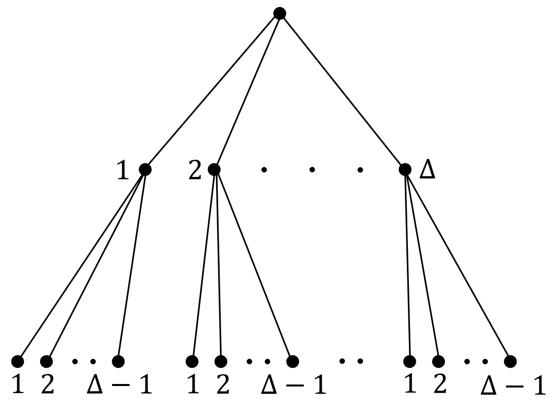


Figure B.1: A graph-based illustration of the tight example

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:

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

B.1.2 Proof of Lemma 3.4.8

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

B.2 Extension to Cross-Terms

B.2.1 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 in order 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:

$$
\begin{array}{llll}
(P_{\mathrm{CT}}) & \max & \displaystyle\sum_{t=1}^{T} \alpha_t \prod_{v \in V} B_{vt}^{x_{vt}} \prod_{u<v} B_{uvt}^{x_{ut}x_{vt}} & \\
& (C_1) & \displaystyle\sum_{t=1}^{T} x_{vt} \leq C_v & \forall\, v \in V \\
& (C_2) & \displaystyle\sum_{v \in V} x_{vt} \leq L_t & \forall\, t \in [T] \\
& (C_3) & x_{vt} \in \{0,1\} & \forall\, v \in V,\, t \in [T]
\end{array}
$$

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 3.4.

B.2.2 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 3.4.1, 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 3.4.1). 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 B.2.1. 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:

$$\begin{aligned} (P_{\text{CT}}) \quad & \max \quad \prod_{v \in N} B^{x_{v,1}} \cdot \prod_{(u,v) \in E} B^{-2n \cdot x_{u,1} x_{v,1}} & \\ (C_1) \quad & x_{v,1} \leq 1 & \forall\, v \in N \\ (C_2) \quad & \sum_{v \in N} x_{v,1} \leq n & \\ (C_3) \quad & x_{v,1} \in \{0,1\} & \forall\, v \in N \end{aligned}$$

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

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

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, $\text{OPT}(P_{\text{CT}}) \leq B^n \cdot B^{-2n} = B^{-n}$. 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:

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

□

B.2.3 Applicability of the Greedy Algorithm

A careful inspection of the greedy algorithm proposed in Section 3.4.2 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 non-negative 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.2.4 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 B.2.2. 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_{CT}) and in $1/\epsilon$.

2. (IP_ϵ) provides a $(1-\epsilon)$-approximation to $(P_{\rm CT})$. That is, $\mathrm{OPT}(IP_\epsilon) \geq (1-\epsilon)\cdot\mathrm{OPT}(P_{\rm CT})$, and moreover, any solution to (IP_ϵ) can be efficiently translated to $(P_{\rm CT})$ without any loss in optimality.

To avoid redundancies, since the basics of our approach are thoroughly discussed in Section 3.4.3, 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 non-negative real numbers, with $0 \in \mathcal{D}$. Based on this set, our integer program (IP_ϵ) is defined as follows:

$$
\begin{array}{lll}
(IP_\epsilon) \quad \max \displaystyle\sum_{t=1}^{T} \alpha_t \sum_{r\in\mathcal{D}} (e^r \cdot y_{tr}) & & \\
(C_1) \quad \displaystyle\sum_{t=1}^{T} x_{vt} \leq C_v & & \forall\, v \in V \\
(C_2) \quad \displaystyle\sum_{v\in V} x_{vt} \leq L_t & & \forall\, t \in [T] \\
(C_3) \quad \displaystyle\sum_{r\in\mathcal{D}} y_{tr} = 1 & & \forall\, t \in [T] \\
(C_4) \quad y_{tr} \leq \dfrac{1}{r}\left(\displaystyle\sum_{v\in V} \ln(B_{vt})\cdot x_{vt} + \sum_{u<v} \ln(B_{uvt})\cdot z_{uvt}\right) & & \forall\, t \in [T],\, r \in \mathcal{D}\setminus\{0\} \\
(C_5) \quad z_{uvt} \leq x_{ut},\, z_{uvt} \leq x_{vt},\, z_{uvt} \geq x_{ut} + x_{vt} - 1,\, z_{uvt} \geq 0 & & \forall\, u, v \in V,\, u<v,\, t \in [T] \\
(C_6) \quad x_{vt},\, y_{tr} \in \{0,1\} & & \forall\, u \in V,\, t \in [T],\, r \in \mathcal{D}
\end{array}
$$

As in Section 3.4.3, 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 $x_{ut}x_{vt}$, in order to linearize constraint (C_4). It is easy to verify that, for any binary assignment to the x-variables, constraint (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 3.4.3, we begin by initializing $\mathcal{D} = \{0\}$. This set is then augmented by all breakpoints that are created when the interval $[\frac{\epsilon}{2}, \Delta^2 \cdot \ln(B_{\max})] \subseteq (0, \infty)$ is geometrically partitioned by powers of $1+\frac{1}{M}$, where $M = \frac{\Delta^2}{\epsilon} \cdot \ln(B_{\max})$. 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. Theorem B.2.2, 1. In order to show that the size of (IP_ϵ) is polynomial in the input size of (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:

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

□

Proof. Theorem B.2.2, 2. To prove that $\text{OPT}(IP_\epsilon) \geq (1-\epsilon) \cdot \text{OPT}(P_{\text{CT}})$, letting x^* be a fixed optimal solution to (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 (IP_ϵ) with an objective value of at least $(1-\epsilon) \cdot \text{OPT}(P)$.

To this end, for every time period t, let

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

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 $\prod_{v \in V} B_{vt}^{x_{vt}} \prod_{u<v} B_{uvt}^{x_{ut} x_{vt}} \geq 1$ 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 (IP_ϵ): The constraints (C_1) and (C_2) are clearly satisfied, as they also appear in (P_{CT}); constraint (C_4) is guaranteed to be satisfied by the way we defined y; constraint (C_3) is taken care of by the fact that $0 \in \mathcal{D}$; and constraint (C_5) follows from the definition of z. Furthermore, the objective value of (x^*, y, z) with respect to (IP_ϵ) is precisely

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

In order to derive a lower bound on the latter term, we prove the following lemma.

Lemma B.2.3. For every time period t,

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

Proof. See Appendix B.2.5. □

As a result, we have just shown that

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

To conclude the proof of item 2, it remains to show that any feasible solution (x, y, z) to (IP_ϵ) can be efficiently translated to (P_{CT}) without any loss in optimality. Clearly, x must be a feasible solution to (P_{CT}), 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_{CT}) is:

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

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

B.2.5 Proof of Lemma B.2.3

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:

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

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

B.3 Additive Model

One can consider an alternative class of additive demand models, in the sense that the price and vehicle effects are additively separable. Such models can be expressed as:

$$d_t(p_t, \{x_{vt}\}_{v\in V}) = h_t^A(p_t) + \sum_{v\in V} B_{vt} x_{vt}. \tag{B.1}$$

The function $h_t^A(p_t)$ represents the effect of the price vector p_t on demand, and the boost parameter $B_{vt} \geq 0$ corresponds to the absolute increase in demand at time t when vehicle v is used. For example, if $B_{vt} = 65$, assigning vehicle v at time t yields an additional 65 units in sales, relative to the case where this vehicle is not used.

The objective function of the promotion vehicle scheduling problem for the additive model (B.1) asks to maximize total profit over the selling season:

$$\sum_{t=1}^{T}(p_t - c_t) \cdot d_t(p_t, \{x_{vt}\}_{v \in V}) = \sum_{t=1}^{T}(p_t - c_t) \cdot h_t^A(p_t) + \sum_{t=1}^{T} \alpha_t^A \sum_{v \in V} B_{vt} x_{vt}, \tag{B.2}$$

where $\alpha_t^A = p_t - c_t$. In this formulation, the first term on the right hand side does not affect the vehicle optimization problem. In addition, α_t^A corresponds to the effect of the price on profits at time t and is simply equal to the profit margin at time t. Since all prices and costs are assumed to be given, α_t^A is a given quantity as well.

The objective function (B.2), along with the linear constraints specified in Section 3.2, can easily be expressed as a bipartite b-matching problem, and therefore, can be solved efficiently by various methods. Consequently, the problem is scalable, with running times in milliseconds even for large instances. Nevertheless, as shown in Section 3.3.2, the additive demand model does not provide a good fit to the data and as a result, is not an appropriate model in this context. The additive model suffers from a scale independence due to assuming an absolute boost independent of the number of sales. In particular, different data points can have a wide range of sales units, such that the additive parameter for the promotion vehicle might be hard to estimate. Thus, a relative term such as in the multiplicative model seems to be more suitable. We refer the reader to Section 3.3.2, where we examine the prediction accuracy of both models and conclude that the multiplicative model is more suitable for our purposes.

B.4 PTAS for Uniform Boosts

In what follows, we devise a polynomial-time approximation scheme for the setting where all vehicles boosts are uniform, and where the base profits of all time periods are also uniform. In other words, the underlying assumption is that $B_{vt} = B$ for any vehicle v and time period t, and in addition, $\alpha_1 = \cdots = \alpha_T = 1$, after normalization. Specifically, for any accuracy level $\epsilon \in (0, 1/2]$, we show how to efficiently construct $\tilde{O}((T/\epsilon)^{O(1/\epsilon^2)})$ linear assignment problems, such that the optimal solution to at least one of them guarantees a $(1-\epsilon)$-approximation for the original problem. (To avoid cumbersome expressions, we use the notation $\tilde{O}(f(n)) = f(n) \cdot \text{polylog}(n)$.)

It is worth mentioning that our algorithm is also useful in the general case, where the vehicle boosts take arbitrary values, say in $[B_{\min}, B_{\max}]$, and similarly, the base profits take arbitrary values in $[\alpha_{\min}, \alpha_{\max}]$. It is not difficult to verify that, by replacing each of the boosts B_{vt} by

$B_{\min}$ and each of the base profits α_t by $\alpha_{\min}$, we scale down the optimal objective value by a factor of at most

$$\left(\frac{B_{\max}}{B_{\min}}\right)^{\Delta} \cdot \frac{\alpha_{\max}}{\alpha_{\min}} ,$$

where $\Delta = \max_t L_t$ is the maximum capacity of any time period. We can then apply our approximation scheme, losing an additional factor of $1 - \epsilon$, and obtaining a solution whose objective value with respect to the original boosts and base profits can only improve.

Profit classes. Let x^* be an optimal solution to (P), when the latter is specialized according to our assumptions above, that is,

$$\begin{array}{lll} (P) \quad \max & \displaystyle\sum_{t=1}^{T} B^{\sum_{v \in V} x_{vt}} & \\ (C_1) & \displaystyle\sum_{t=1}^{T} x_{vt} \leq C_v & \forall\, v \in V \\ (C_2) & \displaystyle\sum_{v \in V} x_{vt} \leq L_t & \forall\, t \in [T] \\ (C_3) & x_{vt} \in \{0,1\} & \forall\, v \in V,\, t \in [T] \end{array}$$

We assume from this point on that the value of $\mathrm{OPT} = \sum_{t=1}^{T} B^{\sum_{v \in V} x^*_{vt}}$ is known up to an ϵ-factor, meaning that we have an estimate $\widetilde{\mathrm{OPT}} \in [(1-\epsilon) \cdot \mathrm{OPT}, \mathrm{OPT}]$. This assumption can be enforced by employing the greedy algorithm (see Section 3.4.2), which is guaranteed to obtain an objective value $\mathcal{V}$ satisfying $\mathcal{V} \leq \mathrm{OPT} \leq (\Delta+1)\mathcal{V}$. We can then test all powers of $1+\epsilon$ within the interval $[\mathcal{V}, (\Delta+1)\mathcal{V}]$ as candidate values for $\widetilde{\mathrm{OPT}}$, run our algorithm with each value, and finally return the best solution found over all candidates.

Now, for purposes of analysis, consider a partition of the time periods $1, \ldots, T$ into classes, based on their associated profits with respect to x^*. Specifically, the class $\mathcal{C}_1$ consists of periods t with $B^{\sum_{v \in V} x^*_{vt}} \geq (1-\epsilon) \cdot \widetilde{\mathrm{OPT}}$. Then, for $2 \leq \ell \leq L$, each class $\mathcal{C}_\ell$ corresponds to periods with

$$B^{\sum_{v \in V} x^*_{vt}} \in \left[(1-\epsilon)^{\ell} \cdot \widetilde{\mathrm{OPT}}, (1-\epsilon)^{\ell-1} \cdot \widetilde{\mathrm{OPT}}\right) ,$$

where $L = \lceil \log_{1+\epsilon}(T/\epsilon) \rceil$. Finally, we define the last class C_{small} to consist of periods with $B^{\sum_{v \in V} x^*_{vt}} \in [0, (1-\epsilon)^{L} \cdot \widetilde{\mathrm{OPT}})$. By this definition, the total contribution of class C_{small} to the objective value of x^* is at most $\epsilon \cdot \widetilde{\mathrm{OPT}}$, since

$$\begin{aligned} \sum_{t \in C_{\text{small}}} B^{\sum_{v \in V} x^*_{vt}} &\leq |C_{\text{small}}| \cdot (1-\epsilon)^L \cdot \widetilde{\mathrm{OPT}} \\ &\leq T \cdot \left(\frac{1}{1+\epsilon} \right)^L \cdot \widetilde{\mathrm{OPT}} \\ &\leq T \cdot \left(\frac{1}{1+\epsilon} \right)^{\log_{1+\epsilon}(T/\epsilon)} \cdot \widetilde{\mathrm{OPT}} \\ &= \epsilon \cdot \widetilde{\mathrm{OPT}} \ . \end{aligned} \tag{B.3}$$

Constructing the linear assignment problem. For every $1 \leq \ell \leq L$, let R^*_ℓ be the total contribution of the time periods in $\mathcal{C}_\ell$ to the optimal objective function, i.e., $R^*_\ell = \sum_{t \in \mathcal{C}_\ell} B^{\sum_{v \in V} x^*_{vt}}$. Now suppose we have at our possession an integer-valued vector $(k_1, \ldots, k_L)$ that satisfies for every $1 \leq \ell \leq L$

$$k_\ell \cdot \frac{\epsilon \cdot \widetilde{\mathrm{OPT}}}{L} \leq R^*_\ell \leq (k_\ell + 1) \cdot \frac{\epsilon \cdot \widetilde{\mathrm{OPT}}}{L} \ , \tag{B.4}$$

letting $\tilde{R}_\ell = k_\ell \cdot \epsilon \cdot \widetilde{\mathrm{OPT}}/L$ be the resulting lower bound on R^*_ℓ. Obviously, the vector $(k_1, \ldots, k_L)$ is generally unknown, and we explain in the sequel how to enumerate over only polynomially-many vectors. (Note that a simple enumeration, where each of the coordinates k_ℓ takes one of the values $0, \ldots, T$, makes the total number of options $T^{O(L)} = T^{O((1/\epsilon) \cdot \log(T/\epsilon))}$, which is not polynomial in T.) However, for the time being, with this vector at hand, it follows that each class $\mathcal{C}_\ell$ contains at least $\lceil \tilde{R}_\ell / ((1-\epsilon)^{\ell-1} \cdot \widetilde{\mathrm{OPT}}) \rceil$ time periods, since

$$\tilde{R}_\ell \leq R^*_\ell = \sum_{t \in \mathcal{C}_\ell} B^{\sum_{v \in V} x^*_{vt}} \leq |\mathcal{C}_\ell| \cdot (1-\epsilon)^{\ell-1} \cdot \widetilde{\mathrm{OPT}} \ .$$

With this lower bound on $|\mathcal{C}_\ell|$, we create a (feasibility) linear assignment problem, with vehicles on the left and with time periods on the right. Each vehicle v has a supply of C_v units. On the other hand, for each class $\mathcal{C}_\ell$, we take a separate subset of $\lceil \tilde{R}_\ell / ((1-\epsilon)^{\ell-1} \cdot \widetilde{\mathrm{OPT}}) \rceil$ time periods, and associate each of them with a demand of m_ℓ units. Here, m_ℓ stands for the minimal number of vehicles needed in order to obtain a per-period profit within the interval corresponding to class $\mathcal{C}_\ell$. That is, m_ℓ is the minimal integer for which

$$B^{m_\ell} \in \left[(1-\epsilon)^\ell \cdot \widetilde{\mathrm{OPT}}, (1-\epsilon)^{\ell-1} \cdot \widetilde{\mathrm{OPT}} \right) \ . \tag{B.5}$$

Approximation guarantee. By the preceding discussion, the optimal solution x^* is in particular feasible for this assignment problem. As a result, by computing any feasible solution, we obtain an overall profit of at least

$$\begin{aligned}
\sum_{\ell=1}^{L} \left\lceil \frac{\tilde{R}_\ell}{(1-\epsilon)^{\ell-1} \cdot \widetilde{\mathrm{OPT}}} \right\rceil \cdot B^{m_\ell} &\geq \sum_{\ell=1}^{L} \frac{\tilde{R}_\ell}{(1-\epsilon)^{\ell-1} \cdot \widetilde{\mathrm{OPT}}} \cdot (1-\epsilon)^\ell \cdot \widetilde{\mathrm{OPT}} \\
&= (1-\epsilon) \cdot \sum_{\ell=1}^{L} \tilde{R}_\ell \\
&\geq (1-\epsilon) \cdot \sum_{\ell=1}^{L} \left(R_\ell^* - \frac{\epsilon \cdot \widetilde{\mathrm{OPT}}}{L} \right) \\
&= (1-\epsilon) \cdot \left(\sum_{\ell=1}^{L} R_\ell^* - \epsilon \cdot \widetilde{\mathrm{OPT}} \right) \\
&\geq (1-\epsilon) \cdot (1-2\epsilon) \cdot \widetilde{\mathrm{OPT}} \\
&\geq (1-2\epsilon)^3 \cdot \mathrm{OPT} \ .
\end{aligned}$$

The first inequality follows from the bounds on B^{m_ℓ} given in (B.5). The second inequality follows from the relation between $\tilde{R}_\ell$ and R_ℓ^* in (B.4). The third inequality holds since $\sum_{\ell=1}^{L} R_\ell^* \geq (1-\epsilon) \cdot \widetilde{\mathrm{OPT}}$, by the upper bound on the contribution of C_{small} in (B.3). The last inequality holds since $\widetilde{\mathrm{OPT}} \geq (1-\epsilon) \cdot \mathrm{OPT}$.

Efficient enumeration. It remains to show that the number of vectors $(k_1, \ldots, k_L)$ to be considered is polynomial in the input size, for any fixed $\epsilon \in (0, 1/2]$. For this purpose, note that by definition of k_ℓ,

$$\frac{\widetilde{\mathrm{OPT}}}{1-\epsilon} \geq \mathrm{OPT} \geq \sum_{\ell=1}^{L} R_\ell^* \geq \frac{\epsilon \cdot \widetilde{\mathrm{OPT}}}{L} \cdot \sum_{\ell=1}^{L} k_\ell \ ,$$

implying that $\sum_{\ell=1}^{L} k_\ell \leq L/(\epsilon(1-\epsilon)) \leq 2L/\epsilon$. Basic counting arguments show that the number of integer solutions to this inequality is $O(e^{O(L/\epsilon)}) = \tilde{O}((T/\epsilon)^{O(1/\epsilon^2)})$.

Appendix C

Appendix of Chapter 4

C.1 Proofs of Section 4.3

C.1.1 Proof of Theorem 4.3.1

To prove that (IQP) is NP-hard, we will reduce the maximal independent set problem $(MISP)$, which is NP-hard, to the operational vendor fund selection problem in (IQP). An instance of $(MISP)$ is specified by a graph $G = (V, E)$, where V is the set of vertices and E the set of edges. A subset of vertices $U \subseteq V$ is said to be independent if for every pair of vertices $u, v \in U$, there is no edge (u, v) connecting the two. The objective of $(MISP)$ is to compute an independent set of maximal cardinality, and can be formulated as follows:

$$\max_{y} \sum_{v \in V} y_v \tag{C.1a}$$

$$\text{s.t. } y_u + y_v \leq 1, \qquad \forall (u, v) \in E \tag{C.1b}$$

$$y_v \in \{0, 1\}, \qquad \forall v \in V \tag{C.1c}$$

For the reduction, given an instance of $(MISP)$ with $G = (V, E)$, we construct a corresponding instance of (IQP) as follows. First, consider a single item, meaning $N = 1$, with one promotional price, meaning $K = 1$. Let the index t of (IQP) be reset to $v \in V$ so that $T = |V|$, and reset the index j of (IQP) to $(u, v) \in E$ so that $J = |E|$. For each time period v, let the prices, costs, and demand function be defined as: $P_0 = 2$, $P_1 = 0$, $c_v = 0$, $d_v(P_0) = 0.5$, $d_v(P_1) = 0.5$. For each vendor fund (u, v), let the discount be $\delta_{(u,v)} = 1$, the pass-through be $\gamma_{(u,v)} = 0.5$, the vendor fund time periods are $\tau_{(u,v)} = \{u, v\}$, and the forward-buying time periods are $\theta_{(u,v)} = \emptyset$. Finally, let the other parameters be given by: $L = |V|$, $S = 0$, $Q = |E|$.

Altogether, we obtain the following instance of (IQP):

$$\max_{x,z} \sum_{v=1}^{|V|} x_{0v} + \sum_{(u,v)\in E} 2z_{(u,v)} \tag{C.2a}$$

$$\text{s.t.}\ x_{0v} + x_{0u} \leq 2 - z_{(u,v)} \qquad \forall (u,v) \in E \tag{C.2b}$$

$$x_{0v} \in \{0,1\}, z_{(u,v)} \in \{0,1\} \qquad \forall v \in \{1,\dots,|V|\}, \forall (u,v) \in E \tag{C.2c}$$

The original constraints (4.2d), (4.4), (4.5), and (4.2f) can be removed, as they are always satisfied under this business rule parameter setting. To finalize the reduction, we only need to prove that $z_{(u,v)} = 1, \forall (u,v) \in E$, in any optimal solution to this instance of (IQP). Once we have proven this, the above instance of (IQP) is equivalent to $(MISP)$ and we have shown the reduction.

For purposes of contradiction, suppose that x^*_{0v}, $\forall v \in V$, and $z^*_{(u,v)}$, $\forall (u,v) \in E$, is the optimal solution to this instance of (IQP) and some $z^*_{(u,v)} = 0$. Let E' be the set of $(u,v) \in E$ for which $z^*_{(u,v)} = 0$. Now, fix arbitrary $(u',v') \in E'$, then there are two cases, either $x^*_{0u'} + x^*_{0v'} \leq 1$, or $x^*_{0u'} + x^*_{0v'} = 2$.

In case $x^*_{0u'} + x^*_{0v'} \leq 1$, we create a new solution where $\widetilde{x}_{0u} = x^*_{0u}$, $\forall u \in V$, and $\widetilde{z}_{(u,v)} = z^*_{(u,v)}$, $\forall (u,v) \in E$, except that $\widetilde{z}_{(u',v')} = 1$. This new solution remains feasible, because $\widetilde{x}_{0u'} + \widetilde{x}_{0v'} = x^*_{0u'} + x^*_{0v'} \leq 1 = 2 - \widetilde{z}_{(u',v')}$ and the decision variables remain binary. Note that going from the original optimal solution to this solution has increased the objective by 2, which contradicts the optimality assumption.

In case $x^*_{0u'} + x^*_{0v'} = 2$, we construct a new solution where $\widetilde{x}_{0u} = x^*_{0u}$, $\forall u \in V$, and $\widetilde{z}_{(u,v)} = z^*_{(u,v)}$, $\forall (u,v) \in E$, except for $\widetilde{z}_{(u',v')} = 1$ and without loss of generality $\widetilde{x}_{0v'} = 0$. This new solution is feasible, because $\widetilde{x}_{0u'} + \widetilde{x}_{0v'} = x^*_{0u'} + 0 = 1 = 2 - \widetilde{z}_{(u',v')}$ while the decision variables are still binary. However, the objective solution increased by 1 in the newly constructed solution, contradicting the optimality assumption.

Thus, we have proven that in any optimal solution to the above instance of (IQP), $z_{(u,v)} = 1$, $\forall (u,v) \in E$. Hence, the optimal solution to $(MISP)$ can be found by solving this particular instance of (IQP), proving the reduction.

C.1.2 Proof of Theorem 4.3.3

First, we assess how well step 2a and 2b of the ILR algorithm approximate the optimal solution to (LRP) for a fixed λ. Besides proving a bound on the optimality of the solution after step 2a and 2b, the following lemma actually establishes a bound on the optimality of the solution after any number of iterations of step 2a and 2b.

Lemma C.1.1. For fixed λ, let (x^*, z^*) be the optimal solution to (LRP) and let (x_s, z_s) denote the approximate solution after iterating s times over step 2a and 2b for this fixed λ. Then,

$$\pi_R(x^*, z^*|\lambda) - \pi_R(x_s, z_s|\lambda) \leq D(\lambda),$$

where $D(\lambda) = \max_{G \subseteq \cup_{i=1}^N F_i : |G| \leq Q} \sum_{i=1}^N \sum_{j \in F_i \cap G} \left[\sum_{t \in \tau_j \cup \theta_j} \delta_j P_{i0} d_{it}(P_{iK}) + \sum_{t \in \tau_j} \lambda_{ij} \gamma_j P_{i0} \right]$.

Proof. See Appendix C.1.3. □

The bound from Lemma C.1.1 shows that, for any fixed λ, the ILR algorithm solution will be within at least $D(\lambda)$ of the optimal solution to (LRP), which we denote by (x^{LRP}, z^{LRP}),

$$\pi_R(x^{LRP}, z^{LRP}|\lambda) - \pi_R(x^{ILR}, z^{ILR}|\lambda) \leq D(\lambda).$$

According to weak duality, the optimal objective value of the maximization problem (IQP) can be at most the optimal objective value when minimizing (LRP) over λ,

$$\pi_R(x^*, z^*|\lambda) \leq \pi_R(x^{LRP}, z^{LRP}|\lambda).$$

Putting the two inequalities together, we get

$$\pi_R(x^*, z^*|\lambda) - \pi_R(x^{ILR}, z^{ILR}|\lambda) \leq \pi_R(x^{LRP}, z^{LRP} - \pi_R(x_s, z_s|\lambda) \leq D(\lambda).$$

Now, before running the algorithm, we cannot know at which λ the optimal solution lies, hence we maximize $D(\lambda) = \max_{G \subseteq \cup_{i=1}^N F_i : |G| \leq Q} \sum_{i=1}^N \sum_{j \in F_i \cap G} \left[\sum_{t \in \tau_j \cup \theta_j} \delta_j P_{i0} d_{it}(P_{iK}) + \sum_{t \in \tau_j} \lambda_{ij} \gamma_j P_{i0} \right]$ with respect to λ. Because $\gamma_j P_0 \geq 0$, this means that $\lambda_{ij} = 0, \forall i \in [N], j \in F_i$, establishing the a-priori bound as $D(0)$. On the other hand, when we run the algorithm we can use λ^{ILR} that gives the solution of the ILR algorithm.

C.1.3 Proof of Lemma C.1.1

For a fixed $\lambda \leq 0$, we want to bound the absolute difference between the objectives of (x^*, z^*), the optimal (LRP) solution, and (x_s, z_s), the approximate solution after s iterations of step 2a and 2b of the ILR algorithm. To prove the bound, we first add and subtract $\pi_R(x^*, z_s|\lambda)$ on the left-hand side of the inequality to obtain

$$\begin{aligned}
\pi_R(x^*, z^*|\lambda) - \pi_R(x_s, z_s|\lambda) &= \pi_R(x^*, z^*|\lambda) - \pi_R(x^*, z_s|\lambda) + \pi_R(x^*, z_s|\lambda) - \pi_R(x_s, z_s|\lambda) \\
&\leq \pi_R(x^*, z^*|\lambda) - \pi_R(x^*, z_s|\lambda),
\end{aligned}$$

as by definition of the algorithm $\pi_R(x^*, z_s|\lambda) - \pi_R(x_s, z_s|\lambda) \leq 0$. Next, we analyze this upper bound,

$$\begin{aligned}
\pi_R(x^*, z^*|\lambda) - \pi_R(x^*, z_s|\lambda) &= \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=0}^{K}(P_{ik} - w_i)\, d_{it}\,(P_{ik})\, x^*_{ikt} + \sum_{i=1}^{N}\sum_{j\in F_i} z^*_j \sum_{t\in\tau_j\cup\theta_j}\sum_{k=0}^{K}\delta_j P_{i0} d_{it}\,(P_{ik})\, x^*_{ikt} \\
&+ \sum_{i=1}^{N}\sum_{j\in F_i}\lambda_{ij}\sum_{t\in\tau_j}\left[\sum_{k=0}^{K} P_{ik}x^*_{ikt} + \gamma_j P_{i0} z^*_j - P_{i0}\right] \\
&- \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=0}^{K}(P_{ik} - w_i)\, d_{it}\,(P_{ik})\, x^*_{ikt} - \sum_{i=1}^{N}\sum_{j\in F_i} z_{s,j} \sum_{t\in\tau_j\cup\theta_j}\sum_{k=0}^{K}\delta_j P_{i0} d_{it}\,(P_{ik})\, x^*_{ikt} \\
&- \sum_{i=1}^{N}\sum_{j\in F_i}\lambda_{ij}\sum_{t\in\tau_j}\left[\sum_{k=0}^{K} P_{ik}x^*_{ikt} + \gamma_j P_{i0} z_{s,j} - P_{i0}\right] \\
&= \sum_{i=1}^{N}\sum_{j\in F_i}\left(z^*_j - z_{s,j}\right)\left[\sum_{t\in\tau_j\cup\theta_j}\sum_{k=0}^{K}\delta_j P_{i0} d_{it}\,(P_{ik})\, x^*_{ikt} + \sum_{t\in\tau_j}\lambda_{ij}\gamma_j P_{i0}\right] \\
&\leq \sum_{i=1}^{N}\sum_{\substack{j\in F_i:\\ z^*_j=1\\ z_{s,j}=0}}\left[\sum_{t\in\tau_j\cup\theta_j}\delta_j P_{i0} d_{it}\,(P_{iK}) + \sum_{t\in\tau_j}\lambda_{ij}\gamma_j P_{i0}\right] \\
&\leq \max_{G\subseteq\cup_{i=1}^{N}F_i:|G|\leq Q}\sum_{i=1}^{N}\sum_{j\in F_i\cap G}\left[\sum_{t\in\tau_j\cup\theta_j}\delta_j P_{i0} d_{it}(P_{iK}) + \sum_{t\in\tau_j}\lambda_{ij}\gamma_j P_{i0}\right].
\end{aligned}$$

The first inequality maximizes the difference by only considering those vendor funds selected in z^* and not in z_s, as well as selecting the largest possible demand in each time period. The second inequality selects the Q largest positive benefits.

C.2 Proofs of Section 4.4

C.2.1 Proof of Proposition 4.4.1

To establish when the retailer accepts or rejects a vendor fund (δ, γ), we analyze constraint (4.12b), which compares the retailer's profit under the vendor fund with the profit of the outside option. We plug in the optimal pricing policy,

$$
\begin{aligned}
&((1-\gamma)P_0 - w + \delta P_0)(a - b(1-\gamma)P_0) + \theta(P_0 - w + \delta P_0)(a - bP_0) \geq \overline{\pi} \\
\Leftrightarrow\ & \delta \geq \frac{\overline{\pi} - \theta(P_0 - w)(a - bP_0) - ((1-\gamma)P_0 - w)(a - b(1-\gamma)P_0)}{P_0(a(\theta+1) - b(\theta+1-\gamma)P_0)} \\
\Leftrightarrow\ & \delta \geq \frac{w}{P_0} + \frac{\overline{\pi} + \gamma P_0(a - b(1-\gamma)P_0)}{P_0(a(\theta+1) - b(\theta+1-\gamma)P_0)} - 1.
\end{aligned}
$$

This bound holds as long as γ is set such that the price in the vendor fund period is $(1-\gamma)P_0$, which according to equation (4.13) holds when

$$
\gamma \geq 1 - \frac{a + bw - b\delta P_0}{2bP_0} \Leftrightarrow \delta \leq \frac{w}{P_0} + \frac{a}{bP_0} - 2(1-\gamma).
$$

Hence, the above bound holds when

$$
\begin{aligned}
&\frac{w}{P_0} + \frac{\overline{\pi} + \gamma P_0(a - b(1-\gamma)P_0)}{P_0(a(\theta+1) - b(\theta+1-\gamma)P_0)} - 1 \geq \frac{w}{P_0} + \frac{a}{bP_0} - 2(1-\gamma) \\
\Leftrightarrow\ & \frac{\overline{\pi} + \gamma P_0(a - b(1-\gamma)P_0)}{P_0(a(\theta+1) - b(\theta+1-\gamma)P_0)} - \frac{a}{bP_0} + 1 - 2\gamma \geq 0 \\
\Leftrightarrow\ & \gamma \geq \frac{\sqrt{b\overline{\pi} + (a - bP_0)^2(\theta + \theta^2)} - (a - bP_0)(\theta+1)}{bP_0} = \overline{\gamma},
\end{aligned}
$$

and when $\gamma \leq \overline{\gamma}$ we can just plug in $\overline{\gamma}$ to form the following efficient frontier between acceptance and rejection:

$$
\delta \geq \begin{cases} \dfrac{w}{P_0} + \dfrac{\overline{\pi} + \overline{\gamma} P_0(a - b(1-\overline{\gamma})P_0)}{P_0(a(\theta+1) - b(\theta+1-\overline{\gamma})P_0)} - 1 & \text{if } 0 \leq \gamma \leq \overline{\gamma} \\ \dfrac{w}{P_0} + \dfrac{\overline{\pi} + \gamma P_0(a - b(1-\gamma)P_0)}{P_0(a(\theta+1) - b(\theta+1-\gamma)P_0)} - 1 & \text{if } \overline{\gamma} \leq \gamma \leq 1 \end{cases}.
$$

C.2.2 Proof of Lemma 4.4.2

In problem (4.14), the bilevel problem has been rewritten as a two-dimensional optimization problem with the efficient frontier that links δ and γ in constraint (4.14b) and a boundary condition in constraint (4.14c). Taking the derivative of the objective with respect to δ yields

$$\frac{\partial \pi_S(\delta,\gamma)}{\partial \delta} = -P_0(a(\theta+1) - b(\theta+1-\gamma)P_0) < 0 \Leftarrow a - bP_0 > 0,$$

which implies that the profit's derivative with respect to δ is negative if demand is positive at the regular price P_0. Hence, irrespective of γ, the derivative of the supplier's profit in δ is negative, which means that in the optimal solution we want to set δ as small as possible. Hence, the efficient frontier in constraints (4.14b) will be binding in the optimal solution.

C.2.3 Proof of Proposition 4.4.3

In order to solve problem (4.15), we differentiate the vendor fund profits with respect to γ, resulting in

$$\begin{aligned}
\frac{\partial \pi_S(\gamma)}{\partial \gamma} &= \frac{\partial((1-\gamma)P_0 - c)(a - b(1-\gamma)P_0)}{\partial \gamma} \\
&= \frac{\partial(aP_0 - a\gamma P_0 - ac - bP_0^2 + b\gamma P_0^2 - bcP_0 + b\gamma P_0^2 - b\gamma^2 P_0^2 - bc\gamma P_0)}{\partial \gamma} \\
&= -aP_0 + bP_0^2 + bP_0^2 - 2b\gamma P_0^2 - bcP_0 = 2b(1-\gamma)P_0^2 - (a+bc)P_0 \\
&\Rightarrow 2b(1-\gamma^*)P_0^2 - (a+bc)P_0 = 0 \Rightarrow \gamma^* = 1 - \frac{a+bc}{2bP_0},
\end{aligned}$$

which leads to the maximum as the profit is concave in γ, the second derivative with respect to γ is negative

$$\frac{\partial^2 \pi_S(\gamma)}{\partial \gamma^2} = \frac{\partial(2b(1-\gamma)P_0^2)}{\partial \gamma} = -2bP_0^2 < 0.$$

Plugging γ^* into the efficient frontier and optimal pricing policy

$$\delta^* = \frac{w}{P_0} + \frac{2\overline{\pi} + (P_0 - \frac{a}{2b} - \frac{c}{2})(a - bc)}{P_0(a - bc + 2\theta(a - bP_0))} - 1 \text{ and } (1-\gamma^*)P_0 = \frac{a+bc}{2b}.$$

C.2.4 Proof of Proposition 4.4.4

Regardless of whether the retailer is forward-buying or not, Proposition 4.4.3 shows that the supplier's optimal vendor fund profit is $\pi_S^* = \dfrac{(a-bc)^2}{4b} - \overline{\pi} + \theta(P_0 - c)(a - bP_0)$ when imposing pass-through. This means that the supplier is not impacted by forward-buying. However, we need to check whether vendor funds without a pass-through constraint also eliminate forward-buying. The case of no pass-through can be modeled by setting $\gamma = 0$ in problem (4.12).

In problem (4.12), γ only appears in constraint (4.12e). This means that setting $\gamma = 0$ changes the optimal pricing scheme to $\frac{a+bw-b\delta P_0}{2b}$. According to Proposition 4.4.1, constraint (4.12b) then becomes

$$\delta \geq \frac{w}{P_0} - \frac{a}{bP_0} + \frac{2\sqrt{b\overline{\pi} + (a-bP_0)^2(\theta+\theta^2)} - 2\theta(a-bP_0)}{bP_0} = \delta^B$$

Under the updated pricing scheme, objective (4.12a) is

$$\begin{aligned}\overline{\pi}_S(\delta) &= (w-c-\delta P_0)\left(a - b\cdot\frac{a+bw-b\delta P_0}{2b}\right) + \theta(w-c-\delta P_0)(a-bP_0)\\ &= \frac{1}{2}(w-c-\delta P_0)\left(a-b(w-\delta P_0)\right) + \theta(w-c-\delta P_0)(a-bP_0).\end{aligned}$$

The objective is quadratic, concave and has a single maximum, taking the derivative with respect to δ

$$\begin{aligned}\frac{d\overline{\pi}_S(\delta)}{d\delta} &= P_0\left(b(\theta P_0 + w - \delta P_0) - \frac{1}{2}a(2\theta+1) - \frac{1}{2}bc\right)\\ &\Rightarrow P_0\left(b(\theta P_0 + w - \delta^U P_0) - \frac{1}{2}a(2\theta+1) - \frac{1}{2}bc\right) = 0\\ &\Rightarrow \delta^U = \frac{w}{P_0} - \frac{a(2\theta+1)}{2bP_0} - \frac{c}{2P_0} + \theta.\end{aligned}$$

Due to constraint (4.12b) and concavity of the objective we know $\delta^* = \max\{\delta^U, \delta^B\}$. So the supplier's optimal profits are

$$\overline{\pi}_S^* = \begin{cases} \overline{\pi}_S(\delta^U) = \dfrac{(a(2\theta+1) - bc - 2\theta bP_0)^2}{8b} \\ \quad \text{if } \delta^U \geq \delta^B \\ \overline{\pi}_S(\delta^B) = \dfrac{(a(2\theta+1) - bc - 2\theta bP_0)\sqrt{b\overline{\pi} + (a-bP_0)^2(\theta+\theta^2)} - 2(b\overline{\pi} + (a-bP_0)^2(\theta+\theta^2)}{b} \\ \quad \text{if } \delta^B \geq \delta^U \end{cases}.$$

To conclude, because the supplier's optimal profit without pass-through under a non-forward-buying retailer and a forward-buying retailer is different, we observe that the supplier's optimal profit is impacted by the behavior of the retailer, while this is not the case with pass-through. Hence, imposing the pass-through constraint eliminates forward-buying.

C.2.5 Proof of Proposition 4.4.5

To show that the pass-through constrained vendor fund coordinates the supply chain, we have to show that the total optimal profits of the supplier and retailer combined equal what a single monopolist could earn in the centralized supply chain. In this case, the single monopolist would maximize the revenue of the product by setting the optimal price to solve:

$$\max_p \overline{\pi}_{SC}(p) = (p-c)(a-bp) + \theta(P_0 - c)(a - bP_0)$$

Solving this problem we get $p^* = \dfrac{a+bc}{2b}$ which implies that

$$\overline{\pi}^*_{SC} = \left(\frac{a+bc}{2b} - c\right)\left(a - b\cdot\frac{a+bc}{2b}\right) + \theta(P_0 - c)(a - bP_0) = \frac{(a-bc)^2}{4b} + \theta(P_0 - c)(a - bP_0) = \overline{\pi}^*_S + \overline{\pi}^*_R$$

Hence, the pass-through constrained vendor fund coordinates the supply chain, because together the supplier and retailer earn the optimal profit available in the centralized supply chain.

C.2.6 Proof of Lemma 4.4.6

In order to solve problem (4.16), we take the derivative with respect to δ, where $f(\cdot)$ is the pdf corresponding to $F(\cdot)$, resulting in

$$\begin{aligned}
&\frac{\partial \mathbb{E}[\overline{\pi}_S(\delta,\gamma)]}{\partial \delta} = -P_0(a(\theta+1) - b(\theta+1-\gamma)P_0)\cdot F\left(\overline{\pi}_R\right) \\
&+ \left[(w-c-\delta P_0)(a(\theta+1) - b(\theta+1-\gamma)P_0) - \frac{1}{2}(w-c)(a(2\theta+1) - b(2\theta P_0 + w))\right] \\
&\cdot P_0(a(\theta+1) - b(\theta+1-\gamma)P_0)\cdot f\left(\overline{\pi}_R\right) \\
&= \left[\left[(w-c-\delta P_0)(a(\theta+1) - b(\theta+1-\gamma)P_0) - \frac{1}{2}(w-c)(a(2\theta+1) - b(2\theta P_0 + w))\right]\cdot f\left(\overline{\pi}_R\right) - F\left(\overline{\pi}_R\right)\right] \\
&\cdot P_0(a(\theta+1) - b(\theta+1-\gamma)P_0),
\end{aligned}$$

which equates to 0, if and only if,

$$(w - c - \widetilde{\delta}P_0)(a(\theta+1) - b(\theta+1-\widetilde{\gamma})P_0) - \frac{1}{2}(w-c)(a(2\theta+1) - b(2\theta P_0 + w)) = \frac{F\left(\overline{\pi}_R\right)}{f\left(\overline{\pi}_R\right)}.$$

Next, we take the derivative with respect to γ,

$$\frac{\partial \mathbb{E}[\overline{\pi}_S(\delta,\gamma)]}{\partial \gamma} = bP_0(w-c-\delta P_0)\cdot F\Big(\overline{\pi}_R\Big)$$
$$-\Big[(w-c-\delta P_0)(a(\theta+1)-b(\theta+1-\gamma)P_0)-\frac{1}{2}(w-c)(a(2\theta+1)-b(2\theta P_0+w))\Big]$$
$$\cdot P_0(a+bw-bP_0(2(1-\gamma)+\delta))\cdot f\Big(\overline{\pi}_R\Big).$$

Equating to zero and substituting in the expression found above,

$$bP_0(w-c-\widetilde{\delta}P_0)\cdot F\Big(\overline{\pi}_R\Big)-P_0(a+bw-bP_0(2(1-\widetilde{\gamma})+\widetilde{\delta}))\cdot F\Big(\overline{\pi}_R\Big)=0$$
$$\Leftrightarrow \Big[bP_0(w-c-\widetilde{\delta}P_0)-P_0(a+bw-bP_0(2(1-\widetilde{\gamma})+\widetilde{\delta}))\Big]\cdot F\Big(\overline{\pi}_R\Big)=0$$
$$\Leftrightarrow \Big[bP_0(-c)-aP_0+2bP_0^2(1-\widetilde{\gamma})\Big]\cdot F\Big(\overline{\pi}_R\Big)=0$$
$$\Leftrightarrow \Big[2b(1-\widetilde{\gamma})P_0^2-(a+bc)P_0\Big]\cdot F\Big(\overline{\pi}_R\Big)=0 \Leftrightarrow \widetilde{\gamma}=1-\frac{a+bc}{2bP_0}.$$

For the objective $\mathbb{E}[\overline{\pi}_S(\delta,\gamma)]$ to be concave its Hessian has to be negative semi-definite, i.e.,

$$\frac{\partial^2 \mathbb{E}[\overline{\pi}_S(\delta,\gamma)]}{\partial \delta^2}\leq 0 \text{ and } \frac{\partial^2 \mathbb{E}[\overline{\pi}_S(\delta,\gamma)]}{\partial \delta^2}\cdot\frac{\partial^2 \mathbb{E}[\overline{\pi}_S(\delta,\gamma)]}{\partial \gamma^2}-\left(\frac{\partial^2 \mathbb{E}[\overline{\pi}_S(\delta,\gamma)]}{\partial \delta\partial\gamma}\right)^2\geq 0.$$

These second partial derivatives are given by

$$
\begin{aligned}
&\frac{\partial^2 \mathbb{E}[\overline{\pi}_S(\delta,\gamma)]}{\partial \delta^2} = -2P_0^2(a(\theta+1) - b(\theta+1-\gamma)P_0)^2 \cdot f\Big(\overline{\pi}_R\Big) \\
&+ \Big[(w-c-\delta P_0)(a(\theta+1) - b(\theta+1-\gamma)P_0) - \frac{1}{2}(w-c)(a(2\theta+1) - b(2\theta P_0 + c))\Big] \\
&\cdot P_0^2(a(\theta+1) - b(\theta+1-\gamma)P_0)^2 \cdot \frac{\partial f(\pi)}{\partial \pi}\Big|_{\pi=\overline{\pi}_R}, \\
&\frac{\partial^2 \mathbb{E}[\overline{\pi}_S(\delta,\gamma)]}{\partial \delta \partial \gamma} = -bP_0^2 \cdot F\Big(\overline{\pi}_R\Big) \\
&+ P_0^2\Big[(a(\theta+1) - b(\theta+1-\gamma)P_0)(a+3bw-2bc-bP_0(2(1-\gamma)+3\delta)) \\
&- \frac{1}{2}b(w-c)(a(2\theta+1) - b(2\theta P_0 + w))\Big] \cdot f\Big(\overline{\pi}_R\Big) \\
&- \Big[(w-c-\delta P_0)(a(\theta+1) - b(\theta+1-\gamma)P_0) - \frac{1}{2}(w-c)(a(2\theta+1) - b(2\theta P_0 + w))\Big] \\
&\cdot P_0^2(a(\theta+1) - b(\theta+1-\gamma)P_0)(a+bw-bP_0(2(1-\gamma)+\delta)) \cdot \frac{\partial f(\pi)}{\partial \pi}\Big|_{\pi=\overline{\pi}_R}, \\
&\frac{\partial^2 \mathbb{E}[\overline{\pi}_S(\delta,\gamma)]}{\partial \gamma^2} = -2bP_0^2\Big[(w-c-\delta P_0)(a(\theta+2) + bw - bP_0(\theta+3(1-\gamma)+\delta)) \\
&+ \frac{1}{2}(w-c)(a(2\theta+1) - b(2\theta P_0 + w))\Big] \cdot f\Big(\overline{\pi}_R\Big) \\
&+ \Big[(w-c-\delta P_0)(a(\theta+1) - b(\theta+1-\gamma)P_0) - \frac{1}{2}(w-c)(a(2\theta+1) - b(2\theta P_0 + w))\Big] \\
&\cdot P_0^2(a+bc-bP_0(2(1-\gamma)+\delta))^2 \cdot \frac{\partial f(\pi)}{\partial \pi}\Big|_{\pi=\overline{\pi}_R}.
\end{aligned}
$$

For a general distribution $F(\cdot)$, the two conditions for negative semi-definiteness need not be met. For example, $\frac{\partial^2 \mathbb{E}[\overline{\pi}_S(\delta,\gamma)]}{\partial \delta^2} \geq 0$ for any parameter instance and f that satisfy

$$
\frac{1}{2}\Big[(w-c-\delta P_0)(a(\theta+1) - b(\theta+1-\gamma)P_0) - \frac{1}{2}(w-c)(a(2\theta+1) - b(2\theta P_0 + c))\Big] \cdot \frac{\partial f(\pi)}{\partial \pi}\Big|_{\pi=\overline{\pi}_R} \geq f\Big(\overline{\pi}_R\Big).
$$

C.2.7 Proof of Proposition 4.4.7

Using Lemma 4.4.6, the optimal uniform discount solves:

$$
(w-c-\delta^* P_0)(a(\theta+1) - b(\theta+1-\gamma^*)P_0) - \frac{1}{2}(w-c)(a(2\theta+1) - b(2\theta P_0 + w)) = \frac{F\Big(\overline{\pi}_R\Big)}{f\Big(\overline{\pi}_R\Big)}.
$$

Substituting in the optimal pass-through $\gamma^* = 1 - \frac{a+bc}{2bP_0}$ and the uniform distribution yields the following:

$$
\begin{aligned}
&(w - c - \delta^* P_0)\left(a(\theta+1) - b\theta P_0 - \frac{a+bc}{2}\right) - \frac{1}{2}(w-c)(a(2\theta+1) - b(2\theta P_0 + w)) \\
&= \left(\frac{a+bc}{2b} - w + \delta^* P_0\right)\frac{a-bc}{2} + \theta(P_0 - w + \delta^* P_0)(a - bP_0) - \overline{\pi}_L \\
&\Leftrightarrow \frac{1}{2}b(w-c)\left(w+c\right) - \delta^* P_0\left(a(\theta+\frac{1}{2}) - b\theta P_0 - \frac{bc}{2}\right) \\
&= \left(\frac{a+bc}{2b} - w\right)\frac{a-bc}{2} + \theta(P_0 - w)(a - bP_0) + \delta^* P_0\left(a(\theta+\frac{1}{2}) - \theta b P_0 - \frac{bc}{2}\right) - \overline{\pi}_L \\
&\Leftrightarrow \delta^* P_0\left(a(2\theta+1) - 2\theta b P_0 - bc\right) = \frac{1}{2}b(w-c)\left(w+c\right) - \left(\frac{a+bc}{2b} - w\right)\frac{a-bc}{2} - \theta(P_0 - w)(a - bP_0) + \overline{\pi}_L \\
&\Leftrightarrow \delta^* = \frac{\frac{1}{2}b\left(w^2 - c^2\right) - \frac{a^2 - b^2c^2}{4b} + w\frac{a-bc}{2} - \theta P_0(a - bP_0) + \theta w(a - bP_0) + \overline{\pi}_L}{P_0\left(a - bc + 2\theta(a - bP_0)\right)} \\
&\Leftrightarrow \delta^* = \frac{w}{2P_0} + \frac{\overline{\pi}_L + \frac{1}{2}b\left(w^2 - c^2\right) - \frac{a^2 - b^2c^2}{4b} + P_0(a - bc + \theta(a - bP_0))}{P_0\left(a - bc + 2\theta(a - bP_0)\right)} - 1 \\
&\Leftrightarrow \delta^* = \frac{w}{2P_0} + \frac{2\overline{\pi}_L + (2P_0 - \frac{a}{2b} - \frac{c}{2})(a - bc) + b\left(w^2 - c^2\right) + 2\theta P_0(a - bP_0)}{2P_0\left(a - bc + 2\theta(a - bP_0)\right)} - 1.
\end{aligned}
$$

Bibliography

Acimovic, Jason, Stephen C. Graves. 2015. Making better fulfillment decisions on the fly in an online retail environment. *Manufacturing & Service Operations Management* **17**(1) 34–51.

Agrawal, Shipra, Nikhil R Devanur. 2014. Bandits with concave rewards and convex knapsacks. *Proceedings of the Fifteenth ACM Conference on Economics and Computation*. 989–1006.

Ailawadi, Kusun, Paul Farris, Ervin Shames. 1999. Trade promotion: Essential to selling through resellers. *MIT Sloan Management Review* **41**(1) 83–92.

Alimonti, Paola, Viggo Kann. 2000. Some APX-completeness results for cubic graphs. *Theoretical Computer Science* **237**(1-2) 123–134.

Amin, Kareem, Michael Kearns, Peter Key, Anton Schwaighofer. 2012. Budget optimization for sponsored search: Censored learning in mdps. Available at arXiv:1210.4847.

Arkin, Esther M., Refael Hassin. 1998. On local search for weighted k-set packing. *Mathematics of Operations Research* **23**(3) 640–648.

Auer, Peter. 2002. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research* **3** 397–422.

Auer, Peter, Nicolo Cesa-Bianchi, Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* **47**(2-3) 235–256.

Badanidiyuru, Ashwinkumar, Robert Kleinberg, Aleksandrs Slivkins. 2018. Bandits with knapsacks. *Journal of the ACM* **65**(3) 13:1–13:55.

Balseiro, Santiago, Yonatan Gur. 2018. Learning in repeated auctions with budgets: Regret minimization and equilibrium. Available at SSRN 2921446.

Bell, David R., Xavier Drèze. 2002. Changing the channel: A better way to do trade promotions. *MIT Sloan Management Review* **43**(2) 42–49.

Berman, Piotr. 2000. A $d/2$ approximation for maximum weight independent set in d-claw free graphs. *Nordic Journal of Computing* **7**(3) 178–184.

Berman, Piotr, Toshihiro Fujito. 1999. On approximation properties of the independent set problem for low degree graphs. *Theory of Computing Systems* **32**(2) 115–132.

Bernstein, Fernando, Awi Federgruen. 2005. Decentralized supply chains with competing retailers under demand uncertainty. *Management Science* **51**(1) 18–29.

Berstein, Yael, Jon Lee, Hugo Maruri-Aguilar, Shmuel Onn, Eva Riccomagno, Robert Weismantel, Henry Wynn. 2008. Nonlinear matroid optimization and experimental design. *SIAM Journal on Discrete Mathematics* **22**(3) 901–919.

Berstein, Yael, Shmuel Onn. 2008. Nonlinear bipartite matching. *Discrete Optimization* **5**(1) 53–65.

Bertsimas, Dimitris, David B Brown, Constantine Caramanis. 2011. Theory and applications of robust optimization. *SIAM Review* **53**(3) 464–501.

Bertsimas, Dimitris, Dessislava Pachamanova, Melvyn Sim. 2004. Robust linear optimization under general norms. *Operations Research Letters* **32**(6) 510–516.

Bertsimas, Dimitris, Melvyn Sim. 2004. The price of robustness. *Operations Research* **52**(1) 35–53.

Bertsimas, Dimitris, Robert Weismantel. 2005. *Optimization over integers*, vol. 13. Dynamic Ideas Belmont.

Besbes, Omar, Assaf Zeevi. 2015. On the (surprising) sufficiency of linear models for dynamic pricing with demand learning. *Management Science* **61**(4) 723–739.

Blattberg, Robert C., Richard Briesch, Edward J. Fox. 1995. How promotions work. *Marketing Science* **14**(3) 122–132.

Blattberg, Robert C., Scott A. Neslin. 1990. *Sales promotion: Concepts, methods and strategies*. Prentice Hall.

Bomze, Immanuel M., Marco Budinich, Panos M. Pardalos, Marcello Pelillo. 1999. The maximum clique problem. Ding-Zhu Du, Panos M. Pardalos, eds., *Handbook of Combinatorial Optimization*. Springer, 1–74.

Borgs, Christian, Jennifer Chayes, Nicole Immorlica, Kamal Jain, Omid Etesami, Mohammad Mahdian. 2007. Dynamics of bid optimization in online advertisement auctions. *Proceedings of the 16th International Conference on World Wide Web*. ACM, 531–540.

Cachon, Gérard P. 2003. *Handbooks in Operations Research and Management Science: Supply Chain Management*, chap. Supply chain coordination with contracts. Elsevier, Amsterdam, 229–340.

Cachon, Gérard P., Martin A. Lariviere. 2005. Supply chain coordination with revenue-sharing contracts: Strengths and limitations. *Management Science* **51**(1) 30–44.

Calmon, Andre P., Stephen C. Graves. 2017. Inventory management in a consumer electronics closed-loop supply chain. *Manufacturing & Service Operations Management* published online in Articles in Advance 19 Apr 2017.

Campo, Katia, Els Gijsbrechts, Patricia Nisol. 2000. Towards understanding consumer response to stock-outs. *Journal of Retailing* **76**(2) 219–242.

Candogan, Ozan, Saša Pekeč. 2018. Efficient allocation and pricing of multifeatured items. *Management Science* **64**(12) 5521–5543.

Caro, Felipe, Jérémie Gallien. 2010. Inventory management of a fast-fashion retail network. *Operations Research* **58**(2) 257–273.

Caro, Felipe, Jérémie Gallien. 2012. Clearance pricing optimization for a fast-fashion retailer. *Operations Research* **60**(6) 1404–1422.

Chakraborty, Tanmoy, Eyal Even-Dar, Sudipto Guha, Yishay Mansour, S Muthukrishnan. 2010. Selective call out and real time bidding. *International Workshop on Internet and Network Economics*. Springer, 145–157.

Chandra, Barun, Magnus M. Halldorsson. 2001. Greedy local improvement and weighted set packing approximation. *Journal of Algorithms* **39**(2) 223–240.

Chandrasekaran, R., Santosh N. Kabadi, Katta G. Murthy. 1982. Some NP-complete problems in linear programming. *Operations Research Letters* **1**(3) 101–104.

Chen, Xin, David Simchi-Levi. 2004. Coordinating inventory control and pricing strategies with random demand and fixed ordering cost: The infinite horizon case. *Mathematics of Operations Research* **29**(3) 698–723.

Cohen, Maxime C., Swati Gupta, Jeremy Kalas, Georgia Perakis. 2018a. An efficient algorithm for dynamic pricing using a graphical representation. Available at SSRN 2772231.

Cohen, Maxime C., Jeremy Kalas, Georgia Perakis. 2017a. Optimizing promotions for multiple items in supermarkets. Available at SSRN 3061451.

Cohen, Maxime C., Ngai-Hang Z. Leung, Kiran Panchamgam, Georgia Perakis, Anthony Smith. 2017b. The impact of linear optimization on promotion planning. *Operations Research* **65**(2) 446–468.

Cohen, Maxime C, Ilan Lobel, Renato Paes Leme. 2018b. Feature-based dynamic pricing. Available at SSRN 2737045.

Cohen, Maxime C., Georgia Perakis, Robert S. Pindyck. 2018c. Pricing with limited knowledge of demand. Available at SSRN 2673810.

Cooper, Lee G, Penny Baron, Wayne Levy, Michael Swisher, Paris Gogos. 1999. Promocast: A new forecasting method for promotion planning. *Marketing Science* **18**(3) 301–316.

Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. 2009. *Introduction to Algorithms*. 3rd ed. MIT Press.

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

Dani, Varsha, Thomas P. Hayes, Sham M. Kakade. 2008. Stochastic linear optimization under bandit feedback. *Proceedings of the Annual Conference on Learning Theory (COLT)*.

Dhillon, Inderjit, Yuqiang Guan, Brian Kulis. 2005. A unified view of kernel k-means, spectral clustering and graph cuts. Tech. rep., UTCS.

Ding, Wenkui, Tao Qin, Xu-Dong Zhang, Tie-Yan Liu. 2013. Multi-armed bandit with budget constraint and variable costs. *Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Drèze, Xavier, David R. Bell. 2003. Creating win-win trade promotions: Theory and empirical analysis of scan-back trade deals. *Marketing Science* **22**(1) 16–39.

Dunning, Iain, Joey Huchette, Miles Lubin. 2017. JuMP: A modeling language for mathematical optimization. *SIAM Review* **59**(2) 295–320. URL http://arxiv.org/abs/1508.01982.

Elmachtoub, Adam N., Retsef Levi. 2016. Supply chain management with online customer selection. *Operations Research* **64**(2) 458–473.

Everett, Hugh. 1963. Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research* **11**(3) 399–417.

Ferreira, Kris, David Simchi-Levi, He Wang. 2018. Online network revenue management using thompson sampling. *Operations Research* **66**(6) 1586–1602.

Ferreira, Kris Johnson, Bin Hong Alex Lee, David Simchi-Levi. 2016. Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management* **18**(1) 69–88.

Fisher, Marshall L. 1981. The lagrangian relaxation method for solving integer programming problems. *Management Science* **27**(1) 1–18.

Foekens, Eijte W., Peter S.H. Leeflang, Dick R. Wittink. 1998. Varying parameter models to accommodate dynamic promotion effects. *Journal of Econometrics* **89**(1) 249–268.

Gabrel, Virginie, Cécile Murat, Aurélie Thiele. 2014. Recent advances in robust optimization: An overview. *European Journal of Operational Research* **235**(3) 471–483.

Golrezaei, Negin, Max Lin, Vahab Mirrokni, Hamid Nazerzadeh. 2019. Boosted second price auctions: Revenue optimization for heterogeneous bidders. Available at SSRN 3016465.

Google Ads. 2019a. About your data's freshness. https://support.google.com/adwords/answer/2544985?hl=en. [Online; accessed 1-February-2019].

Google Ads. 2019b. Daily budget: definition. https://support.google.com/google-ads/answer/6312?hl=en. [Online; accessed 1-February-2019].

Gopalakrishnan, Sanjith, Daniel Granot, Frieda Granot, Greys Sošić, Hailong Cui. 2016. Allocation of greenhouse gas emissions in supply chains. Working Paper.

Gutin, Gregory. 2013. Independence and cliques. Jonathan L. Gross, Jay Yellen, Ping Zhang, eds., *Handbook of Graph Theory*, 2nd ed. Discrete Mathematics and Its Applications, CRC Press, 475–489.

Halldórsson, Magnús M., Kiyohito Yoshihara. 1995. Greedy approximations of independent sets in low degree graphs. *Proceedings of the 6th International Symposium on Algorithms and Computation*. 152–161.

Harsha, Pavithra, Shivaram Subramanian, Joline Uichanco. 2016. Omni-channel revenue management through integrated pricing and fulfillment planning. Working Paper.

Hassin, Refael, Arie Tamir. 1989. Maximizing classes of two-parameter objectives over matroids. *Mathematics of Operations Research* **14**(2) 362–375.

Håstad, Johan. 1996. Clique is hard to approximate within $n^{1-\epsilon}$. *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*. 627–636.

Hazan, Elad, Shmuel Safra, Oded Schwartz. 2006. On the complexity of approximating k-set packing. *Computational Complexity* **15**(1) 20–39.

Heilman, Carrie M, Kent Nakamoto, Ambar G Rao. 2002. Pleasant surprises: Consumer response to unexpected in-store coupons. *Journal of Marketing Research* **39**(2) 242–252.

Hemmecke, Raymond, Matthias Köppe, Jon Lee, Robert Weismantel. 2010. Nonlinear integer programming. *50 Years of Integer Programming 1958-2008*. Springer, 561–618.

Hochbaum, Dorit S. 2007. Complexity and algorithms for nonlinear optimization problems. *Annals of Operations Research* **153**(1) 257–296.

Khan, Mehmood, Matloub Hussain, Hussein M. Saber. 2016. Information sharing in a sustainable supply chain. *International Journal of Production Economics* **181**(A) 208–214.

Kim, Sang Yong, Richard Staelin. 1999. Manufacturer allowances and retailer pass-through rates in a competitive environment. *Marketing Science* **18**(1) 59–76.

Kleinberg, Jon, Éva Tardos. 2005. *Algorithm Design*. Addison Wesley.

Kök, A Gürhan, Marshall L Fisher, Ramnath Vaidyanathan. 2008. Assortment planning: Review of literature and industry practice. *Retail Supply Chain Management*. Springer, 99–153.

Köppe, Matthias. 2012. On the complexity of nonlinear mixed-integer optimization. *Mixed Integer Nonlinear Programming*. Springer, 533–557.

Lai, T.L., Herbert Robbins. 1985. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics* **6**(1) 4–22.

Lee, Jon, Shmuel Onn, Robert Weismantel. 2009. Approximate nonlinear optimization over weighted independence systems. *SIAM Journal on Discrete Mathematics* **23**(4) 1667–1681.

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

Mazzone & Associates. 2014. Supermarkets and grocery stores revenue in the united states from 2006 to 2020.

Möhring, Rolf H. 2001. *Scheduling under Uncertainty: Bounding the Makespan Distribution*, *in Computational Discrete Mathematics, Lecture Notes in Computer Science*, vol. 2122. Springer, Berlin, Heidelberg, 79–97.

Mulmuley, Ketan, Umesh V. Vazirani, Vijay V. Vazirani. 1987. Matching is as easy as matrix inversion. *Combinatorica* **7**(1) 105–113.

Murry, John P. Jr., Jan B. Heide. 1998. Managing promotion program participation within manufacturer-retailer relationships. *Journal of Marketing* **62**(1) 58–68.

Nagarajan, Mahesh, Steven Shechter. 2013. Prospect theory and the newsvendor problem. *Management Science* **60**(4) 1057–1062.

Nagarajan, Mahesh, Greys Sošić. 2008. Game-theoretic analysis of cooperation among supply chain agents: Review and extensions. *European Journal of Operational Research* **187**(3) 719–745.

Nijs, Vincent, Kanishka Misra, Eric T. Anderson, Karsten Hansen, Lakshman Krishnamurthi. 2010. Channel pass-through of trade promotions. *Marketing Science* **29**(2) 250–267.

Onn, Shmuel. 2003. Convex matroid optimization. *SIAM Journal on Discrete Mathematics* **17**(2) 249–253.

Özer, Özalp., Yanchong Zheng. 2016. Markdown or everyday low price? the role of behavioral motives. *Management Science* **62**(2) 326–346.

Özer, Özalp., Yanchong Zheng, Kay-Yut Chen. 2011. Trust in forecast information sharing. *Management Science* **57**(6) 1111–1137.

Pani, Abhishek, S Raghavan, Mustafa Sahin. 2017. Large-scale advertising portfolio optimization in online marketing. Working Paper.

Papadimitriou, Christos H. 1984. Polytopes and complexity. *Progress in Combinatorial Optimization* 295–305.

Papadimitriou, Christos H., Mihalis Yannakakis. 1982. The complexity of restricted spanning tree problems. *Journal of the ACM* **29**(2) 285–309.

Pardalos, Panos M., Jue Xue. 1994. The maximum clique problem. *Journal of Global Optimization* **4**(3) 301–328.

Pasternack, Barry A. 1985. Optimal pricing and returns policy for perishable commodities. *Marketing Science* **4**(2) 166–176.

Pekgün, Pelin, Ronald P. Menich, Suresh Acharya, Phillip G. Finch, Frederic Deschamps, Kathleen Mallery, Jim Van Sistine, Kyle Christianson, James Fuller. 2013. Carlson rezidor hotel group maximizes revenue through improved demand management and price optimization. *INFORMS Journal on Applied Analytics* **43**(1) 21–36.

Phillips, Robert Lewis. 2005. *Pricing and revenue optimization*. Stanford University Press.

Poddar, Amit, Naveen Donthu. 2013. Improving trade promotions through virtual forward buying. *Journal of Business & International Marketing* **28**(1) 16–28.

Poddar, Amit, Naveen Donthu, Atul Parvatiyar. 2013. Drivers of trade promotion receptiveness: The role of relationship and trade promotion satisfaction. *Journal of Marketing Theory and Practice* **21**(1) 45–56.

PricewaterhouseCoopers. 2017. Iab internet advertising revenue report.

PricewaterhouseCoopers. 2018. Iab internet advertising revenue report.

Robbins, Herbert. 1952. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society* **58**(5) 527–535.

Rusmevichientong, Paat, John N. Tsitsiklis. 2010. Linearly parameterized bandits. *Mathematics of Operations Research* **35**(2) 395–411.

Rusmevichientong, Paat, David P. Williamson. 2006. An adaptive algorithm for selecting profitable keywords for search-based advertising services. *Proceedings of the 7th ACM Conference on Electronic Commerce*. 260–269.

Schrijver, Alexander. 1998. *Theory of Linear and Integer Programming*. Wiley.

Silva-Risso, Jorge M., Randolph E. Bucklin, Donald G. Morrison. 1999. A decision support system for planning manufacturers' sales promotion calendars. *Marketing Science* **18**(3) 274–300.

Talluri, Kalyan T., Garrett J. van Ryzin. 2004. *The theory and practice of revenue management*. Springer-Verlag.

Taylor, Terry A. 2002. Supply chain coordination under channel rebates with sales effort effects. *Management Science* **48**(8) 992–1007.

The Nielsen Company. 2014. Cracking the trade promotion code. URL http://www.nielsen.com/us/en/insights/news/2014/cracking-the-trade-promotion-code.html.

The Reinvestment Fund. 2011. Understanding the grocery industry. URL http://www.cdfifund.gov/what_we_do/resources/Understanding%20Grocery%20Industry_for%20fund_102411.pdf.

Tran-Thanh, Long, Archie Chapman, Jose Enrique Munoz De Cote Flores Luna, Alex Rogers, Nicholas R Jennings. 2010. Epsilon-first policies for budget-limited multi-armed bandits. *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Tran-Thanh, Long, Archie C Chapman, Alex Rogers, Nicholas R Jennings. 2012. Knapsack based optimal policies for budget-limited multi-armed bandits. *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Tran-Thanh, Long, Lampros Stavrogiannis, Victor Naroditskiy, Valentin Robu, Nicholas R Jennings, Peter Key. 2014. Efficient regret bounds for online bid optimisation in budget-limited sponsored search auctions.

Tsay, Andy A. 1999. The quantity flexibility contract and supplier-customer incentives. *Management Science* **45**(10) 1339–1358.

van Donselaar, Karel, Tom van Woensel, Rob A.C.M. Broekmeulen, Jan C. Fransoo. 2006. Inventory control of perishables in supermarkets. *International Journal of Production Economics* **104**(2) 462–472.

Van Heerde, Harald J, Peter SH Leeflang, Dick R Wittink. 2000. The estimation of pre- and post-promotion dips with store-level scanner data. *Journal of Marketing Research* **37** 383–395.

Williamson, David P., David B. Shmoys. 2010. *The Design of Approximation Algorithms*. Cambridge University Press.

Wittink, Dick R., Michael J. Addona, William J. Hawkes, John C. Porter. 1988. Scan*pro: The estimation, validation and use of promotional effects based on scanner data. Internal Paper, Cornell University.

Wolsey, Laurence A., George L. Nemhauser. 1999. *Integer and Combinatorial Optimization*. Wiley.

Xia, Yingce, Wenkui Ding, Xu-Dong Zhang, Nenghai Yu, Tao Qin. 2016. Budgeted bandit problems with continuous random costs. *Asian Conference on Machine Learning*. 317–332.

Yi, Tongnyoul, Katta G. Murty, Cosimo Spera. 2002. Matchings in colored bipartite networks. *Discrete Applied Mathematics* **121**(1-3) 261–277.

Yuan, Hong, Miguel I. Gómez, Vithala R. Rao. 2013. Trade promotion decisions under demand uncertainty: A market experiment approach. *Management Science* **59**(7) 1709–1724.