Revenue management in online markets

Revenue management in online markets: pricing and online advertising

A catalogue record is available from the Eindhoven University of Technology Library.
ISBN: 978-90-386-5147-7

SIKS Dissertation Series No. 2020-32
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Printed by: ProefschriftMaken || www.proefschriftmaken.nl
Cover design by Remco Wetzels.

Revenue management in online markets: pricing and online advertising

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op woensdag 9 december 2020 om 16.00 uur

door

Jason Sunil Rhuggenaath

geboren te Willemstad, Curaçao

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	Prof.dr. I.E.J. Heynderickx
promotor:	Prof.dr.ir. U. Kaymak
copromotor(en):	dr. Y. Zhang
	dr. A.E. Akçay
leden:	Prof.dr.ir. I.J.B.F. Adan
	Prof.dr. A. Nowé (Vrije Universiteit Brussel)
	Prof.dr.ir. H.W.G.M. van Heck (Erasmus Universiteit Rotterdam)
	Prof.dr.ir. J.A. La Poutré (Technische Universiteit Delft and CWI)

Het onderzoek dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Acknowledgements

The completion of this dissertation would not have been possible without the help of others. I would like to take this opportunity to express my gratitude towards the many people who have supported and encouraged me during my PhD.

First of all, I would like to thank Saskia Krijger and dr. Adriana Gabor for "introducing me to research" as part of the ESE Research Traineeship Programme at the Erasmus University Rotterdam. A special thanks goes out to Adriana for bringing the vacancy that ultimately resulted in me being hired for a PhD position to my attention

The work presented in this dissertation would not have been possible without the help of my supervision team. I would like to thank my supervisors, dr. Yingqian Zhang, dr. Alp Akçay and prof.dr.ir. Uzay Kaymak for the opportunity they gave me to pursue a PhD, and for guiding and assisting me throughout the PhD trajectory. During the past fours years, I have worked most closely with my copromotors and daily supervisors, Yingqian and Alp. Yingqian and Alp, thank you for all your help over the years. You were always available to listen to my ideas, to provide constructive feedback, and to help me improve my writing. Thank you for replying to all of my emails and for reading the many drafts. Uzay, thank you for your guidance and support. Your supervision style gave me a lot of freedom to explore my own ideas and I really appreciated this.

I would like to thank the members of my doctoral committee, prof.dr.ir. Ivo Adan, prof.dr. Ann Nowé, prof.dr.ir. Eric van Heck, prof.dr.ir. Han La Poutré; it is truly an honor to have such accomplished scholars evaluating my work. I highly appreciate their constructive comments and helpful suggestions.

Part of the work in this thesis was made in collaboration with several people outside of the university. I would like to thank Gönenç Tarakcıoğlu, Fatih Çolak, Muratcan Tanyerli, Wojtek Przedzimirski, and the rest of the team at Triodor, Azerion and OrangeGames for sharing their knowledge and for their assistance with technical issues. Furthermore, I thank Sicco Verwer, Michiel Hilgeman and Charlie Ye for the discussions and fruitful cooperation on other projects, which are not included in this dissertation. Although these projects are not included, the benefits of these collaborations are nonetheless reflected in this dissertation.

Various organizations have played important roles in making this PhD trajectory possible. I would like to thank the Graduate Program Industrial Engineering (GP-IE) for their guidance throughout my PhD trajectory. Additionally, I thank the Dutch Research School for Information and Knowledge Systems (SIKS), the Graduate Program Operations Management & Logistics (GP-OML) and the Dutch Network on the Mathematics of Operations Research (LNMB) for their support and amazing lectures. I would like to thank Jan Brinkhuis, Arnoud den Boer, Bert Zwart, Sem Borst, Dick den Hertog, Frank Thuijsman, Dion Gijswijt, Etienne de Klerk, Jesper Nederlof, Monique Laurent, Lisa Maillart and Marco Slikker for their interesting lectures, and for taking the time to explain complex concepts and ideas, which has helped me during my research. I also thank Collin Drent, Melvin Drent, Youri Raaijmakers, Ernst Roos and Riley Badenbroek for the interesting discussions and for their help with LNMB courses.

I would like to thank my colleagues at Information Systems (IS) group of the School of Industrial Engineering for providing a good academic and social environment. I am grateful for the daily help and support of the IS group secretarial staff, without whom many things would not have run smoothly. Annemarie van der Aa and Emmy Bos, thank you for helping me with all of the administrative tasks, such as arranging travel permits, and for organizing various social activities on behalf of the group.

The past few years would not have been as enjoyable if it were not for my fellow colleagues. In particular, I would like to thank the following staff members and PhD students (listed in pseudo-random order): Jonnro Erasmus, Shaya Pourmirza, Rodrigo Gonçalves, Konstantinos Traganos, Bambang Suratno, Bilge Çelik Aydin, Peipei Chen, Ege Adalı, Rick Gilsing, Paulo Roberto de Oliveira da Costa, Caro Fuchs, Sicui Zhang, Raoul Nuijten, Reza Refaei Afshar, Jian Chen, Juntao Gao, Emil Rijcken, Sudhanshu Chouhan, Amirreza Farahani, Zeynep Ozturk Yurt, Frank Berkers, Jos Trienekens, Rob Kusters, and Sander Peters. I am very grateful for your help and I really enjoyed the interesting discussions, the breaks from work, and the social activities that we did together. Jonnro, Rick and Ege, thanks for making my time in Eindhoven enjoyable, especially during our evening endeavours. Jonnro and Rick, I really enjoyed our squash sessions even though I never actually won a match. Caro, many thanks for lending me your bike: I *really* hope it is still where I left it! Paulo, thank you for introducing me to the many wonderful aspects of the Brazilian cuisine and culture; in order to limit the printing costs of this thesis I just mention *coxinha*, *novinha* and *açaí*. I would also like to thank Afonso Henrique Sampaio Oliveira, Vincent Karels, Joost de Kruijff, Erwin van Wingerden, Taher Ahmadi, and the rest of the PhD students at the OPAC group for including me in various social activities.

Next, I would like to thank my friends and family. I am grateful to my friends for their friendship and for all the fun times we had together over the years. For most of us it has been about 10 years since we left Curaçao and it has been truly amazing to see the kind of people that we have become. I would also like to thank my family –

in particular the many cousins – in The Netherlands for all the good times we had during the last years. I especially enjoyed the Christmas dinners and the impromptu "chillings" that we had.

Last but not least, I would like to thank my parents and my brother. Mom and dad, none of this would have possible without your love, support, encouragement, and all of the sacrifices that you have made for me over the years. Thank you for all your hard work and everything that you have taught me. As for my brother, you have been there from the start. Thanks for looking out for me and for all of the fun we had during my time on Curaçao. I am very proud of you, and I am grateful that you are there to look after mom and dad after I left for The Netherlands. Mom and dad, although you are far away, you are always in my thoughts; this work is dedicated to you.

Jason Rhuggenaath
Rotterdam, October 2020

Table of contents

Chapter 1

Introduction

1.1 Modern online markets

In the current economy goods and services are increasingly sold via the internet. Online retailers, such as Amazon, offer millions of products for sale including books, mobile phones, computers, clothes and various other electronics. Some companies (e.g. Nike and even some supermarkets) have both physical stores and an online channel in order to sell their products. There are also companies, for example companies like eBay, that operate entirely via the internet.

There are multiple ways to sell goods and services via the internet. In some cases, potential consumers can browse through the various pages on the website of the seller, where different products are listed together with their prices. If the consumer finds the product that he is looking for and is willing the pay the displayed price, he can proceed to purchase the item by making a payment via the internet. Another popular way to purchase goods on the internet is via an online auction. An auction is a way of selling items, which can be goods or services, that are put up for bid by an auctioneer. In an auction the potential buyers (or bidders) compete with each other by placing bids for an item. The value of the bid indicates the price they are willing to pay for an item. The higher the bid, the better the chance that a bidder will win. For example, the company eBay offers owners of products the opportunity sell their products to potential buyers via an auction and eBay acts as a market where demand meets supply.

With the rise of the internet, auctions have become increasingly important in the domain of online advertising. In the online display advertising market, publishers (owners of websites) sell page views or *impressions* to interested advertisers. This market has grown rapidly over the last ten years. This growth is associated with a new online selling channel through which publishers can sell impressions to advertisers called the Real-Time Bidding (RTB) market. In the RTB market publishers auction off impressions in real-time as users visit their websites.

The design of the online marketplaces and technological advances have a number of implications for companies that operate on or sell products on online markets. First, it has become possible to store large amounts of information related to various business operations. Second, a lot of transactions in online markets are high volume transactions with a repeated nature. Third, it has become easier and less costly for companies to change key parameters (e.g. prices, website design etc.) that effect sales and revenues.

These developments present both opportunities and challenges for the practice of revenue management. The fact that outcomes at various parameter settings can be logged and stored results in a lot of information. This information provides companies with opportunities in the sense that this information can be leveraged in order to design models and algorithms that can be used for improved decision making in various revenue management problems. However, leveraging this information can also be a challenging task. Most decisions in revenue management problems are made in uncertain environments and often only partial feedback of these decisions is received. As a consequence, designing models and algorithms that leverage available information is not a straightforward task.

The topic of this thesis is revenue management in online markets. We study a number of revenue management problems that arise in online markets such as online advertisement markets and online retail markets. The revenue management problems that are considered in this thesis can be broadly organized in three categories: allocation decisions, pricing decisions, and buying decisions. In the context of online advertising, these decisions are often related to the decisions of buyers and sellers that participate in online auctions for advertisements and to the interaction of various selling mechanisms that are used in order to sell online advertisements. Furthermore, some pricing decisions are also studied in the context of online retail markets. Most of the revenue management problems that are studied involve decision making under some form of uncertainty and where the decisions affect the (partial) feedback that is obtained from the environment.

The rest of this Chapter is organized as follows. In Section 1.2 and Section 1.3 some background information is provided related to the main topics and concepts that are discussed in this thesis. Section 1.2 provides background information on online advertising markets and Section 1.3 provides background information on auctions and dynamic pricing problems. In Section 1.4 the main contributions of this thesis are discussed. Section 1.5 provides an outline of the rest of this thesis.

1.2 Online advertising markets

The two main types of online advertising are display advertising and sponsored search advertising [57, 161]. Display advertising refers to advertisements that are displayed when users visit websites in browsers and are typically displayed in banners located in

specific ad slots on a website. Sponsored search advertising refers to advertisements that are displayed when users search for keywords on search engines. In this thesis we focus on online display advertising.

In online display advertising there are buyers and sellers. The sellers are called publishers and they can be thought of as owners of websites. Publishers sell impressions to potential buyers: when a user visits a webpage with an advertisement slot, an impression is generated and this means that there is an opportunity to display an advertisement to this user in this advertisement slot. The buyers correspond to advertisers that are interested in the procurement of impressions on the websites of publishers in order to reach an intended audience (e.g. potential consumers).

There are a number of channels through which publishers can sell their inventory of impressions. The two main categories are the guaranteed selling channel and the non-guaranteed selling channel. In the guaranteed selling channel a number of impressions are sold in advance at a fixed price that is negotiated up front. In the non-guaranteed selling channel (also called the Real-Time Bidding or Real-Time Buying market) impressions are sold in real-time via auctions on ad exchanges.

In the rest of this section we provide some background information about the various selling channels and discuss the main properties that are relevant for this thesis.

1.2.1 Guaranteed selling channel

In the guaranteed selling channel impressions are bought and sold via guaranteed contracts. Guaranteed contracts are agreed upon ahead of time: the arrangements are made before the users visit the websites of the publisher. A guaranteed contract specifies the number of impressions that will be sold, when these impressions will be sold, at what price they will be sold, and the advertisement that will be displayed.

Guaranteed contracts are often useful when advertisers have a long-standing relationship with the publisher that facilitates the customization of ad formats and price negotiations (e.g., quantity discounts, bundling). If advertisers are risk averse, paying a premium to buy a guaranteed inventory in advance helps mitigate uncertainty in either the auctions' outcomes or the amount of impressions that will be available in RTB on specific dates. Finally, advertisers highly concerned with ensuring brand safety will choose guaranteed contracts so their ads appear on high-quality, reputable websites [57].

1.2.2 Real-Time Bidding market

In the Real-Time Bidding (RTB) market impressions are sold in real-time as users arrive on the websites of the publisher. The RTB market is a non-guaranteed selling channel because the buyer (the advertiser) is not guaranteed to win a fixed number impressions. More specifically, for each impression, the buyer bids against other

buyers in an online auction and he only wins the impression if he wins the auction.

Initially, the RTB market was considered a useful alternative selling channel in addition to the guaranteed selling channel. In the early days, it was mostly used to sell the remnant inventory of impressions that could not be sold via the guaranteed selling channel. However, due to technological advances and the possibility of advanced targeting of users, the RTB market has become an important selling channel in its own right. According to a report by the Interactive Advertising Bureau[1], digital revenues for the full year 2018 surpassed 100 billion US Dollars for the first time. Internet advertising revenues in the United States totaled 107.5 billion US Dollars for the full year of 2018, with Quarter 4 of 2018 accounting for approximately 31.4 billion and Quarter 3 of 2018 accounting for approximately 26.6 billion. Furthermore, revenues for the full year 2018 increased 21.8% over the full year 2017.

The main players in the RTB market are the advertisers, the publishers and the various intermediaries (see Figure 1.1 for a schematic overview). These intermediaries provide the infrastructure, tools and algorithms that needed in order to buy, sell and serve ads. In RTB there are three main intermediaries: Supply Side Platforms (SSPs), Demand Side Platforms (DSPs) and an Ad Exchange (ADX) which connects SSPs and DSPs. Publishers operate on the RTB market via SSPs. The SSPs are intermediaries that provide publishers with the infrastructure and tools to manage their inventory of impressions and to sell their impressions on Ad Exchanges (ADX). Advertisers which are interested in displaying advertisements are connected to DSPs. The DSPs are intermediaries that facilitate the advertisers in the ad buying process. Advertisers can provide DSPs with extra information related to their advertising campaigns. For example, advertisers can specify how much budget they have available for bidding on impressions in specific periods, targets for the amount of impressions that they want to win, and various targeting criteria related to their intended audience (i.e., the users that visit websites of publishers). The DSPs subsequently take this information into account when bidding on behalf of the advertisers on the Ad Exchange.

When a user visits a webpage with an advertisement (ad) slot an impression is generated and the publisher sends a request to the ADX (via an SSP) indicating that an ad can potentially be displayed in this particular ad slot. At the same time, advertisers that are connected to DSPs send bid requests to the ADX indicating that they are willing to bid for this impression. A real-time auction then decides which advertiser is allowed to display its ad and the amount that the advertiser needs to pay.

There are several ways for publishers to sell their impressions via the RTB market. Two of the most common approaches are by using the *waterfall mechanism* and by using *Header Bidding*. These approaches differ with respect to how the publisher interacts with the different selling channels and intermediaries. These approaches are discussed in more detail below.

[1] See `https://www.iab.com/insights/2018-full-year-iab-internet-ad-revenue-report/`

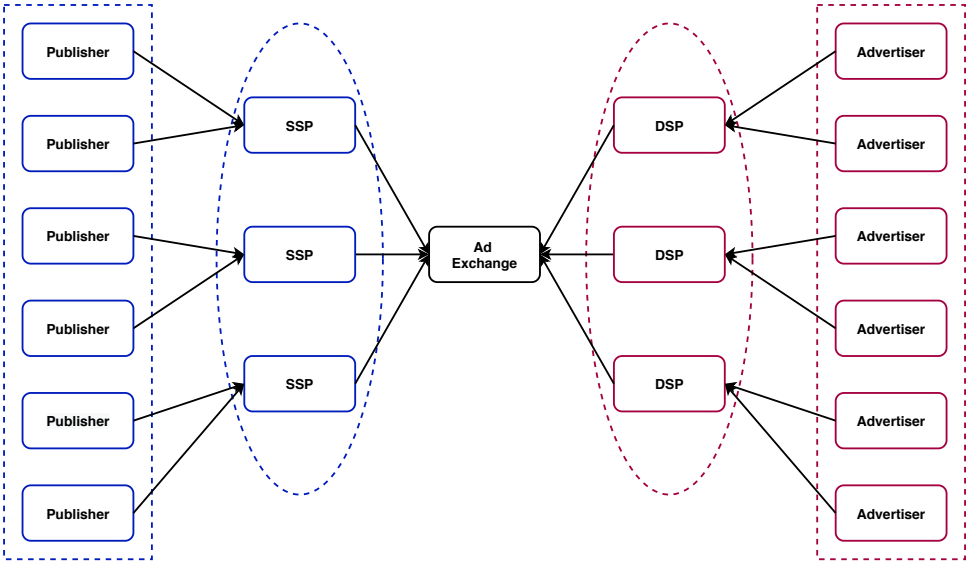


Figure 1.1: Overview of RTB market.

1.2.3 Waterfall mechanism

Publishers are typically connected to multiple SSPs and can choose any one of these SSPs in order sell an impression on the RTB market. The waterfall mechanism is a way to make this decision in a systematic way.

In Figure 1.2 a schematic overview of the waterfall mechanism is presented. In the waterfall mechanism the SSPs are ordered or prioritized in hierarchical levels [12, 97, 129]. When an impression becomes available, the publisher first contacts SSP number 1. If the impression is not sold, then SSP number 2 is contacted. If the impression is not sold, then SSP number 3 is contacted. This process continues until either the impression is sold or when the publisher exhausts the list of available SSPs. If at the end of the process the impression is not sold, the impression can be offered for sale on a sales channel specific for remnant inventory (this is called the *backfill* option) or by placing an in-house ad.

The revenue from selling an impression can vary due to differences in SSPs. Some SSPs are connected to more advertisers and may be able to obtain higher revenues from these advertisers. Typically, the list of SSPs in the waterfall mechanism is ordered based on the historical performance of SSPs (e.g. average revenue obtained from impressions sold in the past).

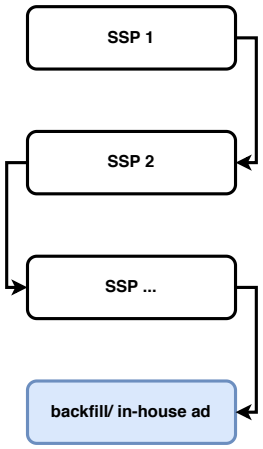


Figure 1.2: Overview of waterfall mechanism.

1.2.4 Header Bidding

Header Bidding (HB) is an alternative approach to sell impressions on the RTB market that has been recently proposed [12, 57, 129]. The main purpose of header bidding was to address some of the drawbacks associated with the waterfall mechanism [12, 57, 129]. One drawback of the waterfall mechanism is related to the publisher side of the market: SSPs are contacted one-by-one and in some cases some SSPs are not contacted at all, and this could lead to missed revenue for the publisher since the publisher does not know the revenue associated with each SSP. In other words, the publisher may potentially not get the best deal for each impression. Another drawback is related to the advertiser side of the market: some advertisers are not connected to all SSPs, and as a consequence, the advertisers that are connected to SSPs that have a high priority in the waterfall have an advantage because they have the opportunity to bid on more impressions.

In Figure 1.3 a schematic overview of the header bidding process is presented. In header bidding, the publisher connects to multiple SSPs simultaneously in order to sell an impression. Each SSP connects to an ADX and is involved in a separate auction (we refer to this as the *internal auction* of the SSP) and reports a value (a bid) back to the publisher indicating the revenue for the publisher. The publisher observes the reported bids from each SSP and determines a winner among the SSPs (the header bidding auction). In practice, the SSP that returns the highest bid is the winner. The winning SSP will notify the winning advertiser and the winning advertiser will then be able to show its advertisement in the ad slot.

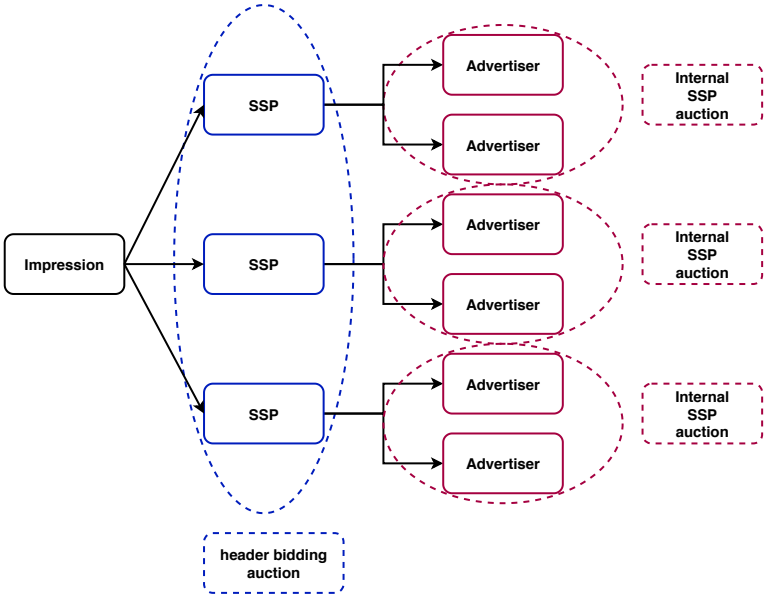


Figure 1.3: Overview of header bidding process.

1.3 Auctions and dynamic pricing

Auctions have been a central tool in selling goods and services across history. Nowadays, auctions are used in many of industries. Governments for instance use auction to sell public goods such as spectrum licenses. Auctions are also used to sell unique and historical art pieces. Given the rise of technology and new industries, auctions have gained even more prevalence. In online advertisement, there are millions of auctions taking place daily.

There are three main players in an auction: the seller, the auctioneer (auction organizer), and a set of potential buyers. In most cases the seller and the auctioneer are the same entity but this need not be the case.[2] The seller owns the item and wants to sell it to a potential buyer. The auctioneer organizes (or conducts) the auction on behalf of the seller by using a specific *auction format*. There are many auction formats: two common formats are the *English* auction and the sealed-bid *first-price* auction.

In one variant of the English auction [108], the sale is conducted by an auctioneer who begins by calling out a low price and raises it, typically in small increments, as long as there are at least two interested bidders. The auction stops when there is only one interested bidder. Each bidder indicates an interest in purchasing at the current

[2]In online advertising markets the publisher is the seller, but the auction is organized by (and takes place on) the Ad Exchange. Furthermore, the auction format is also decided by the Ad Exchange and not the seller.

price in a manner apparent to all by, say, raising a hand. Once a bidder finds the price to be too high, he signals that he is no longer interested by lowering his hand. The auction ends when only a single bidder is still interested. This bidder wins the object and pays the auctioneer an amount equal to the price at which the second-last bidder dropped out.

In the sealed-bid first-price auction bidders submit bids in sealed envelopes and the bidder submitting the highest bid wins the object and pays what he bid [108].

While there are many auction formats, all auctions tend to have some common properties. A common property of auctions is that they elicit information (typically in the form of bids)[3], from potential buyers regarding their willingness to pay, and the outcome—that is, who wins what and pays how much—is determined solely on the basis of the received information and the rules of the auction [108].

As was mentioned before, auctions arise in many settings and there are a number of different auction formats. The auction formats that are relevant for this thesis are discussed in more detail in the next sections.

1.3.1 Second-price auctions

With the rise of the internet, auctions have become increasingly important in the domain of online advertising. Most of the inventory of impressions in display advertising is sold via second-price auctions. Consider the following setting. There is a seller that wants to sell an item and there are a number of buyers (or bidders) that are interested in the item. In a single item sealed-bid second-price auction, the protocol is as follows: (i) each buyer submits a bid for the item; (ii) the buyer with the highest bid will receive the item; (iii) the buyer that receives the item pays an amount equal to the second highest bid; (iv) the revenue for the seller equals the second highest bid.

There is another variant of the second-price auction that is often used in online advertisement markets and this variant is called the second-price auction with a *reserve price*. In a second-price auction with a reserve price[4], the seller specifies a value (called the reserve price) which represents the minimum price that he wants for the item. The revenue for the seller at a particular reserve price depends on bids placed in the auction and the value of the reserve price. The revenue of the seller is determined according to the following rules: (i) if the highest bid in the auction is at least as large as the reserve price, then the revenue equals the maximum of the reserve price and the second highest bid; (ii) if the highest bid in the auction is smaller than the reserve price, then the revenue equals zero (and the item is not sold).

In Figure 1.4 and Figure 1.5 numerical examples are given of second-price auctions in the context of online advertising on the RTB market. In Figure 1.4 an example

[3] An example where information is not elicited using bids, is the English auction. In the example of the English auction, potential buyers provided the required information by raising and lowering their hands.

[4] In online advertising applications, the reserve price may or may not be disclosed to the bidders (see e.g., [169]).

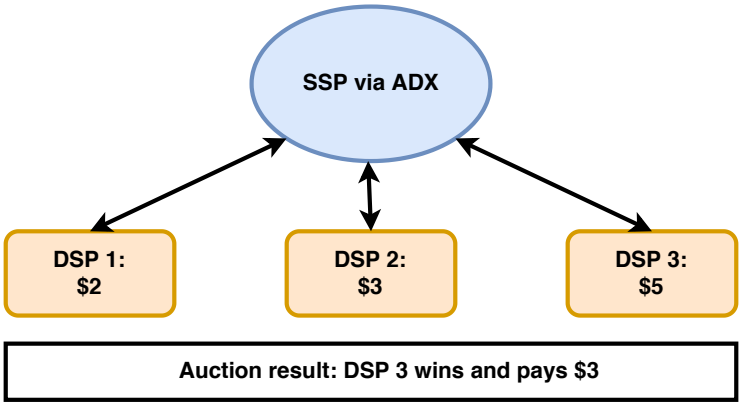


Figure 1.4: Example of second-price auction.

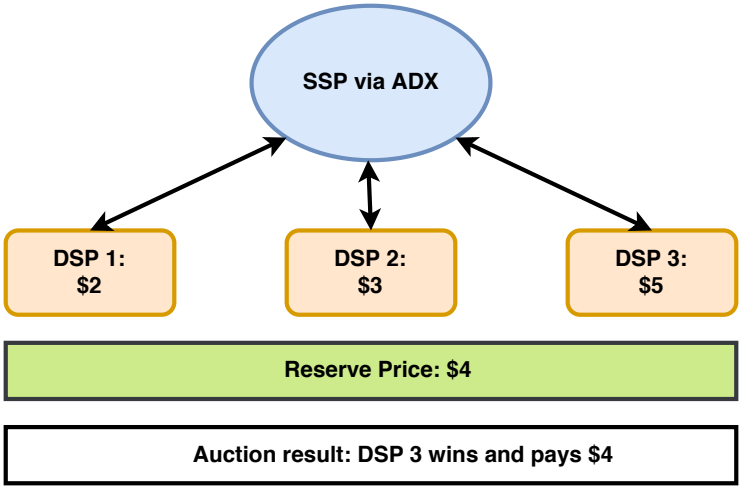


Figure 1.5: Example of second-price auction with a reserve price.

of a second-price auction is presented and in Figure 1.5 an example of a second-price auction with a reserve price is presented. In these examples, the DSPs submit bids to an SSP (via an ADX) on behalf of the connected advertisers (the buyers). In the examples, DSP 3 is the winner of the auction. When there is no reserve price, the revenue for the seller (the publisher) equals the second highest bid. In the auction with a reserve price, the revenue equals the reserve price.

The numerical examples show the impact that a reserve price can have on the revenue of the seller. The reserve price provides the seller with a parameter that he can adjust in order to influence the revenue from the auction. By properly adjusting the reserve price the seller can increase his revenue and this is especially beneficial in situations where there is a large gap between the second highest bid and the highest

bid [108].

1.3.2 Posted-price auctions

In a posted-price auction [20, 106] the protocol is much simpler compared to second-price auctions. In a posted-price auction, the protocol is as follows: (i) the seller announces (or posts) a price; (ii) the buyer decides to buy or decides not to buy; (iii) if the buyer decides to buy, the revenue for the seller equals the announced price; (iv) if the buyer decides not to buy, the revenue for the seller equals zero.

Posted-price auctions can be seen as a special case of a second-price auction with a reserve price when the reserve price is announced to the bidders. Consider a second-price auction with a reserve price where there is a single buyer. If the buyer places a bid that is at least as high as the reserve price, then the item will be sold and the revenue for the seller equals the reserve price. If the buyer places a bid below the reserve price, then the item is not sold and the revenue for the seller equals zero. The link with posted-price auctions is made by interpreting the reserve price as the posted price in a posted-price auction. Thus, placing a bid that is at least as high as the reserve price is equivalent to accepting the posted price. Similarly, placing a bid below the reserve price is equivalent to not accepting the posted price.

1.3.3 Floor prices and auction formats in online advertising

Publishers typically specify a minimum price – called the *floor price* – for their impressions when selling on the RTB market (via an ad exchange or header bidding). For example, if the publisher thinks that the final result (i.e., the offered revenue) of the header bidding auction is too low, the publisher can use a floor price to indicate that he is not willing to sell the impression for that revenue. Floor prices allow publisher some control on the revenues they receive: if a publisher thinks its inventory is undervalued (and it does not want to sell) it can enforce this via the floor price.

Most of the inventory of impressions in display advertising is sold via second-price auctions with a reserve price [123, 161]. In these auctions, the "reserve price" is actually the floor price: (i) the publisher selects a value for the floor price and this value is passed along as information to the SSP; (ii) the value of the reserve price that is used in the auction is set equal to the value of the floor price. The problem of determining the optimal floor price is then referred to as the reserve price optimization problem. This process is illustrated in Figure 1.6. In Figure 1.6 the value of the floor price equals \$4 and this value is passed to the SSP. The reserve price is set equal to the value of the floor price and the result of the auction and the revenue for the publisher are exactly the same as in Figure 1.5.

In this thesis, when the publisher is using a floor price and the auction format is the second-price auction, it is understood that the auction format is the second-price auction with a reserve price in which the value of the reserve price equals the value

of the floor price.

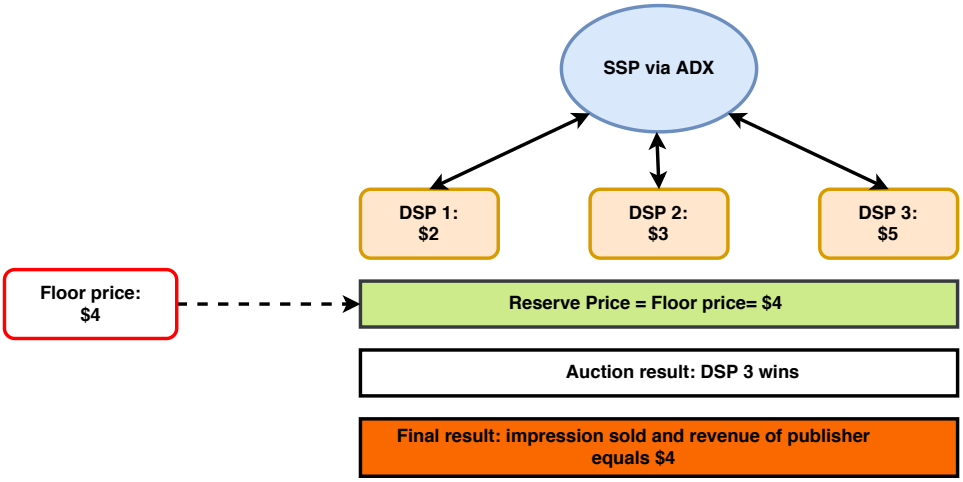


Figure 1.6: Example of second-price auction with a floor price.

Note that the concept of a floor price is separate from the specific auction format that is used: floor prices can also be used in combination with auction formats besides the second-price auction with a reserve price. Figure 1.7 gives an example of a floor price with a first-price auction. In Figure 1.7, the publisher selects a floor price of \$6 and this value is passed to the SSP. The highest bid in the auction is \$5. Since the value of the highest bid (\$5) is lower than the floor price of \$6, the impression is not sold and the revenue for the publisher is zero. Note that in this example, since the auction format is the first-price auction, the floor price does not influence amount that any bidder shall pay (should they win the impression). The floor price is merely "applied" to the outcome of the auction and is used to decide if the payment that results from the bid of the winning bidder is acceptable to the publisher (i.e., whether it is high enough). In the example in Figure 1.7, the bid of the winning bidder was not high enough.

Whenever floor prices are involved in this thesis, we will always specify: (i) what the auction format is; (ii) whether the floor price is passed along to the auction organizer or not; (iii) if the floor price is visible to the bidders or not; (iv) how the final revenue of the auction is determined.

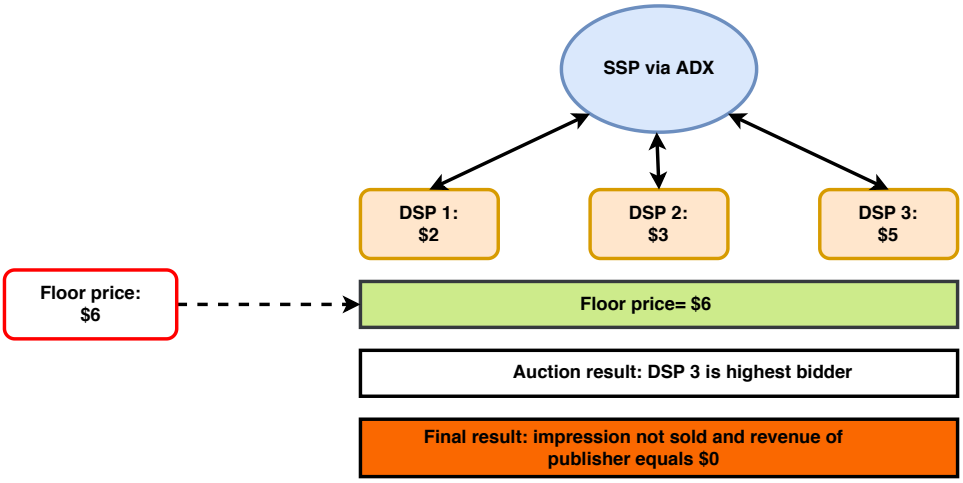


Figure 1.7: Example of first-price auction with a floor price.

1.3.4 Dynamic pricing

In many practical situations the seller of an item does not directly interact with the buyer in order to sell the item, but instead merely offers the buyer the *opportunity* to buy the item. Loosely speaking, in a dynamic pricing problem, the seller sets a price for the item and afterwards the seller observes a *market response*. This market response can have different interpretations and the precise interpretation depends on the context and the modeling assumptions. For example, the market response can represent the demand, sales, revenues or profits. One of the main questions from the seller perspective is how to determine the best price.

Note that in many auction settings there can also be a pricing component. For example, in second-price auctions with a reserve price, the seller is interested in determining the best or optimal reserve price. Similarly, in posted-price auctions, the seller wants to figure out what the optimal posted-price is. The problem of finding the best price (i.e., reserve price or posted-price) can therefore be interpreted as a dynamic pricing problem. Note however, that dynamic pricing problems also arise in other settings besides auctions.

In order to illustrate the differences between a posted-price auctions and dynamic pricing problems, consider the following example. Consider a small webshop that only sells two items, item A and item B, and where item A has price p_A and item B has price p_B. Suppose that we are interested in tracking the sales of item A. In practice, when buyers arrive at the webshop, the seller does not ask them "Are you willing to buy item A at a price of p_A?" and "Are you willing to buy item B at a price of p_B?". Instead, buyers browse in the webshop and the seller observes a market response afterwards: the seller might observe that item A was purchased 10 times

today. In this example, the fact that item A was purchased 10 times, implies that 10 buyers accepted the take-it-or-leave-it price p_A. However, there is no formal "auction protocol" that is used in the transactions.

Auctions are common and useful in the domain of online advertising, since one item (e.g. an advertisement slot) is sold at a time, and buyers and sellers directly interact in order to buy/sell a specific item. Dynamic pricing problems on the other hand, are more general and can model scenarios that arise in different e-commerce settings, such as online retail markets for clothes, electronics. For example, on the website of the sports brand Nike, there are hundreds of products (items) for sale and there are millions of potential buyers. Asking a potential buyer whether he wants to purchase an item at a specific price is not feasible or practical, since there are too many items and buyers would not enjoy the shopping experience on the website. Instead, online retailers observe a market response: for every item, they observe the number of times that it is sold and at what particular price during, say, an hour or a day.

1.4 Contributions of thesis

In this thesis the focus is on revenue management problems in online markets. Section 1.2 and Section 1.3 provided background information on some common online market places and the processes that are involved when selling products on these market places. A common feature of the processes that were discussed, was the fact that there are often key parameters that the decision makers have at their disposal in order to influence certain outcomes (e.g., the revenue from sales). The ability to adjust these key parameters and observe the outcomes of these adjustments gives rise to a number of revenue management problems. The central question in these revenue management problems is: what is the best way to adjust these parameters over time and how do these adjustments depend on the available information? The fact that outcomes at various parameter settings can be logged and stored results in a lot of information that can be useful for the design of algorithms that can be used in various revenue management problems. This thesis provides models and algorithms that aim to leverage this information in order to improve decision making in various revenue management problems.

In this thesis the focus is mostly on revenue management problems where decisions need to be made under some form of uncertainty. The revenue management problems that are considered in this thesis can be broadly organized in three categories: allocation decisions, pricing decisions, and buying decisions. These three categories are discussed in more detail below.

1.4.1 Allocation decisions

Two popular selling channels that publishers can use to sell their impressions are the guaranteed selling channel (by using guaranteed contracts) and the RTB market. In Chapter 2 of this thesis, we consider a publisher that uses both guaranteed contracts and a waterfall mechanism in order to sell impressions. These two channels for selling impressions leads to an allocation decision: (i) what fraction of its inventory should be sold on the RTB market and (ii) which SSPs should the publisher select in order to sell its inventory of impressions? This problem setting has a number interesting features. The first feature is related to the trade-off associated with the allocation of an impression: a specific impression that is allocated to a guaranteed contract cannot be sold via an SSP in the waterfall mechanism (and vice versa). The second feature is related to the uncertainty associated with the ability of an SSP to sell a particular impression and the risk of ending up with unsold inventory: ideally, in a feasible allocation all of the impressions should be sold, but due to the uncertainty of sales on the RTB market there is a risk that an impression is not sold and remains unused. Furthermore, the publisher has to take into account a constraint that ensures that enough impressions are allocated to meet the requirements of the guaranteed contracts.

Previous works do not specify how publishers should make decisions when they have access to both the RTB market (using a waterfall) and use guaranteed contracts. Furthermore, the uncertainty of sales on the RTB market is not taken account. Therefore, in Chapter 2, we address this gap in the literature. In Chapter 2, we propose a model that takes into account both the guaranteed selling channel and the waterfall mechanism, and that also takes the uncertainty of sales on the RTB market into account.

1.4.2 Pricing decisions

Reserve prices in second-price auctions

The first pricing decision that is considered in this thesis is related to the reserve price optimization problem of publishers in online advertising markets. Most of the inventory of impressions in display advertising is sold via second-price auctions with a reserve price [123, 161]. The revenue for the publisher at a particular reserve price depends on bids placed in the auction and the value of the reserve price. Notice that the reserve price is a parameter that the publisher can control and by controlling the reserve price, the publisher can influence the revenue from the auction. Thus, the publisher needs to make a pricing decision: he needs to decide which reserve price is the best one to use. The reserve price optimization problem refers to the problem of how to select the reserve price in order to maximize expected revenue for the publisher.

There is a lot of existing research on second-price auctions with reserve prices and

there are several well-known results [108, 120, 126, 143]. For example, [126] shows how the optimal reserve price can be determined assuming that the distribution of the bids is known. However, this assumption is not realistic because in practice the distribution of the bids is unknown. Also, most of the previous literature studies the reserve price optimization problem from the perspective of the auction designer (or the auction organizer) that has access to the bids that are placed in the auction. This may be a reasonable assumption to make for big companies such as Facebook and Google that own ad exchanges, but it is not a reasonable assumption for most publishers.

Chapter 3 of this thesis takes the perspective of publishers that are small and medium sized enterprises (SMEs), and studies the reserve price optimization problem where the distribution of the bids is unknown and where the bids are not revealed to the publisher. The only feedback that the publisher receives is whether the impression was sold or not and the associated revenue. This setting matches the situation of SME publishers that do not see the bids that are placed on their inventory of impressions (as they are not the auction organizer). In Chapter 3 we propose a method that incorporates knowledge about the rules of second-price auctions into a multi-armed bandit framework for optimizing reserve prices in our limited information setting.

Reserve prices and header bidding

The second pricing decision that is considered in this thesis is related to revenue management in the setting of header bidding. In particular, we consider a situation where the publisher uses header bidding in order to connect to multiple SSPs in order to sell his impressions and where each SSP runs a second-price auction. After the SSPs run their auctions, they return a value back indicating the revenue for the publisher if he sells the impression on that particular SSP. The revenue of the publisher equals the maximum of the values returned back by the SSPs.

Chapter 4 of this thesis studies the following reserve price optimization problem in the context of header bidding: how should the publisher choose a vector of reserve prices (one for each SSP) in order to maximize his expected revenue?

This problem has two challenging aspects. First, the set of possible choices is large and trying each alternative separately would only be useful if the sales horizon (the number of impressions that will be sold) is very large. Second, the optimal vector of reserve prices can in general not be found by finding the best reserve price at the (individual) SSP level and then combining these reserve prices. In Chapter 4 we take both of these aspects into account and we model the reserve price optimization problem on a header bidding platform as slate bandit problem with a non-separable reward function. In a slate bandit problem, a slate consists of a number of slots and each slot has a number of base actions. At the slot level, every base action leads to a reward. The slate-level reward is a function (a combination) of the rewards at the slot level. In the context of header bidding, the slots are the SSPs, the base

actions are the reserve prices for each SSP, and the slate-level reward is obtained by taking the maximum of the rewards at the slot level. The first challenging aspect mentioned above means that the number of slates is large relative to the sales horizon. The second challenging aspect means that reward function is such that the optimal slate-level action cannot be determined by learning the optimal slot-level for each slot individually. We refer to such a reward function as *non-separable*.

In Chapter 4 we propose algorithms for the slate bandit problem with a non-separable reward function. To the best of our knowledge, existing algorithms for slate bandits (see e.g. [70, 99, 132]) do not have performance guarantees when the slate-level reward function is non-separable, and so in Chapter 4, we address this gap in the literature. We are mainly interested in cases where the number of slates is large relative to the time horizon, so that trying each slate as a separate arm in a traditional multi-armed bandit, would not be useful. In Chapter 4 we show that the regret – the performance loss compared to the optimal algorithm – of our proposed algorithms is sub-linear with respect to the time horizon, despite the large number of slates.

Revenue management with header bidding and ad exchanges

The third pricing decision that we study in this thesis is motivated by the fact that publishers typically have access to multiple selling mechanisms. More specifically, some publishers use both header bidding and an ad exchange in order to sell impressions [73, 100, 133].

In Chapter 5, we study how publishers should make decisions in order to maximize revenues when they have access to both header bidding and an ad exchange in order to sell impressions. Previous works typically focus on a single selling mechanism (ad exchanges that use second-price auctions [18, 41, 124, 146, 171]), or focus on revenue maximization from the perspective of the ad exchange or header bidding partner [97, 133]. These papers, however, do not specify how publishers should make decisions when they have access to both an ad exchange and header bidding. Therefore, in Chapter 5, we address this gap in the literature. We consider a publisher that first observes an offer from header bidding and can accept or reject this offer. If the offer is rejected, the publisher can try to sell the impression on an ad exchange. In this problem, the publisher needs to make two decisions: (i) when to accept the offer from header bidding and when to use the ad exchange; and (ii) if the publisher uses the ad exchange, which floor price it should use on the ad exchange. In Chapter 5, we study how publishers should set their floor prices in order to maximize expected revenues when they have access to both selling mechanisms. We propose two algorithms for this problem based on techniques from the multi-armed bandit literature, and show that their regret – the performance loss compared to the optimal algorithm – is sub-linear in the time horizon.

Dynamic pricing in e-commerce

The fourth pricing decision that we study in this thesis is related to dynamic pricing problems in e-commerce settings such as online markets for fashion and electronics. Many e-commerce companies have the ability to change prices at little costs and this price experimentation is useful in order to learn the best price. However, frequent price changes are not always desirable, because it may confuse customers and lead to negative customer feedback [55]. In practice, companies also have limited inventories on hand during a selling period. Due to limited inventories, the observed sales are not equal to demand anymore. As a consequence the true underlying demand is only partially observed, that is, demand is censored.

Motivated by these observations, Chapter 7 of this thesis studies a dynamic pricing problem with demand censoring and limited price changes. More specifically, we consider a seller that faces demand uncertainty and has to adjust his selling price over the selling horizon in order to learn the optimal price and maximize his cumulative revenue over the selling horizon. The seller faces a business constraint on the number of price changes allowed during the selling horizon and the seller only has a limited (finite) amount of inventories on hand in each selling period. The seller can only observe the sales (minimum between realized demand and available inventory) and thus demand is censored. In each period the seller can replenish his inventory to a particular level. The objective of the seller is to set the best price and inventory level in each period of the sales horizon in order to maximize his profit. The profit is determined by the revenue of the sales minus holding costs and costs for lost sales (unsatisfied demand). In Chapter 7 of this thesis, we propose a policy that adjust prices and inventory levels for this problem.

1.4.3 Buying decisions

The revenue management problems discussed above take the perspective of the seller. In this thesis we also study decisions that buyers need to make in online markets. More specifically, we consider the buying decision of advertisers in online advertising markets.

Most impressions in online advertising are sold via second-price auctions, however as indicated by e.g. [10, 11, 122], a non-trivial fraction of auctions only involve a single bidder and this reduces to a posted-price auction [106] when reserve prices are known: the seller sets a reserve price and the buyer decides whether to accept or reject it. A single publisher can track a large number of visitors with similar properties over time and sell the impressions that are generated by these visitors to buyers. Buyers are typically involved in a large number of auctions, and if they repeatedly interact with the same seller, there is an incentive for them to act strategically [10, 11, 78, 122]. These observations have led to the study of repeated posted-price auctions between a single seller and single (strategic) buyer.

Previous work (e.g., [10, 11, 75, 76, 122, 159]) has only studied repeated posted-price auctions from the perspective of the seller that aims to maximize his revenue and not from the buyer side. Furthermore, previous works assume that the buyer knows his own valuation with certainty in each round. However, in many practical situations, the buyer may have a stochastic valuation which is only revealed after he buys the item. For example, in online advertising applications the buyer (advertiser) generally does not know the exact value of showing ads to a set of users: some users may click on the ad and in some cases the ad may lead to a sale, but the buyer only observes a response after he displays the advertisement to the user.

In Chapter 6 of this thesis, we consider a repeated posted-price auction between a single seller and a single utility maximizing buyer. In every round, the seller posts a price and the buyer decides to buy or not at that price. The buyer does not know the probability distribution of his valuation and only observes a sample from the valuation distribution after he purchases the item. Moreover, the buyer does not know the seller's pricing algorithm or the seller's price set. Furthermore, the seller does not know the valuation distribution and needs to learn how to set the price over time. If the buyer purchases the item, he derives utility from item which is defined as the difference between his valuation and the price paid. If the buyer does not purchase the item, the utility is zero. The goal of the buyer is to make buying decisions in order to maximize his expected utility.

We provide algorithms for buying decisions in posted-price auctions and we study two types of buyers: strategic buyers and non-strategic buyers. Non-strategic buyers are only interested maximizing expected utility and given the prices that are observed and do not attempt to manipulate or influence the observed prices. Strategic buyers are also interested in maximizing expected utility given the observed prices, but they also actively attempt to influence future prices that will be offered. We first consider non-strategic buyers and derive algorithms with performance guarantees that hold irrespective of the observed prices offered by the seller. These algorithms are then adapted into algorithms with similar guarantees for strategic buyers.

1.5 Thesis outline and summary

In this dissertation we use a combination of techniques from operations research and computer science to tackle different revenue management problems that arise in online markets. The common theme across the Chapters in this thesis is that the decisions in the revenue management problems are made under uncertainty. The revenue management problems that are considered in this thesis can be broadly organized in three categories: allocation decisions, pricing decisions, and buying decisions. Chapters 2, 3, 4, 5, 7 focus on the perspective of a seller and tackles allocation decisions and pricing decisions. In Chapter 6, the emphasis is on the buyer perspective and studies buying decisions.

The chapters of this dissertation are based on papers that have been published in or have been submitted to peer-reviewed journals and conferences. As a result, all chapters can be read independently from each other. Below we list the topic of each Chapter of this thesis and give a brief summary of the content of each Chapter.

In Chapter 2 we consider a display-ad allocation problem where an online publisher needs to decide which subset of impressions for advertisement slots should be used in order to fulfill guaranteed contracts and which subset should be sold on supply side platforms (SSPs) in order to maximize the expected revenue. Our modeling approach also takes the uncertainty associated with the sale of an impression by an SSP into account. The way that information is revealed over time allows us to model the display-ad allocation problem as a two-stage stochastic program. Numerical experiments are used to assess the performance of our model and to compare it with a heuristic allocation policy that is used in practice.

In Chapter 3 we consider an online publisher that sells advertisement space via second-price auctions with a reserve price. We study a limited information setting where the probability distribution of the bids from advertisers is unknown and the values of the bids are not revealed to the publisher. Furthermore, we do not assume that the publisher has access to a historical data set with bids. In Chapter 3 we propose a method that incorporates knowledge about the rules of second-price auctions into a multi-armed bandit framework for optimizing reserve prices in our limited information setting. The proposed method can be applied in both stationary and non-stationary environments. We conduct extensive experiments in order to compare our approach with state-of-the-art bandit algorithms. The experiments show that the proposed method outperforms state-of-the-art bandit algorithms in both stationary and non-stationary environments.

In Chapter 4 we study a slate bandit problem with a non-separable reward function and is motivated by the reserve price optimization problem on a header bidding platform. We are mainly concerned with cases where the number of slates is large relative to the time horizon, so that trying each slate as a separate arm in a traditional multi-armed bandit, would not be feasible. Our main contribution is the design of algorithms that still have sub-linear regret with respect to the time horizon, despite the large number of slates. Experimental results on simulated data and real-world data show that our proposed method outperforms popular benchmark bandit algorithms.

In Chapter 5 we study how publishers should set their floor prices in order to maximize expected revenues when they have access to two selling mechanisms, namely an ad exchange and header bidding, in order to sell impressions on the real-time bidding market. We propose two algorithms for this problem based on techniques from the multi-armed bandit literature, and show that their regret – the performance loss compared to the optimal algorithm – is sub-linear in the time horizon. Experiments using simulated data and real-world data illustrate the effectiveness of our algorithms.

In Chapter 6 we study buying decisions in repeated posted-price auction where a seller repeatedly interacts with a single (strategic) utility maximizing buyer for a

number of rounds. We first consider non-strategic buyers and derive algorithms with sub-linear regret bounds that hold irrespective of the observed prices offered by the seller. These algorithms are then adapted into algorithms with similar guarantees for strategic buyers. We provide a theoretical analysis of our proposed algorithms and support our findings with numerical experiments. Our experiments show that, if the seller uses a low-regret algorithm for selecting the price, then strategic buyers can obtain much higher utilities compared to non-strategic buyers. Only when the prices of the seller are not related to the choices of the buyer, it is not beneficial to be strategic, but strategic buyers can still attain utilities of about 75% of the utility of non-strategic buyers.

In Chapter 7 we study a dynamic pricing problem with demand censoring and limited price changes. We consider a seller that faces demand uncertainty and has to adjust his selling price over the selling horizon in order to learn the optimal price and maximize his cumulative revenue over the selling horizon. The seller faces a business constraint on the number of price changes allowed during the selling horizon and the seller only has a limited amount of inventories on hand in each selling period. We propose a heuristic policy for this problem and study its performance using numerical experiments. The results are promising and indicate that the regret of the policy is sub-linear with respect to the sales horizon.

In Chapter 8 we conclude with the main findings of this dissertation. First, we provide an overview of the main contributions of this thesis. Next, we discuss some limitations of the approaches used and we indicate some directions for future research. Finally, we put the contributions of this thesis in a broader context and relate them to other topics and research areas that relate to revenue management in online markets.

Chapter 2

Display-ad allocation with guaranteed contracts and supply side platforms*

In this Chapter we study an allocation problem – the display-ad allocation problem – that publishers face in online advertising. We consider a publisher that sells impressions via guaranteed contracts and via the RTB market by using a waterfall mechanism. In the display-ad allocation problem, the publisher needs to take the uncertainty of the RTB market into account and decide which impressions to allocate to guaranteed contracts and which impressions to allocate to the RTB market.

---

*This chapter is based on Rhuggenaath et al. [141].

2.1 Introduction

In this Chapter, we study the display-ad allocation problem faced by online web publishers. Many online publishers such as news sites have multiple pages (homepage, sports, financial, etc.) where they display different advertisements (ads). Ads can be displayed in different ways such as images, video or text ads and these ads are displayed on specific ad-slots that are on specific webpages. Each time a user visits a webpage that contains ad-slots a number of *impressions* are generated. Two options that these small online publishers usually have to allocate impressions are to (i) enter in guaranteed contracts with advertisers, or to (ii) sell an impression on the online ad auction market via supply side platforms. In this Chapter we focus on the practical problem faced by our industry partner and other online publishers that are small and medium size enterprises (SMEs): determining the optimal allocation of impressions over guaranteed contracts and supply side platforms that maximizes expected revenue.

A guaranteed contract requires a minimum number of impressions to be allocated to an advertiser (the entity that delivers the content of the ad). The online publisher usually enters in guaranteed contract with multiple advertisers and these contracts are specified ahead of time. For example, advertisers can already secure contracts in January for the upcoming holiday season in December. In addition to allocating impressions to meet the requirements of guaranteed contracts, online publishers also have the option to sell the impressions on the online ad auction market. Online publishers at the SME level typically do not enter directly into this market but operate via supply side platforms (SSPs). The online publisher offers the impression to the SSP and the SSP will try to sell it in the online ad market via an auction. Revenue from such a sale to an SSP varies due to the differences in SSPs such as the connected advertisers and the organization of the auctions. In addition, if a particular SSP has been chosen and the auction does not yield a winning advertiser, the publisher may face a risk of not selling the impression. Many possible factors could determine the success of an auction such as the relevant advertisers being out of budget or a particular impression being uninteresting for the connected advertisers. There are several popular SSPs in the market such as Google, LiveRail, OpenX, Appnexus, Smaato, and Rubicon, with possibly different pricing models and with connections to heterogeneous groups of advertisers. Since the SSPs differ with respect to their connections to advertisers, some SSPs are 'safer' than others in the sense that they are almost always able to sell an impression but at a lower price.

A common heuristic (we refer to this as the Priority Assignment (PA) heuristic) that SME publishers use is to first meet all the requirements of the guaranteed contracts, and then sell the remaining inventory on the auction market using a *waterfall* approach. In this approach the publisher has a list with a ranking of different SSPs and the publisher tries to sell the impression to the SSP that is as high as possible on the list. If it is not possible to sell an impression on a particular SSP, then the publisher tries to sell the impression on the next SSP on the list. Usually, as we get

lower on the list the probability of successfully selling the impression increases but at lower prices. At the very bottom the last SSP is usually able to sell almost all the impressions allocated to it, but at substantially lower prices. We refer to such an SSP as a 'safe' SSP. In this Chapter we present a modeling framework that is suitable for publishers, who use a waterfall approach combined with guaranteed contracts, to improve their allocation decisions of impressions.

There are a number constraints that a feasible allocation of impressions needs to satisfy. The first type of constraints stem from the fundamental trade-off associated with the allocation of an impression: a specific impression that is allocated to a guaranteed contract cannot be sold to an SSP (and vice versa). The uncertainty associated with the ability of an SSP to sell a particular impression and the risk of ending up with unsold inventory lies at the heart of the second type of constraints. Ideally, in a feasible allocation all of the impressions should be sold, but due to this uncertainty there is a risk that an impression is not sold and remains unused. This uncertainty is another interesting property of the display-ad allocation problem that is addressed in this work.

Contributions and organization The allocation problem of the online publisher consists of finding a feasible allocation of impressions, between guaranteed contracts and SSPs, that maximizes the expected revenue. There are a limited number of studies on the allocation of impressions to guaranteed contracts and SSPs, but these studies make restrictive assumptions on the available information that the publisher has (for example, that the publisher has information about the bids placed in the online auction). The work in this Chapter therefore aims to fill a gap in the current literature on internet advertisement allocation. We list our contributions as follows.

- We study the display-ad allocation problem by taking both the trade-off between the allocation to guaranteed contracts and allocation to multiple SSPs into account. Our modeling approach also takes the uncertainty associated with the sale of an impression by an SSP into account. The way that information is revealed over time allows us to model the display-ad allocation problem as a two-stage stochastic program. We refer to our model as the Stochastic Programming (SP) model.

- We investigate the quality of the solutions obtained by the SP model. We find that one of the key parameters of the model is the fraction of total impressions that need to be allocated to the guaranteed contracts. The gap between the upperbound and lowerbound of the objective value of the SP model varies considerably with this parameter and is large enough so that it is worthwhile to solve the SP model. We find that the SP model performs very well with a performance gap relative to an upperbound of about 2-3 % in most cases.

- We show the SP model outperforms the allocation policy that is used in practice by our industry partner. Furthermore, we find that the performance gap

between the PA heuristic and SP model also shows a clear relationship with fraction of impressions that are allocated to the guaranteed contracts.

- The results suggest that the benefit of using the SP model is highest in periods where the website traffic is relatively high compared with the targets for the guaranteed contracts. An example of such a situation is when there are seasonal patterns in the demand of advertisers. Another example is when the website traffic varies depending on the time of the day.

The rest of this Chapter is organized as follows. In Section 2.2 we provide a literature review. In Section 2.3 we formally state the advertisement allocation problem. In Section 2.4 we present a stochastic programming formulation and our proposed SP model for the problem. Section 2.4 also discusses bounds for the SP model. Section 2.5 discusses a practical allocation policy that is used in practice. Section 2.6 investigates the properties of the solutions returned by running numerical experiments. Finally, Section 2.7 provides some concluding remarks and directions for future research.

2.2 Related literature

The literature has studied various aspects of internet advertisement allocation. While our problem has some features in common with previously studied problems, the developed methods in the literature cannot be applied to our problem.

Online algorithms for matching between advertisers and ad-slots

Part of the literature approaches the advertisement allocation problem as an online bipartite matching problem where the nodes can be partitioned in two disjoint sets that correspond with the advertisers and the available ad-slots. The ad-slots arrive online and need to be assigned to an advertiser (subject to capacity restrictions) and the objective is to maximize the total number of assigned ad-slots. There are different variations of this problem, see also [117] for an overview. [102] provide a randomized online algorithm that has an approximation ratio of $1 - \frac{1}{e}$, where $e \approx 2.71828$. They further show that the result is tight: in the adversarial model no algorithm can achieve a better ratio. In the random order stochastic model where arriving nodes are drawn repeatedly from a known distribution [80] show that it is possible to achieve an approximation ratio that beats $1-\frac{1}{e}$. Subsequent papers generalize various aspects of this online bipartite matching problem. For example, [118] generalize the results of [80] by considering the case where a match can succeed with a certain probability that can be different for each advertiser.

There are also papers ([50, 79, 160]) that design two-phase online algorithms that have access to a sample of the arriving nodes in a first stage where an offline problem is solved. In the second phase (the online phase) the algorithm decides which ad to serve based on some information obtained from the offline phase. [79] design an

online algorithm that achieves a $1-\epsilon$ approximation ratio where ϵ is a function of the parameters of the optimization model that measures how large a fraction of any resource can be demanded by a single agent, or how much a single agent's value contributes to the total objective. Their algorithm observes the first ϵ fraction of the input and then solves an offline problem on this instance. In the second phase the algorithm uses the dual solution of the first phase problem to determine which ads to serve. They prove that if the demand of each advertiser is large and the contribution of any one impression does not have a big impact on total value of the objective function, then their algorithm provides a $1-\epsilon$ approximation which is nearly optimal. [160] focus on a slightly more general problem called the online assignment problem and their two-phase online algorithm also assumes that a forecast of the underlying bipartite graph is available. In the offline phase the forecast graph is used as input and produces an allocation plan. They prove that if the (possibly non-linear) objective function satisfies some conditions, then there is a concise representation of the solution of the first phase which they refer to as a *compact allocation plan* that is just a few numbers per contract, independent of the number of impressions. The online phase repeatedly takes as input a user visit and decides which ad to serve based on the compact allocation plan. They provide conditions such that the serving decisions made using this allocation plan are nearly optimal, even when the allocation plan was computed on a sampled graph with imperfect forecasts. In [50] a heuristic is developed that is based on the ideas in [160] but is more robust to incorrect forecasts. The work done by [29] further improve upon the methods in [50, 160].

This part of the literature has not (yet) considered the trade-off between allocation to SSPs and guaranteed contracts, therefore these results cannot be applied directly to the problem considered in this Chapter.

Offline algorithms for guaranteed contracts

Another part of the literature uses an offline optimization approach and models the allocation problem as a one-period planning problem. These papers focus on the optimal allocation of ad-slots to guaranteed contracts in order to meet certain targeting and quality constraints. [90] focus on designing predetermined fixed length streams of ads (which they call patterns) that satisfy a number of conditions. They define a pattern as a finite permutation of ads from different advertisers and classify each visitor of a webpage into different types. The key issue is to decide which pattern is to be shown to each type of visitor. In particular they focus on the optimal length of the pattern, the minimum/maximum number of times that a specific ad should appear in a pattern, and the spacing of ads over time. In [158] the allocation problem is modeled as a transportation problem and the key problem is to determine which fraction of visitors of a specific type should be allocated to which guaranteed contract. The author uses a quadratic objective function and derives sufficient conditions that state when this is a good surrogate for several ad delivery performance metrics. In

[69] a chance constrained optimization model for the fulfillment of targeted guaranteed display ad is developed that takes the uncertainty of the supply of viewers into account. These papers focus on the fulfillment of the guaranteed contracts, while explicitly taking the characteristics of the ads and the types of visitors into account. However the trade-off between allocation of ad-slots to SSPs and guaranteed contracts is not addressed.

Trade-off between SSPs and guaranteed contracts

There are very few papers that jointly consider the problem of allocating impressions to guaranteed contracts and selling them in online auctions. In [167] a multi-objective optimization approach is used to determine the fraction of impressions allocated to guaranteed contracts and the fraction to be sold in an online auction. However that approach is not quite suitable for the setting of a small online publisher that we consider in this Chapter since they do not consider the possibility of allocating impressions to different/ multiple SSPs and there is no uncertainty with respect to the ability of an SSP to sell an impression. In [24] a model for yield maximization is developed that takes both guaranteed contracts and SSP's into account. The model determines which impressions should be allocated to guaranteed contracts and which reserve price should be used when selling on the SSP. The objective is to jointly maximize revenue from the AD exchange and some measure for the ad placement quality for impressions allocated to the guaranteed contracts. However their model is not suitable for the problem we consider here since it makes some restrictive assumptions on the information that the publisher has available. The authors assume that the publisher can observe bids on their inventory of impressions and the proposed method exploits this information in order to set optimal reserve prices. In [96] and [47] both guaranteed contracts and SSP's are taken into account and take a more data-driven approach, but they make similar assumptions as in [24]. SME publishers in general do not have access to the information regarding bids and it may be too costly for them to acquire this information, and therefore these methods cannot be applied in our setting. Recent work by [145] proposes a framework that combines guaranteed delivery and pay-per-click auctions in sponsored search. While [145] does combine online auctions and guaranteed contracts, the setting of sponsored search advertising in search engines is quite different from the display ad problem that we address in this Chapter.

Other related problems

Other studies focus on the optimal allocation of advertisements on websites [2, 33, 60, 82, 109, 119]. The focus is on time scheduling (or space sharing) of advertisements on a banner. The banner has multiple time slots through which it is cycled over time, and every slot has a different allocation pattern of advertisements. In some studies, the authors are concerned only with the placement of one advertisement on

a banner or some advertisements side-by-side, with the height of the advertisements equal to the height of the banner. In other studies advertisements are placed in a two-dimensional way. In order to solve these complex optimization problems various approaches have been used such as approximation algorithms ([2, 82]), heuristics ([33, 60, 109]), Lagrangean decomposition and column generation [119]. Some studies also study the effectiveness of certain design choices of advertisements on websites (see e.g. [113]).

We also mention that there is literature that focusses on optimal design of auctions ([71, 86, 101]), the relation between ad auction design and strategies for publishers and advertisers ([23, 54]), the relation between reserve prices and revenues from ad auctions ([3, 4, 42, 44, 123, 127, 137]), and revenue management and optimization in sponsored search advertising ([19, 85, 103, 172]).

2.3 Problem description

The online publisher owns a set N of unique webpages on a domain. Each webpage n has a set M_n of ad-slots, and $|M_n|$ impressions are generated each time a user visits webpage n of the publisher, that is, each ad-slot generates one impression. We assume that the online publisher knows how many users will visit the webpages during the planning period, where the planning period could be for example one hour. For notational convenience, without loss of generality, we consider the case where there is one visitor on each webpage. Note that in general there can be multiple visitors on a particular webpage. However, this can be reduced to the case where there is only one visitor on each webpage by adjusting the total number of webpages, i.e, if webpage n has V_n visitors the adjusted total number of webpages will be $\sum_{v \in N} V_n$.

The online publisher has in total a set C of guaranteed contracts, and each guaranteed contract $c \in C$ is associated with a target level of $\alpha_c \in \mathbb{Z}_+$ impressions which need to be allocated to it during the planning period. The online publisher can choose from a set S of SSPs to sell an impression.

Since the planning period is typically shorter than the period for which the guaranteed contract has been negotiated (the contract period), the target level of each contract is determined upfront as a common practice. For example, assume that an advertiser enters into a guaranteed contract with a publisher for 500.000 impressions during a contract period of 3 months and assume a planning period of 12 hours (allocation decisions for impressions are made twice a day). Then the publisher can decide on the amount of impressions to be allocated during the first day of this contract period and thus can decide on appropriate (and possibly distinct) values of α_c for each of the 2 planning periods on the first day of the contract period. Typically, the advertisers want the amount of impressions to be evenly divided during the contract period, that is, roughly the same amount of impressions should be allocated each day of the contract period. In order to achieve this balanced allocation during the

contract period, the publisher decides (based on patterns of user visits, experience and targeting criteria of the guaranteed contracts) on the appropriate values of α_c for each planning period of 12 hours.

An important characteristic of the display-ad allocation problem is the uncertainty on the outcome of selling an impression on an SSP, i.e. the SSP can be unsuccesful in filling the order. We let Ω denote the finite set of possible scenarios to represent this uncertainty. We define $H^{\omega}_{nms} \in \{0,1\}\ \forall n \in N, m \in M_n, s \in S, \omega \in \Omega$, and let $H^{\omega}_{nms} = 1$ indicate that the request to sell impression m on webpage n on SSP s, was successful in scenario ω. For the SSP $s \in S$ we define $s \in S^{\text{Risky}}$ if and only if there exists an $\omega \in \Omega$ such that $H^{\omega}_{nms} = 0$. S^{Safe} is defined as $S^{\text{Safe}} = S \setminus S^{\text{Risky}}$. The online publisher can estimate the scenarios and the probability of each scenario using historical data. It is known that some SSPs are 'safer' than others in the sense that they are always able to sell an impression. The SSPs in S^{Risky} have the property that they are not always able to sell an impression, whereas the SSPs in S^{Safe} are always able to sell an impression (the Google SSP is an example). The revenue from selling impression m on webpage n to SSP s is R_{nms}. We assume that the revenue parameters R_{nms} are known because, revenue typically does not change in the planning horizon and they can be easily estimated from historical data. The publisher receives a revenue of λ_c per impression allocated to guaranteed contract c.

Definition 2.1 (The display-ad allocation problem)**.** *Refers to the publisher's question of finding a feasible allocation of the impressions such that its expected revenue over all the scenarios is maximized. Furthermore a feasible allocation must satisfy the following conditions: (i) each impression needs to be sold in each scenario; (ii) impressions must be allocated to precisely one SSP or to precisely one guaranteed contract; and (iii) the requirements of each guaranteed contract must be fulfilled.*

We illustrate the display-ad allocation problem with an example.

Example 2.1. *We consider the situation where there is 1 webpage with 3 ad slots, 2 guaranteed contracts, 1 "risky" SSP, 1 "safe" SSP and 2 scenarios, i.e,* $|N| = 1, |M_n| = 3\ \forall n \in N, |S| = 2, |S^{Safe}| = 1, |C| = 2,\ |\Omega| = 2$ *and* $\alpha_c = 1\ \forall c \in C$. *Furthermore, for each guaranteed contract the publisher needs to allocate 1 impression. The revenue for each ad-slot is given in the Table 2.1. If the impression associated with ad-slot 1 is sold on SSP 2, the publisher will receive a revenue of 3 units. Table 2.2 describes the two possible scenarios. Note that there exists a scenario* $\omega \in \Omega$ *such that SSP 1 is not able to sell an impression (this is indicated with a zero in the table). In this example, SSP 1 is therefore the "risky SSP" and SSP 2 is the "safe SSP". Note furthermore that in each scenario any impression can be used in order to fulfill the requirements of the guaranteed contract. That explains why there is always a "1" for each combination of ad-slot and guaranteed contract.*
The task here is to find a feasible allocation of the impressions such that the expected revenue is maximized while satisfying the conditions described above. □

Table 2.1: Revenue for impression associated with each ad-slot.

	slot 1	slot 2	slot 3
SSP 1	6	7	10
SSP 2	3	4	5
Contract 1	5	5	5
Contract 2	2	2	2

Table 2.2: Scenarios and realizations.

	Scenario 1: probability 0.7			Scenario 2: probability 0.3		
	slot 1	slot 2	slot 3	slot 1	slot 2	slot 3
SSP 1	1	1	0	0	0	1
SSP 2	1	1	1	1	1	1
Contract 1	1	1	1	1	1	1
Contract 2	1	1	1	1	1	1

Notes: Scenario 1 corresponds to the left half of the table and Scenario 2 corresponds to the right half of the table. A "1" indicates that a sale is successful and a "0" indicates that a sale is unsuccessful.

2.4 Proposed Stochastic Programming (SP) model

In this section we formulate the display-ad allocation problem of the online publisher as a two-stage stochastic programming model. We will refer to this model as the Stochastic Programming (SP) model. We also discuss upper- and lowerbounds for the objective value of the SP model that can be used to assess the solution quality of the SP model.

2.4.1 Modeling approach

The objective of the online publisher is to find a feasible allocation of impressions between guaranteed contracts and SSPs that maximizes the expected revenue where the expectation is taken over the scenarios. In order to ensure that all impressions are sold (that is, no ad-slots remain unused) we need a solution that balances the impact of the various scenarios on both the revenue and the feasibility of the allocation. Because information is revealed sequentially over time, we adopt a stochastic programming approach in order to model and solve the display-ad allocation problem. We distinguish two stages where the online publisher needs to make decisions, as illustrated in Figure 2.1.

First stage In the first stage, the publisher makes an initial decision about which impressions to sell on which SSP and which impressions to allocate to which guar-

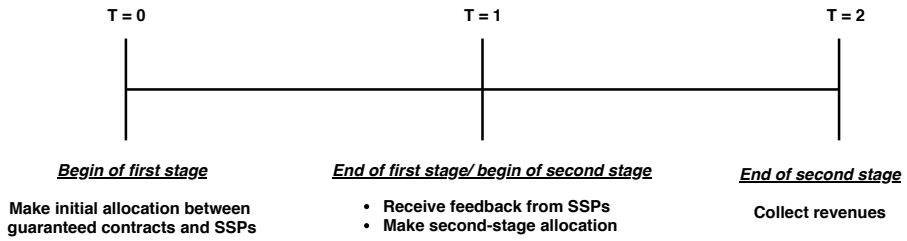


Figure 2.1: Timeline of events in the display-ad allocation problem.

anteed contracts. Impressions allocated to guaranteed contracts cannot be allocated to SSP's later (that is, in the second stage). After the first-stage decisions have been made, there are two possibilities (and one of these must occur) for each impression: (i) the impression has either been allocated to a guaranteed contract; or (ii) the impression has been allocated to an SSP.

Second stage The second stage begins when the publisher receives feedback from the SSP regarding the sale of the impressions. More specifically, the publisher observes whether the SSP was able to sell the impression (that is, the sale was successful) or not (that is, the sale was *rejected*). In terms of the terminology used in stochastic programming, it means that the publisher observes the realization of a specific scenario. After observing the realization of the scenario, the publisher can make a "corrective" second-stage decision. In the second stage the publisher has to make decisions regarding the *rejected* impressions. There are two options for each rejected impression: (i) allocate the impression to *another* SSP; or (ii) allocate the impression to a guaranteed contract. Note that (similar to the situation in the first stage) after the second-stage decisions have been made, each impression has to be allocated and only (i) or (ii) can hold for a particular impression.

In our model, we assume without loss of generality, that the sales to SSP's are always successful in the second stage. In other words, we assume that there is a set of 'risky SSPs' and 'safe SSPs' and in the second stage allocations to 'risky SSPs' are not allowed. This assumption ensures that no impression remains unsold after the second-stage decisions have been made.

Example 2.2. *Consider Example 2.1. We now illustrate a feasible allocation of impressions that is consistent with the two-stage stochastic programming framework. At each stage we need to determine how to allocate impressions to SSPs and guaranteed contracts in order to end up with a feasible allocation.*
Table 2.3 shows the first-stage decisions and Table 2.4 shows the second-stage decisions corresponding to a feasible allocation (i.e, 1 represents an allocation). Here we observe that initially (in the first stage) the publisher attempts to sell the impres-

Table 2.3: First-stage allocation of impressions.

	slot 1	slot 2	slot 3
SSP 1	0	1	1
SSP 2	0	0	0
Contract 1	1	0	0
Contract 2	0	0	0

Table 2.4: Second-stage allocation of impressions.

	Scenario 1: probability 0.7			Scenario 2: probability 0.3		
	slot 1	slot 2	slot 3	slot 1	slot 2	slot 3
SSP 1	0	0	0	0	0	0
SSP 2	0	0	0	0	0	0
Contract 1	0	0	0	0	0	0
Contract 2	0	0	1	0	1	0

sions associated with ad-slot 3 on SSP 1. However, if scenario 1 occurs, the publisher learns that SSP 1 was not able to sell the impression and the impression is re-allocated to guaranteed contract 2 (in the second stage). Note that no impressions can be allocated to SSP 1 in the second stage since it is a risky SSP and because the publisher does not want to end up with unsold inventory. □

2.4.2 Two-Stage Stochastic Programming Model

Based on the characteristics of the display-ad allocation problem presented so far, we now present a two-stage stochastic programming formulation of the problem.

Decision variables There are two types of decision variables: first-stage and second-stage. The first-stage decision variables are given by:

$$x_{nmc} \in \{0,1\} \quad \forall n \in N, m \in M_n, c \in C$$
$$x_{nms} \in \{0,1\} \quad \forall n \in N, m \in M_n, s \in S$$

$x_{nmc} = 1$ indicates that impression m on webpage n has been allocated to contract c, and $x_{nms} = 1$ indicates that impression m on webpage n has been allocated to SSP s. The second-stage decision variables are given by:

$$y^{\omega}_{nmc} \in \{0,1\} \quad \forall n \in N, m \in M_n, c \in C, \omega \in \Omega$$
$$y^{\omega}_{nms} \in \{0,1\} \quad \forall n \in N, m \in M_n, s \in S^{Safe}, \omega \in \Omega$$

$y^{\omega}_{nmc} = 1$ indicates that impression m on webpage n has been allocated to contract

c in scenario ω, and $y^{\omega}_{nms} = 1$ indicates that impression m on webpage n has been allocated to SSP s in scenario ω.

Objective function The objective function is given by:

$$\begin{aligned} \max \quad z = & \sum_{w \in \Omega} \mathbb{P}(\omega) \cdot \left(\sum_{n \in N} \sum_{m \in M_n} \sum_{s \in S} (x_{nms} \cdot R_{nms} \cdot H^{\omega}_{nms}) \right) \\ & + \sum_{w \in \Omega} \mathbb{P}(\omega) \cdot \sum_{n \in N} \sum_{m \in M_n} \sum_{s \in S^{Safe}} (y^{\omega}_{nms} \cdot R_{nms}) \\ & + \sum_{w \in \Omega} \mathbb{P}(\omega) \cdot \left(\sum_{c \in C} \left(\sum_{n \in N} \sum_{m \in M_n} (y^{\omega}_{nmc} + x_{nmc}) \right) \lambda_c \right) \end{aligned} \tag{2.1}$$

The first line in (2.1) represents the expected revenue from the first-stage allocations to SSPs. Note the role that the random variable H^{ω}_{nms} plays in the first term: if SSP s fails to sell impression m on webpage n in scenario ω, then $H^{\omega}_{nms} = 0$ and there is no revenue. The second line in (2.1) represents the expected revenue from the second-stage allocation of impressions to SSPs. Notice that an SSP can always sell the impression in the second stage because it is a safe SSP and hence there is no randomness. The third line in (2.1) represents the expected revenue from allocating impressions to the guaranteed contracts.

Constraints The constraints in the display-ad allocation problem are as follows.

$$\sum_{s \in S} x_{nms} + \sum_{c \in C} x_{nmc} = 1 \quad \forall n, m \tag{2.2}$$

$$\sum_{n \in N} \sum_{m \in M_n} (x_{nmc} + y^{\omega}_{nmc}) = \alpha_c \quad \forall c, \omega \tag{2.3}$$

$$\sum_{s \in S} (x_{nms} \cdot H^{\omega}_{nms}) + \sum_{c \in C} x_{nmc} + \sum_{s \in S^{Safe}} y^{\omega}_{nms} + \sum_{c \in C} y^{\omega}_{nmc} = 1 \quad \forall n, m, \omega \tag{2.4}$$

$$\begin{aligned} x_{nmc} &\in \{0,1\} \quad \forall n \in N, m \in M_n, c \in C \\ x_{nms} &\in \{0,1\} \quad \forall n \in N, m \in M_n, s \in S \\ y^{\omega}_{nmc} &\in \{0,1\} \quad \forall n \in N, m \in M_n, c \in C, \omega \in \Omega \\ y^{\omega}_{nms} &\in \{0,1\} \quad \forall n \in N, m \in M_n, s \in S^{Safe}, \omega \in \Omega \end{aligned} \tag{2.5}$$

Constraint (2.2) represents the trade-off between guaranteed contracts and SSPs in the first-stage decisions. This constraint indicates that impression m on webpage n has to be allocated to precisely one SSP or has to be allocated to precisely one contract. Constraint (2.3) indicates that the number of impressions that are allocated to a contract type, must meet a certain target value in the planning period under

consideration, and this needs to hold under all scenarios. Note that the target values α_c are set by the publisher in such a way that, over the course of the contract period, enough impressions are allocated to each contract. Since the publisher does not receive any additional revenue for impressions allocated beyond the contracted amount, we have an equality in Constraint (2.3). Constraint (2.4) is similar to Constraint (2.2) and represents the trade-off between guaranteed contracts and SSPs in the second-stage decisions. This constraint ensures that after the second-stage decision has been made, impression m on webpage n has to be allocated to precisely one SSP or has to be allocated to precisely one contract, and this needs to hold under all scenarios. It thus links the first-stage decisions with the second-stage decisions. More specifically, this contraint implies that if impression m on webpage n was allocated to any contract c, then it cannot be allocated to an SSP or guaranteed contract in the second stage (and therefore $\sum_{s\in S^{Safe}} y^{\omega}_{nms} + \sum_{c\in C} y^{\omega}_{nmc} = 0$). Similarly, if in the first stage impression m on webpage n was allocated to any SSP s and the impression was sold by the SSP (such that $\sum_{s\in S} x_{nms} \cdot H^{\omega}_{nms} = 1$), then it is again not possible to allocate the impression to an SSP or guaranteed contract in the second stage (and therefore $\sum_{s\in S^{Safe}} y^{\omega}_{nms} + \sum_{c\in C} y^{\omega}_{nmc} = 0$). Finally, Constraint (2.5) states that the decision variables are binary.

The model formulation using Constraints (2.3) and (2.4) reflects the preference of the publisher to fulfill the guaranteed contracts in all scenarios in order to maintain a good reputation among advertisers. Nevertheless, the formulation can be adjusted in a straightforward way in order to allow allocations to risky SSPs in the second stage, i.e. by modifying Constraint (2.3) so that at least a certain percentage of guaranteed contracts are fulfilled.

In our model we assume that the online publisher has specified the scenarios in such a way that given a choice of first-stage variables the online publisher will be able to deduce the scenario that has occurred in the second stage. If this is not the case, then so-called nonanticipativity constraints should be added to the formulation. More specifically, let $\omega_1, \omega_2 \in \Omega$ with $\omega_1 \neq \omega_2$. If $\sum_{s\in S, n\in N, m\in M_n} x_{nms} \cdot |H^{\omega_1}_{nms} - H^{\omega_2}_{nms}| = 0$, then $y^{\omega_1}_{nms} = y^{\omega_2}_{nms}$ $\forall s \in S, n \in N, m \in M_n$. That is, if given a choice of first-stage variables the online publisher cannot distinguish between two scenarios, then the second-stage decisions should be the same in each of these two scenarios.

2.4.3 Bounding the objective value of the SP model

In order to assess the solution quality of the SP model, we calculate the objective value of the solution of the so-called 'Expected Value problem' (ExpVal) and the objective value of the solution of the so-called 'Wait-and-see problem' (WS).
The objective value of ExpVal is calculated as follows. First, we replace the random variables by their expected value, that is, we calculate the expectation of the random variables H^{ω}_{nms} given by $\bar{H}^{+}_{nms} = \sum_{\omega\in\Omega} \mathbb{P}(\omega) \cdot H^{\omega}_{nms}$. Next, we solve the model given by (2.1)-(2.5) by setting $|\Omega| = 1$ and replacing the random variables with the values of

$\bar{H}^{+}_{nms}$. Let x_{EV} denote the values of the first-stage variables in the resulting solution. For each $\omega \in \Omega$ let EV_ω denote the objective value that is obtained by solving the model given by (2.1)-(2.5) with $|\Omega| = 1$ and imposing x_{EV}. The objective value of ExpVal can now be calculated as $EV = \sum_{\omega \in \Omega} \mathbb{P}(\omega) \cdot EV_\omega$.
In the WS problem we calculate the optimal allocation given perfect information (that is, prior knowlegde of $\omega \in \Omega$). Given the realizations of a specific scenario $\omega \in \Omega$, we calculate the optimal allocation of impressions over SSPs and the guaranteed contracts. Let us denote the obtained objective value by WS_ω. Next, we calculate the expectation over the distribution of the scenarios in order to get the objective value of the WS problem, that is, $WS = \sum_{\omega \in \Omega} \mathbb{P}(\omega) \cdot WS_\omega$. Note that, given a specific scenario $\omega \in \Omega$, the display-ad allocation problem reduces to a much simpler allocation problem since it is known which SSPs will be able to sell the impressions allocated to them. Intuitively, WS is an upperbound for the objective value of SP since the optimal allocation can be determined for each scenario $\omega \in \Omega$ whereas in the SP model there are first-stage variables that force some part of the allocation to be the same across all scenarios. This comparison between WS, ExpVal and SP is useful since the objective value of ExpVal is a lowerbound and the objective value of WS is an upperbound of the objective value for SP, see for example [30]. Note that in ExpVal only the expected scenario is used in order to determine the first-stage decisions (and not the individual scenarios themselves) and, in the second stage, decisions are made depending on the observed scenario. If the gap between the objective value of ExpVal and WS is large, then this is an indication that it might be worthwhile to solve SP (which is more complicated to solve) since taking the individual scenarios themselves into consideration leads to additional benefits compared to just looking at the expected scenario.

2.5 Benchmark algorithm: Priority Assignment (PA) heuristic

In this section we discuss a practical allocation policy that is used in practice by our industry partner. We refer to this practical allocation policy as the Priority Assignment (PA) heuristic.

The PA heuristic is based on the idea that the fulfillment of guaranteed contracts has priority during the planning period. As impressions are generated by users, the publisher first focusses on achieving the targets for the guaranteed contracts and after these targets have been reached, the remaining impressions that are generated during the planning period are allocated to SSPs. More specifically, we first select a random subset of the webpages and use the impressions of these webpages in order to meet the demands of the guaranteed contracts. Given this allocation to the guaranteed contracts, we then proceed to allocate the remaining impressions to the SSPs for each specific scenario such that the revenue is maximized. Let us denote the obtained

objective value under scenario $\omega \in \Omega$ by PA_ω. Finally, we take the expectation over the distribution of the scenarios in order to calculate the expected revenue of the PA heuristic, that is, $PA = \sum_{\omega\in\Omega} \mathbb{P}(\omega) \cdot PA_\omega$. The PA heuristic is also called a 'waterfall' approach in practice, because typically the publisher ranks the SSPs (based on (expected) revenue) and allocates the remaining impressions according to this ranking in order to maximize revenue. The pseudocode for the PA heuristic is provided in Algorithm 2.1.

Algorithm 2.1 Pseudocode for PA heuristic

Require: A set of scenarios Ω, SSPs S, guaranteed contracts C, webpages N, impressions $M_i,\ i \in N$, revenues $R_{nms},\ n \in N, m \in M_n, s \in S$, targets for guaranteed contracts $\alpha_c, c \in C$, sale outcomes $H^\omega_{nms},\ n \in N, m \in M_n, s \in S, \omega \in \Omega$ and a probability distribution $\mathbb{P}(\omega),\ \omega \in \Omega$.

Make a list L where element $L[i] = i,\ i = 1, \ldots, |N|$ represents webpage $i \in N$.
Make a random permutation P of the list L.
Set $N^* := \inf\{m \in \mathbb{Z} \mid \sum_{i=1}^{m} |M_{P[i]}| \geq \sum_{c\in C} \alpha_c\}$. {Random selection of visitors.}
Set $G^* := \{P[1], \ldots, P[N^*]\}$. {Random selection of webpages.}
for $\omega \in \Omega$ **do**
    Set $PA_\omega := 0$.
    for $i \in G^*$ **do**
        Allocate impressions from webpage $i \in G^*$ to the guaranteed contracts until the demands of all the guaranteed contracts have been met.
    end for
    if Some subset $\bar{M} \subseteq M_{P[N^*]}$ of impressions from webpage $P[N^*]$ have not been allocated. **then**
        for $k \in \bar{M}$ **do**
            Allocate impression k from webpage $P[N^*]$ to the SSP s^* with the highest return such that $H^\omega_{P[N^*],k,s^*} = 1$ and collect revenue $R_{P[N^*],k,s^*}$.
            Set $PA_\omega := PA_\omega + R_{P[N^*],k,s^*}$
        end for
    end if
    Set $PA^C := \sum_{c\in C} \alpha_c \cdot \lambda_c$. {Revenue from guaranteed contracts.}
    for $i \in N \setminus G^*$ **do**
        for $k \in M_i$ **do**
            Allocate impression k from webpage i to the SSP s^* with the highest return such that $H^\omega_{i,k,s^*} = 1$ and collect revenue R_{i,k,s^*}.
            Set $PA_\omega := PA_\omega + R_{i,k,s^*}$
        end for
    end for
    Set $PA_\omega := PA_\omega + PA^C$ {Revenue of PA heuristic in scenario ω.}
end for
Set $PA := \sum_{\omega\in\Omega} \mathbb{P}(\omega) \cdot PA_\omega$.
return PA

2.6 Experiments

In this section we construct various instances of the display-ad allocation problem and run experiments in order to assess the performance of the SP model. We measure the performance using the expected revenue relative to an upperbound and relative to a practical allocation policy (the PA heuristic) that is used in practice by our industry partner.

2.6.1 Setup of experiments

We consider the situation where there are 10000 webpages that each have 3 ad-slots. We assume that there are 3 SSPs, one of which is a 'safe' SSP and we assume that there are 3 guaranteed contracts, i.e., $|N| = 10000, |M_n| = 3 \ \forall n \in N, |S| = 3, |S^{Safe}| = 1$ and $|C| = 3$. In the experiments, the values for α_c are given by:

$$\alpha_c = \left\lfloor \gamma \cdot \frac{|N| \cdot |M|}{|C|} \right\rfloor, \quad 0.1 \leq \gamma \leq 0.9 \ \ \forall c \in C \tag{2.6}$$

The expression for α_c indicates that a fraction γ of all available impressions is distributed equally over the $|C|$ guaranteed contracts.

Generating returns for each ad-slot In order to generate the returns for each ad-slot/impression we assume that the revenue from the sale by the SSPs are normally distributed. More specifically we have the following return structure for website n.

$$R_n = \begin{bmatrix} N(\mu_1, \sigma_1) & N(\mu_2, \sigma_1) & N(\mu_3, \sigma_1) \\ N(\mu_4, \sigma_2) & N(\mu_5, \sigma_2) & N(\mu_6, \sigma_2) \\ N(\mu_7, \sigma_3) & N(\mu_8, \sigma_3) & N(\mu_9, \sigma_3) \\ \kappa_1 & \kappa_1 & \kappa_1 \\ \kappa_2 & \kappa_2 & \kappa_2 \\ \kappa_3 & \kappa_3 & \kappa_3 \end{bmatrix}$$

The columns denote the 3 ad-slots that are associated with the impressions that need to be allocated. The first 3 rows of R_n are the revenues for the 3 SSPs, where the third row gives the revenues for the safe SSP. The last 3 rows of R_n are the revenues for the 3 guaranteed contracts. The relationship between the standard deviations are given by $\sigma_2 = 0.85\sigma_1$ and $\sigma_3 = 0.65\sigma_1$ and we set $\sigma_1 = 3$. The difference in standard deviations is meant to model the heterogeneity in the distribution of revenues and to capture the property that the revenues from "safe" SSPs are less variable. The returns for the other webpages are independent and identically distributed (i.i.d.) with the same distribution as R_n. The random differences in revenues between R_{n+1} and R_n captures the property that some webpages generate more revenue than others.

Generating the scenarios Initially we assume that the realizations of the random variables in the different scenarios are generated from an i.i.d. Bernoulli distribution with success parameter equal to p. More specifically, for a scenario ω we can encode the random variables as follows:

$$H^{+}_{\omega n} = \begin{bmatrix} H^{\omega} \\ 1_{|S^{Safe}|} \\ 1_{|C|} \end{bmatrix} \quad \forall \omega, n \tag{2.7}$$

Here H^{ω} is an $|S^{Risky}| \times |M|$ matrix where each element is a draw from an i.i.d. random variable that follows a Bernoulli distribution with parameter p. $1_{|S^{Safe}|}$ is a $|S^{Safe}| \times |M|$ matrix where each element equals 1. $1_{|C|}$ is a $|C| \times M$ matrix where each element equals 1. The interpretation of the resulting matrices $H^{+}_{\omega n}$ is similar to the description given by Table 2.2 in Example 2.1. During the experiments we also test the sensitivity of the results with respect to the independence assumption and the probability of success p.

Generating the probability distribution of the scenarios The probability distribution of the scenarios for the base case are generated as follows:

$$\mathbb{P}(\omega_v) = \frac{I_v}{\sum_{v=1}^{v=|\Omega|} I_v} \tag{2.8}$$

Here I_v is a draw from an i.i.d. random variable that follows a uniform distribution on the interval (0,1). In this specification we assume that each scenario is equally likely in expectation. We also consider the case where one particular scenario is more likely than others. By adjusting the distribution and the success probability we can model a situation where there is a high probability scenario that in general most SSPs (if p is low) or few SSPs (if p is high) are able to sell impressions.

Set of cases to consider In the experiments we consider four cases: Case A, B, C and D. In our experiments, we are interested in analyzing how the performance of the SP model and PA heuristic depends on: (i) revenues from SSPs and guaranteed contracts; (ii) success probabilities in different scenarios; (iii) correlation of success probabilities between SSPs.

- Impact of revenues. In Case A and B we analyze the impact of revenues on the performance of the algorithms. Case A represents a situation that is commonly observed in practice where guaranteed contracts have a higher revenue compared to ad slots sold on SSPs. In Case A we assume that the mean revenues from the different SSPs are relatively close in expectation except for the safe SSP. In Case B we allow for more overlap between revenues from the guaranteed contracts and SSPs.

- Impact of scenarios. In Case C we adjust the probability distribution of the scenarios such that a particular scenario $\omega_1 \in \Omega$ has a higher probability of occurring. In this case we model a situation where there is a high probability scenario that in general most SSPs or few SSPs are able to sell impressions.

- Impact of correlation. One might expect the ability of SSPs to sell impressions to be correlated, since advertisers can be connected to multiple SSPs. Therefore, in Case D we relax the assumption of independence when generating Bernoulli random variables in $H^+_{\omega n}$.

Parameter settings of cases In the experiments we use the following parameter settings for the cases.

- Case A. For the revenues in R_n we set $\mu_1 = \mu_2 = \mu_3 = 15$, $\kappa_1 = 35$, $\kappa_2 = 30$, $\kappa_3 = 25$. In order to generate the random variables for the scenarios, we use a Bernoulli distribution with parameter $p = 0.5$ in $H^+_{\omega n}$. The probability distribution of the scenarios is generated according to Eqn. (2.8). We investigate Case A with $|\Omega| = 4$ and $|\Omega| = 8$ and for various values of γ from the range specified in Eqn. (2.6).

- Case B. This case is similar to Case A except for the revenues such that we allow for more overlap between revenues from the guaranteed contracts and SSPs. In Case B we set $\mu_1 = 25$, $\mu_2 = 20$, $\mu_3 = 10$, $\kappa_1 = 35$, $\kappa_2 = 25$, $\kappa_3 = 20$.

- Case C. We now adjust the probability distribution of the scenarios. We bias the distribution such that a particular scenario ω_1 has a higher probability of occurring. We use the following distribution $\mathbb{P}(\omega_v) = \frac{I_v}{\sum_{v=|\Omega|} I_v}$. Here I_v is a draw from an i.i.d. random variable that follows a uniform distribution on the interval $(0.6, 1)$ for $v = 1$ and on $(0, 0.2)$ for $v \neq 1$. In order to generate the random variables for scenario ω_1 we use a Bernoulli distribution with parameter $p^* \in \{0.2, 0.8\}$ in $H^+_{\omega n}$ and for the remaining scenarios we use $p = 0.5$. These settings model a situation where there is a high probability scenario that in general most SSPs (if $p^* = 0.8$) or few SSPs (if $p^* = 0.2$) are able to sell impressions. The other settings are similar to Case B.

- Case D. In Case D we relax the assumption of independence when generating Bernoulli random variables in $H^+_{\omega n}$. We instead let the linear correlation between risky SSPs vary from 0.1 to 0.9. The other settings are similar to Case C.

Motivation for parameters in experiments Publishers typically update their allocation decisions every hour, and for each domain separately, as they have hourly estimates of revenues and website traffic. Typically there are 2 - 3 slots on a wegpage (above the fold, below the fold and next to a video being displayed). Based on data

from our industry partner, there are about 8000 - 10000 visitors on a particular domain (which has a number of webpages) per hour. Therefore the choice of 10000 webpages and 3 ad slots in our experiments is realistic. In our experiments we considered cases with 4 and 8 scenarios. As there are about 3 slots on a webpage, there are $2^3 = 8$ possibilties for these slots to be sold on a particular SSP. Based on empirical data, it is often the case that only a subset of these 8 cases (about 2-3) take up most of the probability mass. Therefore, for two risky SSPs there are about 4 - 9 possible scenarios to consider for each arriving visitor of a webpage. Therefore the choice for 4 and 8 scenarios is reasonable.

General scheme for generating instances The general scheme for generating multiple instances is presented in Algorithm 2.2.

Algorithm 2.2 Pseudocode for generating instances

Require: a case from $\mathcal{C} = \{A, B, C, D\}$
  for $i = 1$ **to** 30 **do**
    Step 1: Generate the returns R_n for each AD-slot.
    Step 2: Generate the scenarios using Eqn. (2.7).
    Step 3: Generate the probability distribution of the scenarios using Eqn. (2.8).
    Step 4: Calculate the objective values of WS, SP, PA and ExpVal.
    Step 5: Scale the objective values of SP, PA and ExpVal by the objective value of WS, i.e. divide SP, PA and EV by WS.
  end for
  return Objective values of SP, PA and ExpVal for 30 instances scaled by objective value of WS.

2.6.2 Results of the experiments

In this section, we report about computational experiments to evaluate the different models. The algorithms are coded in Python 2.7 and run on Intel(R) Core(TM) CPU i5-6300U @ 2.40GHz with 8GB RAM under Windows 7 environment. The mathematical formulation (Equations 2.1-2.5) presented in Section 2.4 is solved by calling IBM ILOG CPLEX 12.6.1 from Python.

Results for Case A and B

We solve 30 instances of Case A. The results are presented in Table 2.5 and Figure 2.2. The results indicate that the SP model performs very well in the sense that it is very close to the best possible solution with a performance gap relative to WS of about 2-3% in most cases.

One of the key parameters of the model is the fraction of impressions that need to be allocated to the guaranteed contracts. Both the gap between the objective value of ExpVal and WS, and the performance gap between the PA heuristic and SP

varies considerably with this parameter. If the fraction of impressions that need to be allocated to guaranteed contracts is relatively low, the gap between the objective value ExpVal and WS is larger and this gap decreases as more impressions need to be allocated to guaranteed contracts. If very few contracts need to be allocated to guaranteed contracts, the PA heuristic outperforms the SP model slightly. The largest performance gap between SP model and the PA heuristic is when a moderate fraction (around 40% - 50%) of the impressions are allocated to guaranteed contracts. In this case the gap is about 8% - 10%.

Table 2.5: Statistics of SP, PA and ExpVal solutions for Case A and B.

Panel A: results for Case A										
	$\|\Omega\| = 4$ and $\gamma = 0.4$					$\|\Omega\| = 4$ and $\gamma = 0.8$				
	min	max	mean	median	std	min	max	mean	median	std
SP	98.64	99.06	98.86	98.85	0.13	99.98	99.99	99.98	99.98	0.00
PA	90.04	90.27	90.15	90.14	0.06	95.65	95.80	95.71	95.70	0.03
ExpVal	76.98	77.40	77.17	77.17	0.08	95.31	95.50	95.39	95.39	0.04
	$\|\Omega\| = 8$ and $\gamma = 0.4$					$\|\Omega\| = 8$ and $\gamma = 0.8$				
	min	max	mean	median	std	min	max	mean	median	std
SP	97.66	98.41	98.00	98.02	0.14	99.97	99.97	99.97	99.97	0.00
PA	90.08	90.24	90.15	90.15	0.04	95.66	95.76	95.71	95.71	0.03
ExpVal	72.18	72.39	72.29	72.28	0.05	92.18	92.26	92.22	92.22	0.02
Panel B: results for Case B										
	$\|\Omega\| = 4$ and $\gamma = 0.4$					$\|\Omega\| = 4$ and $\gamma = 0.8$				
	min	max	mean	median	std	min	max	mean	median	std
SP	98.60	99.14	98.88	98.86	0.16	100.00	100.00	100.00	100.00	0.00
PA	88.94	89.17	89.05	89.03	0.06	94.07	94.26	94.16	94.15	0.04
ExpVal	73.85	74.29	74.03	74.03	0.08	93.23	93.50	93.36	93.36	0.07
	$\|\Omega\| = 8$ and $\gamma = 0.4$					$\|\Omega\| = 8$ and $\gamma = 0.8$				
	min	max	mean	median	std	min	max	mean	median	std
SP	97.51	98.34	97.88	97.88	0.16	100.00	100.00	100.00	100.00	0.00
PA	88.99	89.13	89.05	89.06	0.04	94.09	94.20	94.15	94.15	0.03
ExpVal	68.62	68.82	68.71	68.71	0.04	88.97	89.06	89.02	89.02	0.02

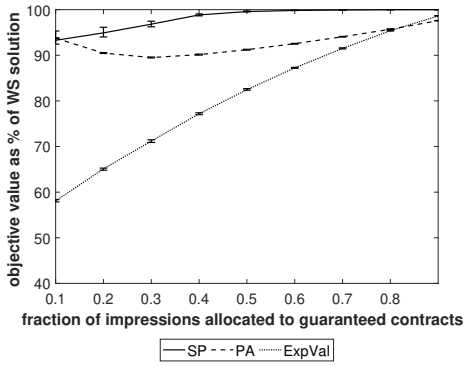
Notes: This table shows the results over 30 experiments with the settings of Case A and B. The rows with SP, PA and ExpVal report the objective value of SP, PA and ExpVal as a percentage of the objective value of WS.

In order get a better understanding of this relationship, Figure 2.2 (panels (a) - (d)) shows how various choices for γ affects the performance of the models. Here we see that the SP model performs well. In particular, for $\gamma \geq 0.5$ the objective value of the SP model is within 0.5 % of the objective value of WS. The PA heuristic displays a non-linear relationship with γ. This can be explained by noting that for $\gamma \approx 0$ the PA heuristic boils down to selling each impression on the SSP with the highest revenue

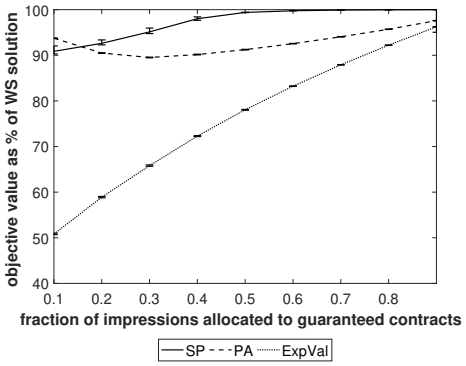
and this is why it outperforms SP for small values of γ. For $\gamma \approx 1$ the total revenues are almost completely determined by the revenues for the guaranteed contracts, which are fixed and known in advance and so in this case the revenue from WS and PA are also close to each other. From Figure 2.2 we also observe that the performance of ExpVal shows a clear increasing relationship with γ. For small values of γ, the first-stage solution from solving the expected value problem is likely to be sub-optimal for the individual scenarios and since most impressions are to be allocated to SSPs, this stochastic component of the problem has a relatively large impact. When γ is large, this stochastic component does not play a large role and the objective value of WS and ExpVal are closer to each other.

The overall results for Case B are very similar to Case A. For $\gamma = 0.4$ we now observe a gap between SP and PA of about 8-9% which is an increase of about 1% relative to Case A. For $\gamma = 0.8$ we see a similar pattern.

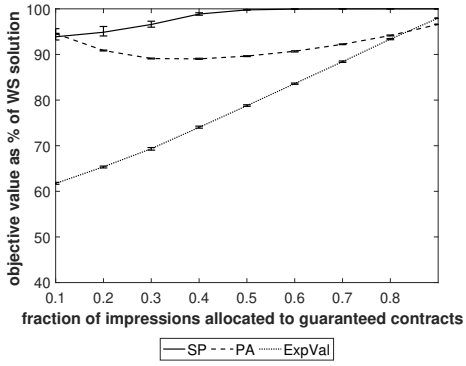
We also investigate the relationship between the running time in CPLEX of the SP model and the problem size (i.e., in terms of the number of websites $|N|$ and the number of scenarios $|\Omega|$). The results in Figure 2.2 (panel (e)) indicate that for a fixed number of scenarios with $|\Omega| = 16$, the running time is roughly linear in the number of websites $|N|$. Figure 2.2 (panel (f)) shows how the running time varies with the number of scenarios $|\Omega|$ for a fixed number of websites ($|N| = 10000$). The pattern suggests that for a fixed number of websites, the running time is polynomial in the number of scenarios. The results indicate that the SP model can be solved in a reasonable amount of time (within 10 minutes) for instances with realistic problem sizes (16 scenarios and 12000 visitors on a domain). Publishers typically update their allocation decisions every hour and for each domain separately as they have hourly estimates of revenues and website traffic. Based on data from our industry partner, there are about 8000 - 10000 visitors on a particular domain per hour. Therefore our model is able to handle real-world cases.



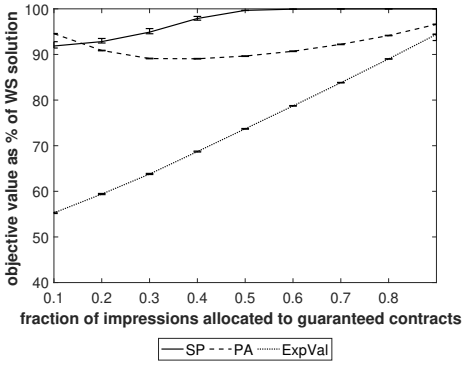
(a) Case A, $|\Omega| = 4$.



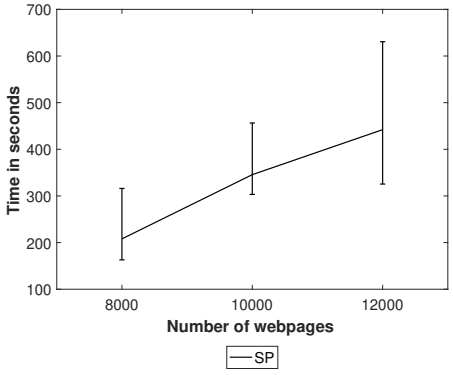
(b) Case A, $|\Omega| = 8$.



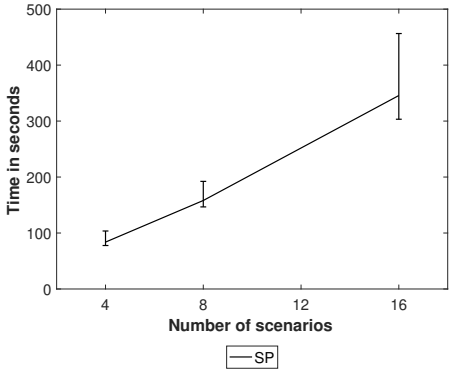
(c) Case B, $|\Omega| = 4$.



(d) Case B, $|\Omega| = 8$.



(e) Running time and number of websites.



(f) Running time and number of scenarios.

Figure 2.2: Panel (a) - (d): objective value of SP, PA and ExpVal relative to WS objective value over 30 experiments with settings of case A and B for different values of γ. Panel (e): running time of SP with settings of case A for a different number of websites $|N|$ and $|\Omega| = 16$. Panel (f): running time of SP with settings of case A for a fixed number of websites $|N| = 10000$ and for a different number of scenarios $|\Omega|$ with $\gamma = 0.4$. Error bars indicate the maximum, median and minimum values.

Results for Case C and D

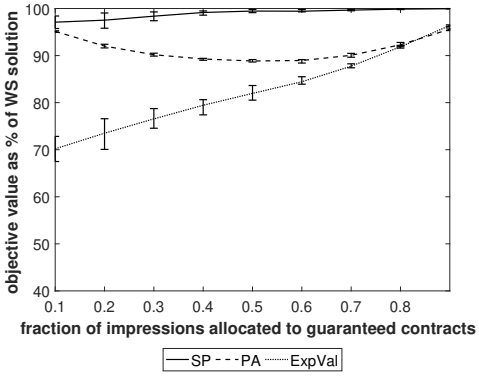
Figure 2.3 (panels (a) - (d)) shows results for the settings of Case C. We see that the performance of SP does not change considerably since the gap between SP and WS is at most about 2-3% for most cases considered and is hardly affected by the value of p^*. The performance of PA improves for higher values of p^*. In Table 2.6 we see that for $p^* = 0.8$ the performance of PA is about 3% higher than for $p^* = 0.2$. For example, if $|\Omega| = 4$, $\gamma = 0.4$ and $p^* = 0.2$ then the gap between PA and SP based on the median is about 8%-9% compared a gap of 5%-6% with $|\Omega| = 4$, $\gamma = 0.4$ and $p^* = 0.8$. The finding that PA performs better for higher values of p^* can be explained by noting that higher values of p^* imply that the negative impact of the initial allocation to guaranteed contracts has a smaller contribution to the total expected revenue, compared with the case of lower values of p^*.

In Case D we relax the assumption that the ability of SSPs to sell the impressions are independent from each other. Figure 2.3 (panels (e) - (f)) shows results for different values of γ and for $|\Omega| = 4$. The lines in the figures are relatively flat (with sometimes a slightly negative trend) indicating that the performance does not vary systematically with the correlation between SSPs. The results for other parameter values and for $|\Omega| = 8$ are very similar.

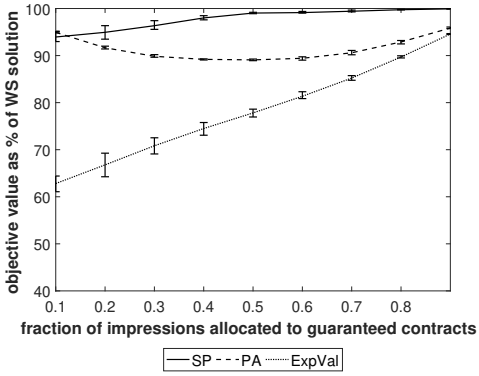
Table 2.6: Statistics of SP, PA and ExpVal solutions for Case C.

	$\|\Omega\| = 4$, $\gamma = 0.4$ and $p^* = 0.2$					$\|\Omega\| = 4$, $\gamma = 0.8$ and $p^* = 0.2$				
	min	max	mean	median	std	min	max	mean	median	std
SP	98.59	99.49	99.13	99.15	0.26	99.77	99.95	99.87	99.87	0.04
PA	89.00	89.41	89.25	89.27	0.10	91.89	92.79	92.35	92.30	0.22
ExpVal	77.40	80.63	79.29	79.44	0.87	91.59	92.17	91.89	91.87	0.15
	$\|\Omega\| = 8$, $\gamma = 0.4$ and $p^* = 0.2$					$\|\Omega\| = 8$, $\gamma = 0.8$ and $p^* = 0.2$				
	min	max	mean	median	std	min	max	mean	median	std
SP	97.59	98.48	98.01	98.00	0.24	99.70	99.83	99.75	99.74	0.03
PA	89.06	89.30	89.20	89.20	0.05	92.48	93.20	92.88	92.89	0.15
ExpVal	73.08	75.78	74.47	74.50	0.66	89.49	89.95	89.69	89.69	0.10
	$\|\Omega\| = 4$, $\gamma = 0.4$ and $p^* = 0.8$					$\|\Omega\| = 4$, $\gamma = 0.8$ and $p^* = 0.8$				
	min	max	mean	median	std	min	max	mean	median	std
SP	97.93	99.05	98.55	98.56	0.31	99.82	99.96	99.89	99.89	0.04
PA	92.03	93.28	92.76	92.81	0.33	95.37	96.03	95.72	95.74	0.17
ExpVal	74.83	75.48	75.11	75.08	0.19	95.14	95.51	95.33	95.32	0.09
	$\|\Omega\| = 8$, $\gamma = 0.4$ and $p^* = 0.8$					$\|\Omega\| = 8$, $\gamma = 0.8$ and $p^* = 0.8$				
	min	max	mean	median	std	min	max	mean	median	std
SP	96.97	97.91	97.34	97.27	0.28	99.74	99.85	99.78	99.77	0.02
PA	91.27	92.37	91.75	91.68	0.31	94.99	95.64	95.25	95.26	0.13
ExpVal	67.21	67.84	67.50	67.50	0.17	88.77	88.99	88.90	88.90	0.05

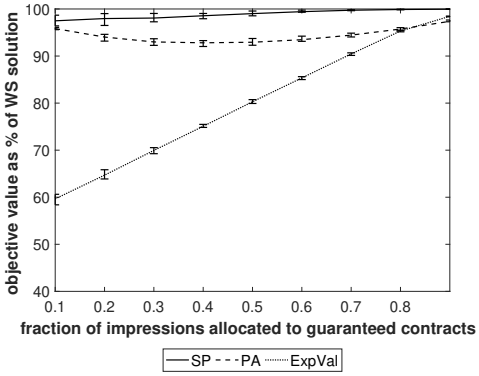
Notes: This table shows the results over 30 experiments with the settings of Case C. The rows with SP, PA and ExpVal report the objective value of SP, PA and ExpVal as a percentage of the objective value of WS.



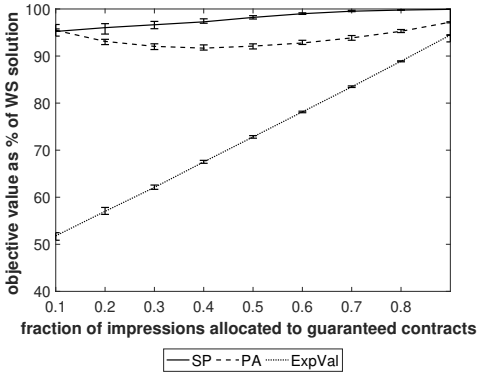
(a) Case C, $p^* = 0.2$ and $|\Omega| = 4$.



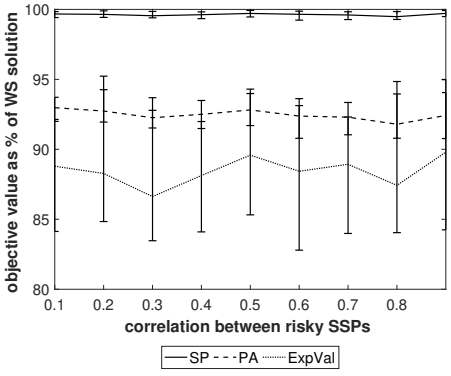
(b) Case C, $p^* = 0.2$ and $|\Omega| = 8$.



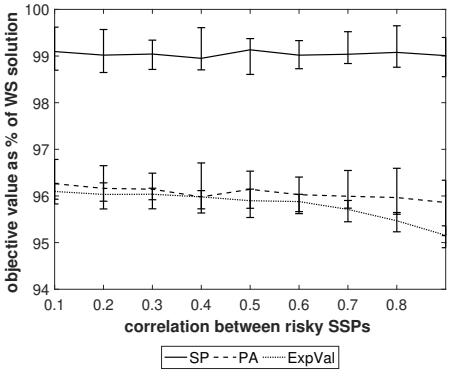
(c) Case C, $p^* = 0.8$ and $|\Omega| = 4$.



(d) Case C, $p^* = 0.8$ and $|\Omega| = 8$.



(e) Case D, $p^* = 0.2$, $|\Omega| = 4$ and $\gamma = 0.4$.



(f) Case D, $p^* = 0.8$, $|\Omega| = 4$ and $\gamma = 0.8$.

Figure 2.3: Objective value of SP and GH relative to WS objective value over 30 experiments with settings of case C (in panels (a) - (d)) and case D (in panels (e) - (f)) with $p^* = 0.2$ and $p^* = 0.8$. Error bars indicate the maximum, median and minimum values.

2.7 Conclusion

In this Chapter we model the display-ad allocation problem faced by online publishers as an integer linear programming problem. Our modeling approach takes the uncertainty associated with the sale of an impression by a SSP into account in the allocation of display ads. The way that information is revealed over time allows us to model the display-ad allocation problem as a two-stage stochastic program. We refer to our model as the Stochastic Programming (SP) model. We investigate the quality of the solutions obtained by the SP model in numerical experiments. We find that one of the key parameters of the model is the fraction of total impressions that need to be allocated to the guaranteed contracts. We show the SP model outperforms the allocation policy (Priority Assignment (PA) heuristic) that is used in practice by our industry partner. Furthermore, we find that the performance gap between the PA heuristic and SP model also shows a clear relationship with fraction of impressions that are allocated to the guaranteed contracts.

Our results suggest that the benefit of using the SP model is highest in periods where the website traffic is high compared with the targets for the guaranteed contracts. An example of such a situation is when there are seasonal patterns in the demand of advertisers. If advertisers have high demand for (premium) guaranteed campaigns for the month of December, then it is possible that a higher fraction of available impressions is sold on the auction market in April. Another example is when the website traffic varies depending on the time of the day. Consider the case of a publisher where most of the users are kids that visit pages to play online games. In this case it makes sense to allocate more impressions to guaranteed contracts in the afternoon hours and allocate much less during the late hours because the guaranteed contracts are primarily targeted towards kids.

In practice, publishers also need to make other decision problems which have not been discussed in this Chapter. For example, publishers typically specify a minimum price – called the *floor price* – for their impressions when selling on the RTB market. The next Chapter considers a situation where impressions are sold via second-price auctions and where the publisher needs to learn the best floor price over time.

Chapter 3

Setting reserve prices in second-price auctions with unobserved bids*

In this Chapter, we shift our attention to pricing decisions that publishers need to make. Publishers typically specify a minimum price – called the *floor price* – for their impressions when selling on the RTB market via an ad exchange. This Chapter considers a situation where impressions are sold via second-price auctions and where the publisher needs to learn the best floor price over time. In second-price auctions, the floor price is often referred to as the reserve price and so we refer to this problem as the reserve price optimization problem.

---

*This chapter is based on Rhuggenaath et al. [137, 138].

3.1 Introduction

One of the main mechanisms that web publishers use in online advertising to sell their advertisement space is real-time bidding (RTB) [161]. In RTB there are three main platforms: supply side platforms (SSPs), demand side platforms (DSPs) and an ad exchange (ADX) which connects SSPs and DSPs. The SSPs collect inventory of different publishers and thus serve the supply side of the market. Advertisers which are interested in showing online advertisements are connected to DSPs. When a user visits a webpage with an advertisement (ad) slot, the publisher sends a request to the ADX (via an SSP) indicating that an impression can potentially be displayed in this particular ad slot. At the same time, advertisers that are connected to DSPs send bid requests to the ADX indicating that they are willing to bid for this impression. A real-time auction then decides which advertiser is allowed to display its ad and the amount that the advertiser needs to pay.

Most of the ad inventory is sold via second-price auctions with a reserve price [123, 147, 161]. All bids below the reserve price are disregarded, and the auction has no winner if there are no bids remaining (i.e., there is a possibility that an ad slot is not sold). If the auction does have a winner, the winner pays the maximum of the second highest bid and the reserve price. In this Chapter, we study the reserve price optimization problem from the perspective of the publisher: the publisher submits his inventory of advertisement space to an SSP and needs to set the reserve price. In our problem the publisher does not observe the actual values of the winning bid and second highest bid. After each sale attempt on the RTB market, the publisher only knows whether the sale was successful and only observes the revenue that is received from that sale. This setting is relevant for publishers that are small and medium size enterprises (SMEs), since the ADX and connected SSPs typically do not reveal the actual bids placed in the auction but only the result of the auction. Due to the limited feedback, the publisher faces an exploration-exploitation trade-off. He needs to experiment with different reserve prices to figure out which one works best (exploration), but at the same time, he does not want to explore too much since he wants to use the best reserve price as much as possible (exploitation). This exploration-exploitation trade-off also arises in other settings, for example on auction sites such as eBay. In eBay auctions, items are sold via second-price auctions where sellers can set a reserve price and where sellers receive limited feedback, since the highest bid is not revealed after a sale (see e.g., [144]).

To the best of our knowledge, there is no algorithm available for the aforementioned reserve price optimization problem with unobserved bids that uses the rules of second-price auctions in order to set reserve prices. As we assume that the bids are not observed, previous approaches based on machine learning techniques such as in [18, 44, 123, 144, 147, 166] cannot be applied. Therefore, one approach is to formulate it as a standard multi-armed bandit (MAB) problem [37] where the arms are the different reserve prices that the publisher can choose, and the reward for each

arm is the revenue at a particular reserve price. For this general multi-armed bandit problem, there are existing algorithms with performance guarantees. However, these algorithms do not take the properties of second-price auctions into account in order to set the reserve prices. In the MAB algorithms each arm is treated as an independent option with an unknown mean reward and the fact that the options represent prices is not exploited. Furthermore, the algorithms do not take the link that exists in the second-price auction mechanism between the observed revenue and the reserve price into account. Our method, on the other hand, builds upon the multi-armed bandit framework and explicitly takes the properties of second-price auctions into account in order to set the reserve prices. By taking the properties of second-price auctions into account, our method aims to improve upon the performance of standard bandit algorithms.

The idea of improving multi-armed bandit algorithms by exploiting additional information about the structure of the problem at hand has been successfully applied in other settings. Some examples are, online pricing with discounted valuations [116], bandits with trends in the reward distribution [34] and bandits with graph structured feedback [40, 61, 115]. However, to the best of our knowledge, the idea combining MAB algorithms with second-price auctions has not been considered before. We summarize the main contributions of this Chapter as follows:

- We propose a method for learning reserve prices in a practically-relevant limited-information setting where the probability distribution of the bids from advertisers is unknown and the values of the bids are not revealed to the publisher. Furthermore, we do not assume that the publisher has access to a historical data set with bids.
- Our method incorporates knowledge about the rules of second-price auctions into a multi-armed bandit framework for optimizing reserve prices. To the best of our knowledge, we are the first to consider this combination.
- We introduce an extension of our proposed method for non-stationary environments.
- Experiments using real-life ad auction data show that the proposed method outperforms state-of-the-art bandit algorithms in both stationary and non-stationary environments.

The remainder of this Chapter is organized as follows. In Section 3.2 we discuss the related literature. Section 3.3 provides a formal description of the problem. In Section 3.4 we present our proposed method for setting reserve prices. In Sections 3.6 and 3.7 we perform experiments and compare our method with baseline strategies in order to assess the quality of our proposed method. Section 3.8 concludes this Chapter and provides some directions for further research.

3.2 Related Literature

We organize the related literature in three parts: reserve price optimization in second-price auctions, dynamic pricing with demand learning, and multi-armed bandits.

Reserve price optimization in second-price auctions

The problem of maximizing revenues in online advertising has received increasing attention in the machine learning literature over the last decade (see e.g. [1, 3, 18, 23, 24, 44, 161, 169]). Some studies use historical datasets containing the top two bids and supervised machine learning techniques in order to set the optimal reserve price. There are a number of studies that focus on incorporating the revenue function of the auction with a prediction model, most of the time by defining a surrogate loss function, in order to learn a mapping from features to a reserve price [18, 44, 123, 144, 147, 166]. In [169] a number of methods for reserve price optimization that use historical bidding data are compared and the authors propose an algorithm based on game theoretical arguments. As a key difference between this Chapter and the aforementioned works, we assume that the bids are not observed and that there is a no historical dataset available for the winning bid and/ or second highest bid.

In [42] an online learning approach is used to learn optimal reserve prices, but it is assumed that the number of bidders is known and that all the bids in an auction are independent draws from the same distribution. In this Chapter we do not make these assumptions. Other related work is [135, 136, 137]. These papers also assume that the values of the bids are not observed. However, different from them, we explicitly exploit properties and feedback from second-price auctions. Furthermore, the aforementioned papers make parametric assumptions on the distribution of the bids whereas we do not make such assumptions.

Finally, we note that there is some work on reserve price optimization in sponsored search advertising and keyword auctions which use another auction format called the generalized second-price auction (see e.g. [127, 151]) and reserve price optimization (in strategic settings) from the perspective of the organizer of auction (see e.g. [91]). To summarize, the main difference between the work in this Chapter and previous works are as follows. First, we do not assume that the publisher observes the top two bids in the ad auction, but only observes the revenue of each auction. Second, we explicitly take the properties of second-price auctions into account (and use a multi-armed bandit framework) in order to set reserve prices.

Dynamic pricing with demand learning

In general, the problem considered in this Chapter can be interpreted as a dynamic pricing problem with demand learning [63, 107]. In the standard dynamic pricing problem there is a seller who wants to maximize revenue over some selling horizon by choosing prices in an optimal way. However, the precise relationship between

price demand in unknown. This gives rise to the so-called exploration-exploitation trade-off.

There are different variations on the general dynamic pricing problem in terms of the assumptions regarding the inventory [65, 66], the demand model [28] and the market environment [64, 98, 105]. We refer the reader to [63] and [107] for a detailed overview of the dynamic pricing problem.

The main differences between the work in this Chapter and the most of literature on dynamic pricing is as follows. First, in a standard dynamic pricing problem, the seller chooses a price and the revenue of a sale (if the item is sold) equals the quoted price. However, in our problem, the seller needs to choose a reserve price and the revenue of a sale (if the item is sold) need not be equal to this reserve price. This implies that the revenue at a particular reserve price is a random variable, whereas in a standard dynamic pricing problem it is deterministic. Second, most of the dynamic pricing literature assumes that there is some functional form that relates prices and demand, however, we do not assume any functional form that relates the demand or sale probability at different prices. Third, in a dynamic pricing problem the seller usually interacts with a single buyer at a time, but in our problem the result of a sale depends on the (unobserved) bids from multiple bidders. Fourth, in terms of methodology, we explicitly take the properties of second price auctions into account in order to set reserve prices, and to the best of our knowledge, this is has not been studied before in the literature on dynamic pricing.

Multi-armed bandits

In terms of methodology, the work in this Chapter is related to previous work on the multi-armed bandit problem [17, 37] and studies that improve multi-armed bandit algorithms by exploiting additional information and structure in different problem settings. The idea of improving multi-armed bandit algorithms by exploiting additional information about the structure of the problem at hand has been successfully applied in a number of settings (see e.g. [34, 40, 61, 115, 116]). In [116, 157] and [121] bandit algorithms are used in dynamic pricing settings and they exploit properties of the demand curve and the valuations of buyers. In [34] bandits with trends in the reward distribution are studied and popular bandit algorithms are adapted in order to exploit this extra information. In [58, 130] bandits are studied where the arms have a specific graph structured unimodality property. In [40, 61, 115] bandits with side observations are studied. However, the techniques of the aforementioned papers cannot readily be applied to our problem. The algorithms in [116] and [157] are designed for pricing problems without a reserve price. Furthermore, the assumptions made by the algorithms in [40, 58, 61, 130], and [115] are not necessarily satisfied in our reserve price optimization problem. For example, in the case of bandits with side observations, the key assumption is that by pulling on some arm you get to observe independently and identically distributed (i.i.d.) draws are observed from another

arm. However, this is not the case in our problem. A related paper is [171] but their algorithms cannot be applied to our problem since they assume that the seller can see the bids and knows the number of participants in the auction.

3.3 Problem Statement

We consider a publisher that owns a single advertisement slot and sequentially sells impressions arriving over time. There are a number of *rounds* and in each round one impression becomes available. The total number of rounds is denoted by $T \in \mathbb{N}$. In each round $t \in \{1, \dots, T\}$ the publisher has to decide on a reserve price $p_t \in \mathcal{P}$, where $\mathcal{P}$ is the set of admissible prices. After setting the reserve price, the impression is offered for sale on the RTB-market via a Supply Side Platform (SSP). A second-price auction takes place on the RTB-market and the revenue of the publisher depends on the outcome of this auction. Let X_t and Y_t denote the highest and second highest bid respectively in the auction for impression in round t. Then the revenue (or return) of the publisher in round t is given by $R_t(p_t) = \mathbb{I}\{p_t \leq X_t\} \cdot \max\{Y_t, p_t\}$. Here $\mathbb{I}\{A\} = 1$ if A is true and $\mathbb{I}\{A\} = 0$ otherwise. The expression for $R_t(p_t)$ says that if the reserve price p_t is too high (i.e. if $p_t > X_t$) then the publisher receives zero revenue. Otherwise (i.e. if $p_t \leq X_t$), the revenue is equal to the maximum of the second highest bid and the reserve price. We assume that the bids are drawn from a joint distribution $\mathcal{D}_t$ in round t where $\mathcal{D}_t$ does not depend on p_t. This assumption is common (see e.g., [42]) and it is reasonable when the auction is open to a wide audience of potential bidders and where the pool of bidders can vary from auction to auction. Since the publisher does not interact with the same set of bidders in every round, it is reasonable to assume that the bidders and the publisher are not engaging in strategic behaviour. In this case, it is plausible that the publisher's strategy of choosing reserve prices has no influence on the distribution of bids (see e.g., [10, 75, 76, 122]). We assume that the realized values of X_t and Y_t are not revealed to the publisher (we use the term *unobserved bids* to emphasize that the seller does not observe the value of bids in the auction after each sale). However, as we describe in Section 3.4, the publisher can sometimes infer the value of Y_t.

In summary, we assume auctions proceed according to the following online protocol. For each round $t \in \{1, \dots, T\}$:

1. the publisher selects a reserve price $p_t \in \mathcal{P}$ which is visible to the bidders.

2. the values for X_t and Y_t are drawn, hidden from the publisher, from the joint distribution of the bids $\mathcal{D}_t$ in round t, and this distribution is unknown to the publisher.

3. the publisher observes $\mathbb{I}\{p_t \leq X_t\}$ and receives $R_t(p_t) = \mathbb{I}\{p_t \leq X_t\} \cdot \max\{Y_t, p_t\}$ as revenue.

The objective of the publisher is to determine a sequence of reserve prices $p_1, \ldots, p_T$ in order to maximize the expected cumulative revenue over T rounds. Thus the revenue optimization problem over T rounds or impressions can be expressed as follows:

$$\max_{p_1,\ldots,p_T} \sum_{t=1}^{T} \mathbb{E}_{\mathcal{D}_t} \left\{ \mathbb{I}\{p_t \leq X_t\} \cdot \max\{Y_t, p_t\} \right\} = \max_{p_1,\ldots,p_T} \sum_{t=1}^{T} \mathbb{E}_{\mathcal{D}_t} \left\{ R_t(p_t) \right\}, \tag{3.1}$$

where $\mathbb{E}_{\mathcal{D}_t}\{\cdot\}$ denotes the expectation operator with respect to the joint distribution of the bids $\mathcal{D}_t$. For a generic random variable Z, we denote the expectation and variance with $\mathbb{E}\{Z\}$ and $\mathbb{V}\{Z\}$, respectively, when the distribution is clear from the context.

In addition we make the following assumptions.

Assumption 3.1. *We assume that the bids are bounded and that* $(X_t, Y_t) \in \{[0,1] \times [0,1] \mid X_t \geq Y_t\}$ *and for all* t.

Assumption 3.2. *The set of admissible prices* $\mathcal{P}$ *is finite and* $|\mathcal{P}| = K$.

Assumption 3.3. *Without loss of generality we assume that the prices are ordered such that* $0 \leq p^1 \leq p^2 \leq \cdots \leq p^K \leq 1$ *with* $p^k \in \mathcal{P}$ *for* $k = 1, \ldots, K$.

Assumption 3.1 is common in the literature on multi-armed bandits, dynamic pricing and learning in auctions (see e.g. [15, 17, 42, 91, 157]). Assumption 3.2 states that the set of admissible prices is finite. This assumption is also fairly common in the literature on multi-armed bandits and dynamic pricing [56, 121, 157]. In the setting of reserve price optimization this is also reasonable as the reserve price is rounded to cents in practice (see e.g. [127]). Assumption 3.3 is useful in order to simplify the presentation.

Remark 3.1. *In the literature on online advertising and the RTB-market, the reserve price is sometimes also referred to as the* floor price.

Remark 3.2. *In order to simplify the exposition of our method, we focus on the case where there is a single ad slot. However, in practice, the publisher may want to set a reserve price depending on the characteristics of the user and the ad slot. Our method can also be applied in such a setting by, for example, making segments of user and slot pairs and applying our method for each segment.*

Remark 3.3. *It may happen at some bidders are not interested in the impression at all (i.e., the impression receives less than two bids). If only one bidder submits a bid in the auction, then it is assumed that* $Y_t = 0$. *If zero bidders submit a bid in the auction, then it is assumed that* $X_t = Y_t = 0$.

Remark 3.4. *Reserve prices in second-price auctions can also be hidden from the bidders. For example, in the context of eBay auctions, sellers have the option of*

choosing a hidden reserve price [77]. In online advertising applications, the reserve price may or may not be disclosed to the bidders (see e.g., [169]). If the reserve price is hidden, this will result in the same auction outcome and revenue for the publisher, since we assume that the bids are drawn from a distribution that does not depend on the reserve price. In order to simplify the exposition of our method, and since we use the setting of online advertising as a running example, we follow the RTB specification [94] and assume that the reserve price is visible.

3.4 Proposed Model: BMAB-SPAR

In this section we present a model for learning reserve prices. In Section 3.4.1 we give a standard MAB formulation for pricing problems and explain how our ideas differ. In Section 3.4.2 we present our model BMAB-SPAR (Bayesian Multi-Armed Bandit for Second-Price Auctions with Reserve prices). In the remainder of this section we make the additional assumption that all bids are drawn from a stationary (fixed) joint distribution.

Assumption 3.4. *The pairs of bids (X_t, Y_t) are independently and identically distributed (i.i.d.) draws from an underlying stationary (fixed) joint distribution $\mathcal{D}$ for all t. That is, $\mathcal{D}_t = \mathcal{D}$ for all t.*

3.4.1 Standard MAB formulation and main ideas

In the standard MAB formulation for pricing problems there is a finite set of actions $\mathcal{A} = \{a_1, \ldots, a_K\}$ which represent prices that can be selected. The revenue gained by selecting price a_i is a bounded random variable $V_i = W(a_i) \cdot Z_i$, where $Z_i \sim \mathcal{BN}(\mu(a_i))$ is a Bernoulli variable that represents the outcome (sale/no sale) of the transaction, with $\mu(a_i)$ the probability that the transaction leads to a sale when a_i is selected and $W(a_i)$ a random variable that denotes the revenue at price a_i given that there is a sale (in standard pricing problems we simply have that $W(a_i) = a_i$ because the customer pays the quoted price). The expected revenue at price a_i is given by

$$\mathbb{E}\{V_i\} = \mathbb{E}\{W(a_i)\} \cdot \mu(a_i) \tag{3.2}$$

and the optimal price is $a^* = \text{argmax}_{a \in \mathcal{A}} \mathbb{E}\{W(a)\} \cdot \mu(a)$. BMAB-SPAR builds on this standard MAB formulation. Observe that, given a specific reserve price p, the expected revenue can be written as follows

$$\mathbb{E}_{\mathcal{D}}\{R_t(p)\} = \mathbb{P}\{p \leq X_t\} \cdot \mathbb{E}_{\mathcal{D}}\{\max\{Y_t, p\} |\ p \leq X_t\}. \tag{3.3}$$

Here $\mathbb{E}_{\mathcal{D}}\{\max\{Y_t, p\} |\ p \leq X_t\}$ denotes the expectation of $\max\{Y_t, p\}$ with respect to the distribution $\mathcal{D}$ conditional on the event $\{p \leq X_t\}$. Thus the expected revenue at a reserve price depends on two components. First, the impression needs to be sold.

Second, conditional on the impression being sold, the revenue equals the maximum of the second highest bid and the reserve price. The probability that the impression will be sold, or the success probability, is given by $\mathbb{P}\{p \leq X_t\}$. The expected revenue given success, is equal to $\mathbb{E}_{\mathcal{D}}\{\max\{Y_t, p\} |\ p \leq X_t\}$. Note the similarity in structure of the decomposition of expected revenue in Equation (3.3) and the standard MAB formulation in Equation (3.2): (i) $\mathbb{P}\{p \leq X_t\}$ plays the role of $\mu(a_i)$ and (ii) $\mathbb{E}_{\mathcal{D}}\{\max\{Y_t, p\} |\ p \leq X_t\}$ plays the role of $\mathbb{E}\{W(a_i)\}$.

The solution to the optimization problem in Equation (3.1) is to use the reserve price that maximizes the expectation given in Equation (3.3) in every round t. The main idea in BMAB-SPAR is to learn approximations for each term in Equation (3.3) as the number of rounds progresses and to select the reserve price based on these approximations. A key difference with the standard MAB approach is that BMAB-SPAR explicitly exploits the feedback of second-price auctions in order to update these estimates, whereas standard MAB models assume that the reserve prices are independent options.

3.4.2 BMAB-SPAR formulation

In BMAB-SPAR, there is a set $\mathcal{K} = \{1, \ldots, K\}$ of arms. Each arm $k \in \mathcal{K}$ is associated with a reserve price $p^k \in \mathcal{P}$. The outcome of the sale using reserve price p in round t is given by $S_t(p) = \mathbb{I}\{p \leq X_t\}$ and it is modeled as a random variable with a Bernoulli distribution, that is, $S_t(p) \sim \mathcal{BN}(\mu(p))$. Here $\mu(p) = \mathbb{E}\{S_t(p)\}$ represents the probability that the impression will be sold if a reserve price p is used. We will refer to this as the *probability of success* at a reserve price p. For ease of presentation, we denote the probability of success associated with arm k by $\theta_k = \mu(p^k)$. The value of θ_k is unknown and we take a Bayesian approach in order to update our knowledge about θ_k. More specifically, the parameter θ_k is modeled as a random variable with a Beta distribution as a prior: $\mathcal{B}(a_{k,0}, b_{k,0})$. The prior distribution is subsequently updated based on the feedback that is observed in each round.

Updating posterior distribution

For a given value of v, define $P(v) = \{p^k \in \mathcal{P} | p^k \leq v, k \in \mathcal{K}\}$, and $k^+(v) = \max\{k \in \mathcal{K} | p^k = \max\{P(v)\}\}$. The set $P(v)$ represents the subset of prices in $\mathcal{P}$ that are at most v and $k^+(v)$ is the index of the arm with the highest price in $P(v)$. Let the prior distribution of arm k at the end of round $t-1$ be denoted by $\mathcal{B}(a_{k,t-1}, b_{k,t-1})$. Assume that arm $k^* \in \mathcal{K}$ is played in round t (i.e., $p_t = p^{k^*}$). After arm k^* is played the seller observes $S_t(p^{k^*}) = s_t$ and $R_t(p^{k^*}) = r_t$. We use the following procedure for updating the priors associated with each arm $j \in \mathcal{K}$:

- (Type A update). If $s_t = 1$, then for all $j \in \{1, \ldots, k^* - 1\} \cup \{k^*, \ldots, k^+(r_t)\}$, the posterior of arm j (at the end of round t) becomes $\mathcal{B}(a_{j,t-1} + 1, b_{j,t-1})$.

- (Type B update). If $s_t = 0$, then for all $j \in \{k^*, k^* + 1, \ldots, K\}$, the posterior of arm j (at the end of round t) becomes $\mathcal{B}(a_{j,t-1}, b_{j,t-1} + 1)$.

A Type A update occurs every time that a sale is successful and a Type B update occurs every time that a sale is not successful. In standard bandit algorithms, only the selected arm (arm k^*) would get updated. However, in BMAB-SPAR, the Type A and B updates can also update arms with $j \neq k^*$ in a round. The updating procedure basically adds pseudo-successes or pseudo-failures to arms $j \neq k^*$, depending on whether the sale at price p^{k^*} was successful or not. The motivation for this update scheme comes from two important properties of the problem at hand. The first property stems from the fact that we are getting feedback from *pricing problem*. If a sale was not successful at price p^{k^*}, then we know that the same sale would not have been successful at any price $p > p^{k^*}$ (thus, the Type B updates add pseudo-failures). Similarly, if a sale was successful at price p^{k^*}, then we know that the same sale would also have been successful at any price $p < p^{k^*}$ (thus, the Type A updates add pseudo-successes). We refer to this property as the *pricing problem property*.

The second property that is exploited is related to the *structure of the second-price auction mechanism*. If a sale was successful at price p^{k^*} and the realized revenue is r_t, then we know that the same sale would also have been successful at any price $p < p^{\hat{k}}$ with $\hat{k} = k^+(r_t)$. We refer to this property as the *structured revenue property*. Note that, if the realized revenue is higher than the reserve price ($r_t > p^{k^*}$), then the *structured revenue property* allows us to infer that reserve prices $\bar{p}$ that satisfy $p^{k^*} \leq \bar{p} \leq p^{\hat{k}}$ also would have been successful in round t. Thus, the structured revenue property allows us to potentially update arms with $j > k^*$ and this is reflected by the set $\{k^*, \ldots, k^+(r_t)\}$ in the Type A update. This particular feedback follows from the way that the revenue is determined and is specific to the second-price auction mechanism.

Estimating the expected return for successful sales

We first define some notation. Let $\mathcal{T}_{A,j,t} \subseteq \{1, \ldots, t\}$ denote the indices of the rounds that an update of type A took place for arm j until round t. Let $\mathcal{T}_{S,j,t} = \{l \in \mathcal{T}_{A,j,t} | p_l = p^j\}$ denote the indices of the rounds where arm j was pulled *and* an update of type A took place for arm j until round t. Let $\mathcal{R}_{S,j,t} = \{\max\{Y_l, p^j\} | l \in \mathcal{T}_{S,j,t}\}$ denote the observed revenues for the rounds $l \in \mathcal{T}_{S,j,t}$.

The main idea is to impute a *pseudo-revenue* for arm j every time that a Type A update takes place for arm j. The pseudo-revenue is an approximation of the revenue that could be obtained if the sale in round l at reserve price p^j would be successful. Let $\mathcal{R}_{A,j,t} \subseteq \mathbb{R}$ denote the set of pseudo-revenues for arm j based on information collected during rounds $t \in \mathcal{T}_{A,j,t}$. We now explain how the pseudo-revenues are calculated.

Assume that arm $n_l \in \mathcal{K}$ is played in round $l \in \mathcal{T}_{A,j,t}$ with reserve price p^{n_l} and that the observed revenue equals $R_l(p^{n_l}) = r_l$. The pseudo-revenue $\tilde{R}_{l,j} \in \mathcal{R}_{A,j,t}$ for

arm j in round $l \in \mathcal{T}_{A,j,t}$ is defined as follows:

$$\tilde{R}_{l,j} = \begin{cases} p^j, & \text{if } p^{n_l} = r_l,\ j = n_l \\ \bar{R}_{l,j}(p^n), & \text{if } p^{n_l} = r_l,\ j < n_l \\ R_l, & \text{if } p^{n_l} \neq r_l,\ j \leq k^+(r_l), \end{cases} \tag{3.4}$$

where

$$\bar{R}_{l,j}(v) = \begin{cases} \min\{p^j, v\}, & \text{if } \mathcal{T}_{S,j,l} = \varnothing \\ \sum\limits_{r \in \mathcal{R}_{S,j,l}} \min\{r, v\}/|\mathcal{T}_{S,j,l}|, & \text{otherwise.} \end{cases} \tag{3.5}$$

Given the set $\mathcal{T}_{A,j,t}$, the set $\mathcal{R}_{A,j,t}$ is then defined as

$$\mathcal{R}_{A,j,t} = \cup_{l \in \mathcal{T}_{A,j,t}} \{\tilde{R}_{l,j}\} \tag{3.6}$$

The pseudo-revenue in Equation (3.4) depends on three cases. In case 1 (if $p^{n_l} = r_l,\ j = n_l$), the observed revenue equals the chosen reserve price and the pseudo-revenue for arm n_l simply equals the actual observed revenue for arm n_l. In case 2 (if $p^{n_l} = r_l,\ j < n_l$), we do not know what the exact revenue (call this R^e) would have been *had* arm $j < n_l$ been selected instead of arm n_l. However, from the properties of the second-price auction, we can deduce that $p^j \leq R^e \leq p^{n_l}$ needs to hold for round l. The idea in case 2, is to approximate R^e with $R^e(p^j, p^{n_l}) = \mathbb{E}_{\mathcal{D}}\left\{\min\{Z, p^{n_l}\} |\ Z = \max\{Y_t, p^j\}, p^j \leq X_t\right\}$. Since the distribution $\mathcal{D}$ is unknown, $R^e(p^j, p^{n_l})$ is approximated by observed samples using Equation (3.5). In Equation (3.5), the set $\mathcal{R}_{S,j,l}$ contains observed rewards from the rounds when arm j was selected and the sale was successful (these rounds are in the set $\mathcal{T}_{S,j,l}$). The set $\mathcal{R}_{S,j,l}$ thus contains samples from the distribution of $\max\{Y_t, p^j\} | p^j \leq X_t\}$ and these samples are used to approximate $R^e(p^j, p^{n_l})$. If $\mathcal{T}_{S,j,l}$ is empty, then the reserve price p^j is used instead (since this is a lowerbound for the revenue in case of successful sales). In case 3 (if $p^{n_l} \neq r_l,\ j \leq k^+(r_l)$), the second highest bid is higher than the selected reserve price p^{n_l}, and from the properties of the second-price auction, we can deduce that $R^e = r_l$ needs to hold for round l. That is, the exact revenue R^e *had* arm $j \leq k^+(r_l)$ been selected instead of arm n_l equals $R^e = r_l$.

BMAB-SPAR maintains an estimate of $\mathbb{E}_{\mathcal{D}}\left\{\max\{Y_t, p^j\} |\ p^j \leq X_t\right\}$ based on information collected until round t, which is denoted by $m(j,t)$. The set $\mathcal{R}_{A,j,t}$ is used to calculate $m(j,t)$. Formally, $m(j,t)$ is defined as

$$m(j,t) = \begin{cases} p^j, & \text{if } \mathcal{T}_{A,j,t} = \varnothing \\ \sum\limits_{r \in \mathcal{R}_{A,j,t}} r/|\mathcal{T}_{A,j,t}|, & \text{otherwise} \end{cases} \tag{3.7}$$

If $\mathcal{T}_{A,j,t} = \varnothing$, then no Type A update has occurred until round t and $m(j,t)$ equals the reserve price p^j, since this is a lower bound for the revenue in case of successful sales.

If $\mathcal{T}_{A,j,t} \neq \varnothing$, then $\mathcal{R}_{A,j,t}$ is used to determine the value of $m(j,t)$. Note how the definition of $m(j,t)$ also exploits the *structured revenue property*. More specifically, in case 2 and case 3 in Equation (3.4) we can (approximately) infer what the revenue would have been for an arm j even if j was not pulled in round t. Note that this feedback is specific for the second-price auction.

Example 3.1. *Here we give an example to illustrate the three cases described above. Suppose that* $\mathcal{P} = \{p^1, p^2, p^3, p^4\} = \{0.1, 0.2, 0.4, 0.6\}$. *Let's assume that BMAB-SPAR has been running for 4 rounds such that* $\mathcal{T}_{A,1,t} = \mathcal{T}_{A,2,t} = \mathcal{T}_{A,3,t} = \{1,2,3,4\}$, $\mathcal{T}_{A,4,t} = \mathcal{T}_{S,4,t} = \varnothing$, $\mathcal{T}_{S,2,t} = \{1,2\}$, $\mathcal{T}_{S,3,t} = \{3,4\}$, $\mathcal{R}_{A,2,t} = \mathcal{R}_{A,3,t} = \{0.45, 0.45, 0.5, 0.5\}$, $\mathcal{R}_{S,2,t} = \{0.45, 0.45\}$, $\mathcal{R}_{S,3,t} = \{0.5, 0.5\}$. *Now suppose that in round* $t = 5$, *reserve price* $p^3 = 0.4$ *is selected and that* $R_5 = 0.4$.

- *For the update of reserve price* p^3, *case 1 applies and we have that* $\mathcal{T}_{A,3,t} = \{1,2,3,4\} \cup \{5\}$, $\mathcal{R}_{A,3,t} = \{0.45, 0.45, 0.5, 0.5\} \cup \{0.4\}$, $\mathcal{R}_{S,3,t} = \{0.5, 0.5\} \cup \{0.4\}$, *and* $m(3,5) = (2 \cdot 0.45 + 2 \cdot 0.5 + 0.4)/5$.
- *For the update of reserve price* p^2, *case 2 applies and we have that* $\mathcal{T}_{A,2,t} = \{1,2,3,4\} \cup \{5\}$, $\mathcal{R}_{A,2,t} = \{0.45, 0.45, 0.5, 0.5\} \cup \{x\}$, *and* $m(2,5) = (2 \cdot 0.45 + 2 \cdot 0.5 + x)/5$. *Here* x *is determined by Equation (3.5) (with* $\mathcal{T}_{S,2,t} \neq \varnothing$*) and yields* $x = (2 \cdot \min\{0.45, 0.4\})/2 = 0.4$.
- *For the update of reserve price* p^1, *case 2 applies and we have that* $\mathcal{T}_{A,1,t} = \{1,2,3,4\} \cup \{5\}$, $\mathcal{R}_{A,1,t} = \{0.45, 0.45, 0.5, 0.5\} \cup \{x\}$, *and* $m(1,5) = (2 \cdot 0.45 + 2 \cdot 0.5 + x)/5$. *Here* x *is determined by Equation (3.5) (with* $\mathcal{T}_{S,1,t} = \varnothing$*) and yields* $x = \min\{0.1, 0.4\} = 0.1$.
- *For the reserve price* p^4, *no Type A update occurs (because* $p^4 > p^3$*) and therefore* $m(4,4) = m(4,5)$.

If instead $R_5 = 0.5$, *then case 3 applies to* p^1, p^2 *and* p^3, *and we would have* $\mathcal{T}_{A,1,t} = \mathcal{T}_{A,2,t} = \mathcal{T}_{A,3,t} = \{1,2,3,4\} \cup \{5\}$, $\mathcal{R}_{A,1,t} = \mathcal{R}_{A,2,t} = \mathcal{R}_{A,3,t} = \{0.45, 0.45, 0.5, 0.5\} \cup \{0.5\}$, $m(1,5) = m(2,5) = m(3,5) = (2 \cdot 0.45 + 2 \cdot 0.5 + 0.5)/5$. □

Arm selection procedure

In order to decide which arm or reserve price to select, we construct an index for each arm. In general, UCB-like indexes (see e.g. [17]) depend on two components: (i) a component that measures the mean of an uncertain quantity and (ii) a component that adds an exploration bonus. In BMAB-SPAR, the index for arm j is defined as $I(j,t) = \bar{x}_{j,t} + v_{j,t}$. Here $\bar{x}_{j,t}$ denotes the mean of the posterior distribution for arm j after t rounds, and $v_{j,t}$ denotes the posterior variance of arm j after t rounds. As the posterior distribution of arm j after t rounds follows a Beta distribution with

parameters $\mathcal{B}(a_{j,t}, b_{j,t})$, the index can be written as follows

$$I(j,t) = \bar{x}_{j,t} + v_{j,t} = \frac{a_{j,t}}{a_{j,t}+b_{j,t}} + \frac{a_{j,t}\cdot b_{j,t}}{(a_{j,t}+b_{j,t})^2(a_{j,t}+b_{j,t}+1)}. \tag{3.8}$$

In round $t+1$, BMAB-SPAR selects arm j^* such that $j^* = \text{argmax}_{j\in\mathcal{K}} I(j,t)\cdot m(j,t)$. The pseudo-code for BMAB-SPAR is presented in Algorithm 3.1.

Note that the index as defined above is strictly speaking not an UCB-index, since it is not guaranteed to be an upper bound on the success probability that holds with high confidence. However, the index is easy to compute, and the overall structure (consisting of mean and exploration bonus) makes intuitive sense. Furthermore, this index performs well in our experiments. However, there might be better indexes that can be constructed.

3.5 Model for non-stationary environments: BMAB-SPAR-NS

In this section we present an extension of BMAB-SPAR that can be used for learning reserve prices in non-stationary environments. Accordingly, in the remainder of this section, we no longer assume that Assumption 3.4 holds. We refer to our model as BMAB-SPAR-NS (Bayesian Multi-Armed Bandit for Second-Price Auctions with Reserve prices for Non-Stationary environments).

For an arbitrary set $\mathcal{S} = \{p_1, \ldots, p_Q\}$ containing Q real numbers, define $\bar{\mathbb{V}}(\mathcal{S}) = \sum_{i=1}^{Q} \frac{1}{Q-1}(p_i - \bar{p})^2$ and $\bar{p}(\mathcal{S}) = \sum_{i=1}^{Q} \frac{1}{Q} p_i$. Furthermore, define the operator $\mathcal{M}^x(\cdot)$ that takes as input a set $\mathcal{S}$ and outputs a set $\mathcal{S}^x \subseteq \mathcal{S}$ that contains the last x elements that were added to the set $\mathcal{S}$. If $|\mathcal{S}| \leq x$, then define $\mathcal{M}^x(\mathcal{S}) = \mathcal{S}$.

3.5.1 Formulation of non-stationary environment

In this section we consider a piece-wise stationary environment (similar to e.g. [39]) with M segments in order to characterize a non-stationary environment. Each segment consists of a number of rounds and the distribution of the bids is constant in each segment. However, the distribution can change from one segment to another. For a sales horizon of length T, let $\mathcal{T} = \{1, \ldots, T\}$. Notice that there are $M-1$ change points in a piece-wise stationary environment with M segments. We will denote the change points by $\nu_1, \ldots, \nu_{M-1}$ with $\nu_i \in \{1, \ldots, T-1\}$ for $i = 1, \ldots, M-1$ where $\nu_i < \nu_j$ if $i < j$. Furthermore, we define $\nu_0 = 0$ and $\nu_M = T$. For each segment i, the set $\mathcal{T}_i$ is of the form $\mathcal{T}_i = \{\nu_{i-1}+1, \ldots, \nu_i\}$. Given a segment $i \in \{1, \ldots, M\}$, the joint distribution of the bids satisfies (i) $\mathcal{D}_t = \mathcal{D}_k$ for all $t, k \in \mathcal{T}_i$, and (ii) $\mathcal{D}_t \neq \mathcal{D}_k$ if $t \in \mathcal{T}_i$ and $k \in \mathcal{T}_j$ with $j \in \{i-1, i+1\}$.

Algorithm 3.1 BMAB-SPAR

Require: number of arms K, set of admissible prices $\mathcal{P}$, set of arms $\mathcal{K}$, parameters of prior distribution $\{(a_{j,0}, b_{j,0})\}_{j\in\mathcal{K}}$.

Set $t = 0$.
for $j \in \mathcal{K}$ **do**
  Set $\mathcal{T}_{A,j} = \varnothing$. Set $\mathcal{T}_{S,j} = \varnothing$. Set $\mathcal{R}_{A,j} = \varnothing$. Set $\mathcal{R}_{S,j} = \varnothing$.
end for
while $t \leq T$ **do**
  Set $t = t + 1$.
  if $t \leq K$ **then**
    Set $z = t$.
  else
    Set $z = \text{argmax}_{j\in\mathcal{K}} I(j,t) \cdot m(j,t)$.
  end if
  Use $p_t = p^z$ as reserve price in round t.
  Observe $S_t = \mathbb{I}\{p_t \leq X_t\}$ and $R_t = \mathbb{I}\{p_t \leq X_t\} \cdot \max\{Y_t, p_t\}$.
  for $j \in \mathcal{K}$ **do**
    if $S_t = 1$ **and** $j \leq k^+(R_t)$ **then**
      Update posterior to $\mathcal{B}(a_{j,t-1} + 1, b_{j,t-1})$.
      Set $\mathcal{T}_{A,j} = \mathcal{T}_{A,j} \cup \{t\}$. Set $\mathcal{T}_{A,j,t} = \mathcal{T}_{A,j}$.
      if $p^z = R_t$ **and** $j = z$ **then**
        Set $\mathcal{R}_{S,j} = \mathcal{R}_{S,j} \cup \{p^j\}$. Set $\mathcal{T}_{S,j} = \mathcal{T}_{S,j} \cup \{t\}$. Set $\mathcal{R}_{A,j} = \mathcal{R}_{A,j} \cup \{p^j\}$.
      else if $p^z = R_t$ **and** $j < z$ **then**
        Set $\mathcal{R}_{S,j,t} = \mathcal{R}_{S,j}$. Set $\mathcal{T}_{S,j,t} = \mathcal{T}_{S,j}$. Set $\mathcal{R}_{A,j} = \mathcal{R}_{A,j} \cup \{\bar{R}_{t,j}(p^z)\}$ according to Equation (3.5).
      else if $p^z \neq R_t$ **and** $j \leq k^+(R_t)$ **then**
        Set $\mathcal{R}_{A,j} = \mathcal{R}_{A,j} \cup \{R_t\}$.
        if $j = z$ **then**
          Set $\mathcal{R}_{S,j} = \mathcal{R}_{S,j} \cup \{R_t\}$. Set $\mathcal{T}_{S,j} = \mathcal{T}_{S,j} \cup \{t\}$.
        end if
      end if
      Set $\mathcal{R}_{A,j,t} = \mathcal{R}_{A,j}$.
    else if $S_t = 0$ **and** $j \geq z$ **then**
      Update posterior to $\mathcal{B}(a_{j,t-1}, b_{j,t-1} + 1)$.
    end if
  end for
  for $j \in \mathcal{K}$ **do**
    Set $m(j,t)$ according to Equations (3.7) - (3.6). Set $I(j,t)$ according to Equation (3.8).
  end for
end while

3.5.2 Description of algorithm

Recall that BMAB-SPAR maintains approximations of the two terms in Equation (3.3). In BMAB-SPAR-NS, these approximations need to be able to adjust to a changing environment. The structure of BMAB-SPAR-NS is similar to that of BMAB-SPAR, except with some modifications for the non-stationary setting. Intuitively, the purpose of these changes is to make the algorithm adjust to the changing environment by (i) "forgetting" information obtained in previous rounds and by (ii) exploring the

less used actions in order to learn if the optimal action has changed. The pseudo-code for BMAB-SPAR-NS is presented in Algorithm 3.2.

Algorithm 3.2 BMAB-SPAR-NS

Require: number of arms K, set of admissible prices $\mathcal{P}$, set of arms $\mathcal{K}$, parameters of prior distribution $\{(a_{j,0}, b_{j,0})\}_{j\in\mathcal{K}}$, parameter q, parameter τ, parameter κ, parameter w, $0 < p^{ex} < 1$.

Set $t = 0$. Set $\mathcal{A} = \varnothing$.
for $j \in \mathcal{K}$ **do**
  Set $\mathcal{T}_{A,j} = \varnothing$. Set $\mathcal{T}_{S,j} = \varnothing$. Set $\mathcal{R}_{A,j} = \varnothing$. Set $\mathcal{R}_{S,j} = \varnothing$. Set $n_j = 0$.
end for
while $t \leq T$ **do**
  Set $t = t + 1$.
  if $t \leq K$ **then**
    Set $z = t$.
  else
    Set $z = \text{argmax}_{j\in\mathcal{K}}\, I(j,t) \cdot m(j,t)$. Set $\mathcal{A}_t = \mathcal{M}^q(\mathcal{A})$. Draw Z from $\mathcal{BN}(p^{ex})$.
    if $\bar{\mathbb{V}}(\mathcal{A}_t) < \tau$ **and** $Z = 1$ **then**
      Select a $z \in \mathcal{K}$ where z has probability $\frac{n_z^{-1}}{\sum_{i=1}^{K} n_z^{-1}}$ of being selected.
    end if
  end if
  Use $p_t = p^z$ as reserve price in round t. Set $\mathcal{A} = \mathcal{A} \cup \{p_t\}$.
  Observe $S_t = \mathbb{I}\{p_t \leq X_t\}$ and observe $R_t = \mathbb{I}\{p_t \leq X_t\} \cdot \max\{Y_t, p_t\}$.
  for $j \in \mathcal{K}$ **do**
    Set $a_{j,t-1} = \kappa \cdot a_{j,t-1}$. Set $b_{j,t-1} = \kappa \cdot b_{j,t-1}$.
    if $S_t = 1$ **and** $j \leq k^+(R_t)$ **then**
      Update posterior $\mathcal{B}(a_{j,t-1} + 1, b_{j,t-1})$. Set $\mathcal{T}_{A,j} = \mathcal{T}_{A,j} \cup \{t\}$. Set $\mathcal{T}_{A,j,t} = \mathcal{M}^w(\mathcal{T}_{A,j})$.
      Set $n_j = n_j + 1$.
      if $p^z = R_t$ **and** $j = z$ **then**
        Set $\mathcal{R}_{S,j} = \mathcal{R}_{S,j} \cup \{p^j\}$. Set $\mathcal{T}_{S,j} = \mathcal{T}_{S,j} \cup \{t\}$. Set $\mathcal{R}_{A,j} = \mathcal{R}_{A,j} \cup \{p^j\}$.
      else if $p^z = R_t$ **and** $j < z$ **then**
        Set $\mathcal{R}_{S,j,t} = \mathcal{M}^w(\mathcal{R}_{S,j})$. Set $\mathcal{T}_{S,j,t} = \mathcal{M}^w(\mathcal{T}_{S,j})$. Set $\mathcal{R}_{A,j} = \mathcal{R}_{A,j} \cup \{\bar{R}_{t,j}(p^z)\}$ using Eqn (3.5).
      else if $p^z \neq R_t$ **and** $j \leq k^+(R_t)$ **then**
        Set $\mathcal{R}_{A,j} = \mathcal{R}_{A,j} \cup \{R_t\}$.
        if $j = z$ **then**
          Set $\mathcal{R}_{S,j} = \mathcal{R}_{S,j} \cup \{R_t\}$. Set $\mathcal{T}_{S,j} = \mathcal{T}_{S,j} \cup \{t\}$.
        end if
      end if
      Set $\mathcal{R}_{A,j,t} = \mathcal{M}^w(\mathcal{R}_{A,j})$.
    else if $S_t = 0$ **and** $j \geq z$ **then**
      Update posterior $\mathcal{B}(a_{j,t-1}, b_{j,t-1} + 1)$. Set $n_j = n_j + 1$.
    end if
  end for
  for $j \in \mathcal{K}$ **do**
    Set $m(j,t)$ according to Eqn (3.7) - (3.6). Set $I(j,t)$ according to Eqn (3.8).
  end for
end while

BMAB-SPAR-NS employs three specific strategies in order to adjust to changing environments. First, it uses a discounting procedure to adjust the parameters of the posterior distribution of each arm. Second, it only uses information from the last w updates to the sets $\mathcal{R}_{A,j,t}$ and $\mathcal{R}_{S,j,t}$ in order to determine $m(j,t)$ using Equations (3.6) - (3.7). Third, it monitors the variance of the last q reserve prices that have been used and conducts forced exploration if this variance is below some threshold. We refer to this as *price-variance weighted exploration*.

The first strategy allows the estimate of the first term in Equation (3.3) to adjust to changing environments. The second strategy allows the estimate of the second term in Equation (3.3) to adjust to changing environments. Finally, the third strategy allows both estimates to adjust to changing environments. We explain these strategies in more detail below.

Discounting of posterior distribution

For each arm j, the parameters of the posterior distribution at the end of round $t-1$ are discounted by a factor $0 < \kappa < 1$ so that the posterior before round t equals $\mathcal{B}(a_{j,t}, b_{j,t}) = \mathcal{B}(\kappa \cdot a_{j,t-1}, \kappa \cdot b_{j,t-1})$. Notice that the discounting procedure enables the algorithm to "forget" information obtained in previous rounds. If there is no discounting ($\kappa = 1$), the posterior distribution becomes more concentrated around the posterior mean as the number of rounds increases. By discounting ($0 < \kappa < 1$), this concentration is partially undone (that is, information is "forgotton") after each round.

The effect of this discounting procedure is to increase the posterior variance for each arm, but at the same time, to maintain the original ranking (without discounting) based on the posterior mean (because discounting does not change the mean). Furthermore, this effect is stronger for arms that do not get updated very often. As the posterior variance determines the exploration bonus of the index $I(j,t)$ in Equation (3.8), the discounting procedure essentially implements a mechanism that promotes additional exploration as the number of rounds increases.

Local estimation of expected returns for successful sales

Due to the piece-wise stationary environment, the optimal reserve price can vary across segments. Furthermore, the expected return of the optimal reserve price may change, even if the optimal reserve price does not differ across segments. In BMAB-SPAR, $m(j,t)$ is used to approximate $\mathbb{E}_{\mathcal{D}_t}\left\{\max\{Y_t, p^j\} |\ p^j \leq X_t\right\}$. Note, that $m(j,t)$ uses all of the information gathered up until round t. However, since $\mathcal{D}_t$ can vary across segments, not all of the information gathered up until round t is relevant: only information from the correct segment is relevant for the computation of the approximation of $\mathbb{E}_{\mathcal{D}_t}\left\{\max\{Y_t, p^j\} |\ p^j \leq X_t\right\}$.

BMAB-SPAR-NS uses only the last w elements that were added to the sets $\mathcal{T}_{S,j,t}$, $\mathcal{T}_{A,j,t}$, $\mathcal{R}_{S,j,t}$ and $\mathcal{R}_{A,j,t}$. More formally, BMAB-SPAR-NS applies the operator $\mathcal{M}^x(\cdot)$

with $x = w$ to the sets $\mathcal{T}_{S,j,t}$, $\mathcal{T}_{A,j,t}$, $\mathcal{R}_{S,j,t}$ and $\mathcal{R}_{A,j,t}$ in order to get $\mathcal{M}^x(\mathcal{T}_{S,j,t})$, $\mathcal{M}^x(\mathcal{T}_{A,j,t})$, $\mathcal{M}^x(\mathcal{R}_{S,j,t})$ and $\mathcal{M}^x(\mathcal{R}_{A,j,t})$. Note that this *local estimation* procedure enables the algorithm to "forget" information obtained in previous rounds. In particular, information from many rounds ago (more than w rounds) is not used, because it is likely that the joint distribution of bids has changed in the meanwhile. Thus, this procedure ensures that only recent and relevant information from the correct segment is used for the computation of the approximation of $\mathbb{E}_{\mathcal{D}_t}\left\{\max\{Y_t, p^j\} |\ p^j \leq X_t\right\}$.

Price-variance weighted exploration

The main idea behind price-variance weighted exploration is to keep track of the variance of last q reserve prices that have been used. If the variance is below some threshold, then with some small probability, we choose a reserve price p^j at random with a probability that is inversely proportional to the number of times arm j has been updated. More formally, let n_j denote the number of times that arm j has been updated. If the variance falls below a threshold τ, then with probability $0 < p^{ex} < 1$ we select a reserve price at random, where p^j has probability $\frac{n_j^{-1}}{\sum_{i=1}^{K} n_j^{-1}}$ of being chosen for $j = 1, \ldots, K$.

One would expect that, after a while, the algorithm would spend most of its time exploiting the best reserve price for a particular segment. The price-variance weighted exploration essentially enforces some additional exploration when the algorithm has entered such an "exploitation phase" (within a segment) in order to check if other reserve prices have possibly become more profitable (between segments). This is especially useful in scenarios where the optimal reserve price switches across segments but where the expected revenue for some reserve prices does not change across segments. The additional exploration induced by the price-variance weighted exploration procedure enables BMAB-SPAR-NS to detect changes in the optimal reserve price in such settings, and this leads to information that can be used to update the estimates of both terms in Equation (3.3).

Recommended parameter settings

For some of the parameters their values can be set to intuitively reasonable values. The parameter κ can be set to 0.99, which is a commonly used value for discount rates. The parameters w and q are harder to choose. These choices depend on the length of the segments. Letting L denote the length of the segments, w should be chosen as a fraction of L and q should be chosen as a fraction of w. In online advertising applications (see e.g. [39]) a change occurs roughly every 20000-50000 (i.e., L ranges roughly between 20000-50000). Therefore, assuming a rather conservative and low value of L, $w = 1000$ and $q = 500$ could be considered as reasonable values. To set the value for τ we follow the idea used in the controlled variance pricing (CVP) policy of [63] and we use $\tau = 0.5 \cdot L^{0.500001-1}$ with $L = 25000$ with results in roughly

$\tau = 0.003$. Intuitively, p^{ex} should be set to a relatively low value since the probability of the next round being in a new segment is relatively low. We set $p^{ex} = 0.015$. In the experimental section, we perform a detailed sensitivity analysis with respect to these choices.

3.6 Experimental analysis in stationary environments

In this section we perform experiments for the stationary environments. Section 3.6.1 discusses the data that is used. The benchmark algorithms are discussed in Section 3.6.2. Settings and performance metrics are discussed in Section 3.6.3. Results of experiments are presented in Section 3.6.4.

3.6.1 Data for experiments

In order to evaluate our method we use an eBay dataset that consists of collector sport cards that was used in [123]. These cards were sold using a second price auction with reserve and the full data set can be found at the following website: `http://cims.nyu.edu/~munoz/data`. The dataset contains information about the top two bids and a number of extra features. These extra features include information about the seller such as positive feedback percent, seller rating and seller country; as well information about the card such as whether the player is in the sport's hall of fame. For the purposes of the experiments in this Chapter, these extra features are not needed and will not be used. We refer the reader to [123] for detailed information about this dataset.

The dataset contains 70213 rows and row j in the dataset is a pair (b_1^j, b_2^j) where b_1^j denotes the highest bid and b_2^j denotes the second highest bid. We take all of the 70213 pairs $B = \cup_{j=1}^{70213}\{(b_1^j, b_2^j)\}$ in the dataset to construct a joint distribution for the highest bid and second highest bid. The main idea is to create a family of joint distributions by first clustering the bids and then defining a probability distribution over the obtained clusters.

The clustering is carried out as follows. First, we determine the 95-th percentile of $B_1 = \cup_{j=1}^{70213}\{b_1^j\}$. Denote the 95-th percentile by TP. Second, we remove outliers by removing pairs (b_1^j, b_2^j) for which the value of the top bid b_1^j exceeds the 95-th percentile TP, that is, we remove pairs j for which $b_1^j > TP$. Next, we calculate the relative gap $z^j = (b_1^j - b_2^j)/b_1^j$. Afterwards, we scale the remaining values of (b_1^j, b_2^j) by the maximum value of b_1^j the dataset so that all the bids are in the range $[0, 1]$. We subsequently cluster the remaining bids using the features b_1^j, b_2^j, z^j with the k-means clustering algorithm (see e.g. [31]) with $M = 10$ clusters. By removing extreme values for the top bid (values that exceed TP), we avoid that the normalized bids get "squished" down to low values in $[0, 1]$. Figure 3.1 displays the resulting clustering of

bids.

Given the clustering, we can define a joint distribution for the highest bid and second highest bid. Let $P_m \in \mathbb{R}^{10}$ be a probability distribution over the clusters, where element i is given by $P_m(i)$ and denotes the probability that cluster i is selected. Given the probability distribution P_m we use the following procedure to sample the values for the highest bid and second highest bid at time $t \in \{1, \ldots, T\}$:

1. Sample a cluster $i \in \{1, \ldots, 10\}$ using the probability distribution P_m. Let i^* denote the sampled cluster.

2. Sample a pair (b_1^j, b_2^j) uniformly and at random from cluster i^*. The value for the highest bid at time t is given by b_1^j and the value for the second highest bid at time t is given by b_2^j.

Notice that for a fixed P_m this procedure results in a stationary distribution for the bids and by varying P_m can generate distributions with different properties.

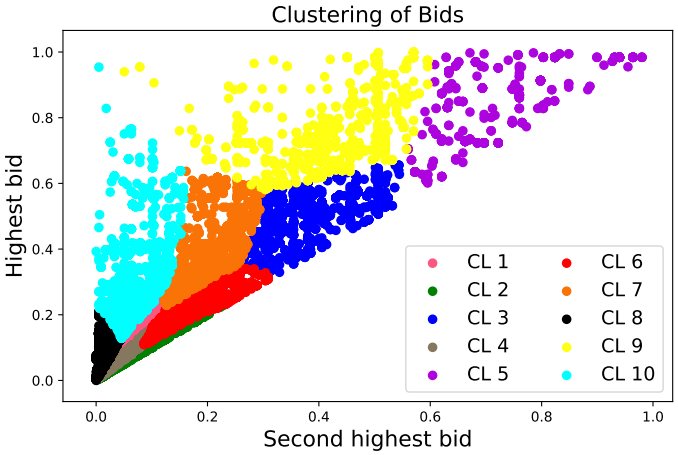


Figure 3.1: Clustering of bids in subset of eBay dataset.

3.6.2 Benchmark algorithms

As there is no algorithm available that is specifically designed for our reserve price optimization problem, one approach to solve this problem is to formulate it as a stochastic multi-armed bandit problem. For this reason, the class of multi-armed bandit algorithms form a natural benchmark for BMAB-SPAR. We compare the performance of BMAB-SPAR with the following benchmark bandit algorithms: UCB1, UCB2, UCB-V, MOSSA, OCUCB, KL-UCB and TS, which are described in detail below.

The UCB1 algorithm is arguably the most popular version of a multi-armed bandit algorithm and is developed in [17]. UCB-V was developed later in [14] and aims to improve the performance of UCB1 by taking the empirical variance of the different arms into account. MOSS [13] is a modified version of UCB1 that has a better worst-case regret bound compared to UCB1. In [62] an improved version of MOSS called MOSS-anytime (MOSSA) is developed that does not require the horizon T as input is developed and experiments showed that it outperformed MOSS. For this reason we include MOSSA in our evaluation. We also include an algorithm called Optimally Confident UCB (OCUCB). This algorithm is based on UCB1, but uses a carefully chosen confidence parameter in order to correctly balance the risk of failing confidence intervals against the cost of excessive optimism [111]. The KL-UCB algorithm [83] uses the Kullback–Leibler divergence in order to determine upper confidence bounds and guide arm selection and showed that KL-UCB outperformed UCB-based algorithms such as UCB1, UCB-V and MOSS. Thompson Sampling (TS) is a randomized algorithm based on Bayesian ideas and has become popular after several studies demonstrated that it has empirical performance that is competitive with alternative state-of-the-art methods. A theoretical analysis of TS was provided in [6] and we use the TS algorithm for bounded rewards as described in [6] as a benchmark.

The benchmarks above are compared with an oracle policy called the *best fixed price in hindsight* (BFPH). This policy looks at all of the draws of the bids, and then determines which reserve price (in hindsight) would maximize the cumulative revenue, and then uses that reserve price in every round. As such, it serves as an upper bound for the maximal performance achievable by the other algorithms. Finally, we also consider the benchmark that always set the reserve price equal to zero (denoted by RPZB), and therefore always receives the second highest bid as revenue.

3.6.3 Settings and performance metrics

We set the prior of each arm j in BMAB-SPAR to $(a_{j,0} = 1.0, b_{j,0} = 1.0)$. The parameters of the benchmark algorithms are set according to the recommendations in the respective papers. UCB1 is tuned according to Theorem 1 in [17]. UCB2 and MOSSA are tuned according to Theorem 3 in [62]. UCB-V is tuned according to Theorem 4 in [14]. OCUCB is tuned according to Section H in [111]. KL-UCB is tuned according to Remark 5 in [83].

We consider a horizon of $T = 40000$. All of the aforementioned algorithms are run with $K \in \{30, 60\}$ arms which are equally spaced in the interval $[0, 1]$.

In order to measure the performance of the methods, we consider two performance metrics. Our main performance metric is the revenue rate, which is defined as $RR(L) = \sum_{t=1}^{L} \hat{R}_t / \sum_{t=1}^{L} X_t$, where $\hat{R}_t$ is the observed return in period t. This measures the rate at which the highest bid is extracted. This metric is useful, because it shows how much revenue is extracted by an algorithm as a fraction of the maximal revenue that is achievable. This metric enables us to rank different algorithms and in-

terpret performance differences in terms of revenue impact. The second performance metric is the average cumulative return, which is defined as $ACR(L) = \sum_{t=1}^{L} \hat{R}_t / L$.

We construct four synthetic datasets using the procedure of Section 3.6.1. In the experiments we report results for the choices in Table 3.1 (additional experimental results can be found in the Appendix (Section 3.C)). In Table 3.1, η_i is an i.i.d. draw from an uniform distribution on $[0.25, 0.75]$ for $i = 1, \ldots, 10$ and Z_i is a normalizing constant. In other words, in order to construct the distribution P_j, we sample the values of η_i for $i = 1, \ldots, 10$ and then we normalize by dividing by Z_i to get a valid probability distribution. Dataset eBay-1 models a case where the gap between the top two bids is relatively small (the clusters correspond to points close to the diagonal in Figure 3.1). Dataset eBay-2 models a case where the gap between the top two bids is relatively large (the clusters correspond to points far off the diagonal in Figure 3.1). In dataset eBay-3 all clusters are equally likely to be sampled in expectation. In dataset eBay-4 all clusters have a positive probability of each selected, but clusters far off the diagonal in Figure 3.1 are more likely. Note that the publisher may not have much information about the joint distribution of the bids. These four cases are interesting since they allow us to investigate how performance varies with different distributions.

Table 3.1: Description of eBay datasets

dataset	distribution
eBay-1	$P_1 = (0, \eta_2, \eta_3, 0, 0, \eta_6, 0, 0, 0, 0)/Z_1$
eBay-2	$P_2 = (0, 0, 0, 0, 0, 0, \eta_7, \eta_8, \eta_9, \eta_{10})/Z_2$
eBay-3	$P_3 = (\eta_1, \eta_2, \eta_3, \eta_4, \eta_5, \eta_6, \eta_7, \eta_8, \eta_9, \eta_{10})/Z_3$
eBay-4	$P_4 = (\eta_1, \eta_2, \eta_3, \eta_4, \eta_5, \eta_6, 4\eta_7, 4\eta_8, 4\eta_9, 4\eta_{10})/Z_4$

3.6.4 Results

Figures 3.2-3.5 show the revenue rate for $K = 30$ and $K = 60$ averaged over 200 independent runs. Results for the average cumulative return can be found in the Appendix (Section 3.C).

Overall, BMAB-SPAR shows a good performance across the different datasets. For long horizons, BMAB-SPAR is as good as the best performing benchmark bandit algorithms, as the revenue rate differs by at most 0.5%. However, there are also instances where BMAB-SPAR outperforms the best bandit algorithms (MOSSA and TS) by about 2% - 4% even after 40000 rounds. In all of the cases considered, BMAB-SPAR outperforms UCB2, UCB-V and UCB1. In general, BMAB-SPAR outperforms the benchmark bandit algorithms for the shorter horizons. Looking at the revenue rate, we see that BMAB-SPAR extracts about 3%-5% more revenue compared to the best benchmark algorithm (MOSSA and TS) up until round 5000. In some cases the performance gap can be as large as 2% - 4% even after 10000 rounds. Furthermore,

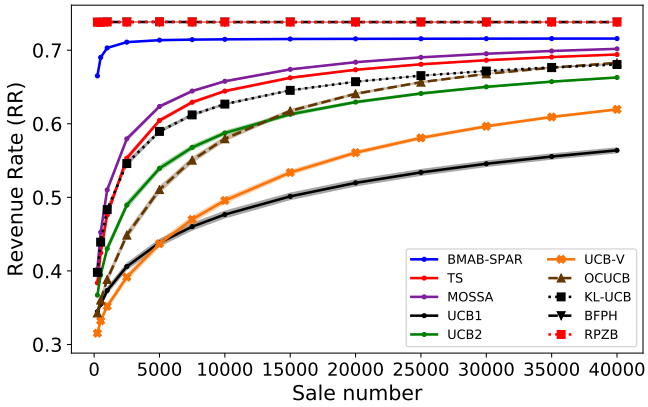
this gap is even larger (between 5% and 8% after 5000 rounds), if we compare BMAB-SPAR with UCB2, UCB-V and UCB1.

The results also indicate that the performance of BMAB-SPAR is very close to the oracle policy (BFPH) that uses the optimal reserve price in every period. In particular, after about 5000 rounds, BMAB-SPAR earns about 94% - 98% of the revenue of BFPH. Furthermore, as the number of rounds increases, the performance of BMAB-SPAR stays close to that of the oracle policy. In contrast, the best benchmark (MOSSA) only earns about 86% - 95% of the revenue of BFPH after 5000 rounds.

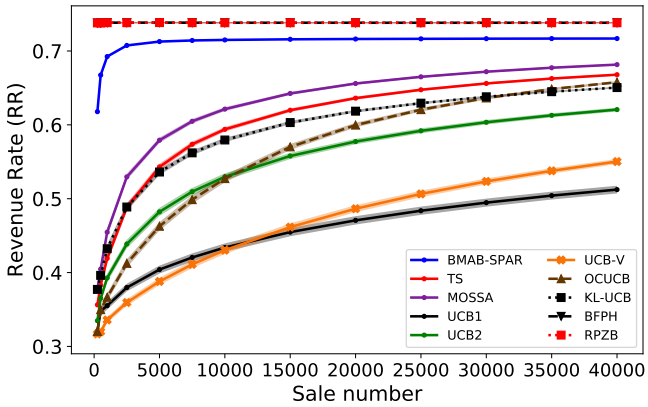
The results show that when the gap between the top two bids is small, then a reserve price of zero tends to perform well, as RPZB and BFPH are very close. The results also show that BMAB-SPAR is very useful when the distribution of the bids is unknown: when gaps are small BMAB-SPAR outperforms the benchmarks and earns about 94% - 98% of RPZB, however, when gaps are large, BMAB-SPAR outperforms RPZB. Therefore, BMAB-SPAR adapts well to the underlying distribution of the bids.

The results indicate that the performance of BMAB-SPAR is robust with respect to the number of arms K. When K increases from $K = 30$ to $K = 60$, the performance gap (measured by the revenue rate) between BMAB-SPAR and MOSSA (after 5000 rounds) is in some cases up to about 2%-3% higher and the performance gap after 10000 rounds about 1%-2% higher. Intuitively, this pattern is in line with expectations, since BMAB-SPAR is able to update information about arm j based on pulls from arms $k \neq j$. As K increases, the learning problem becomes harder, and one would conjecture that this "cross-learning" feature has more added value. The results indeed confirm this conjecture.

The results are promising and indicate that, explicitly taking the rules of second-price auctions into account in order to set reserve prices, has added value for the seller. The results indicate that BMAB-SPAR has the most added value compared to the benchmark algorithms when the seller knows that the selling horizon will be relatively short (without necessarily knowing how long the horizon will be). BMAB-SPAR is especially useful if there is a high number of potential reserve prices and when the gap between the top two bids is relatively large.

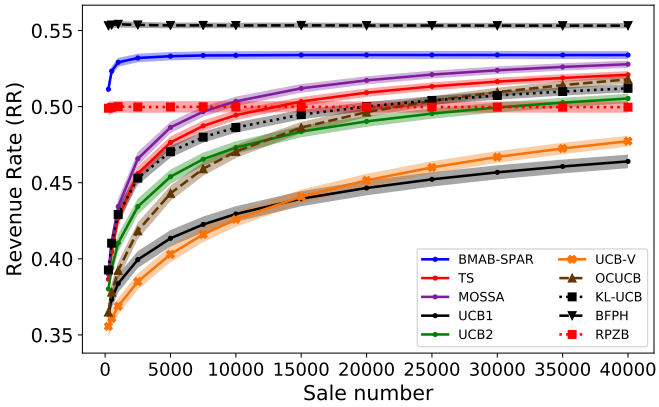


(a) $K = 30$.

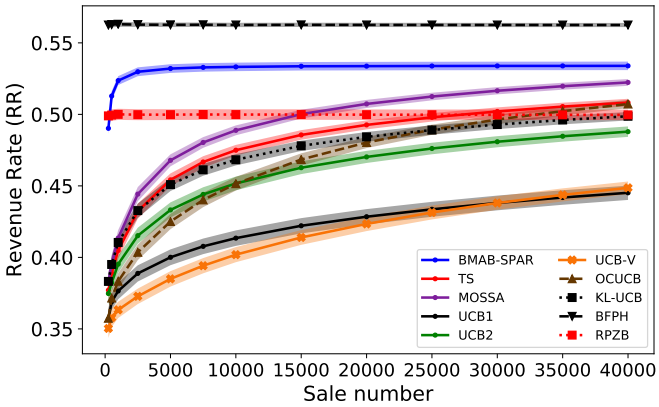


(b) $K = 60$.

Figure 3.2: Performance of the algorithms for dataset eBay-1, averaged over 200 runs. Lines indicate the mean and shaded region indicates 95% confidence interval.

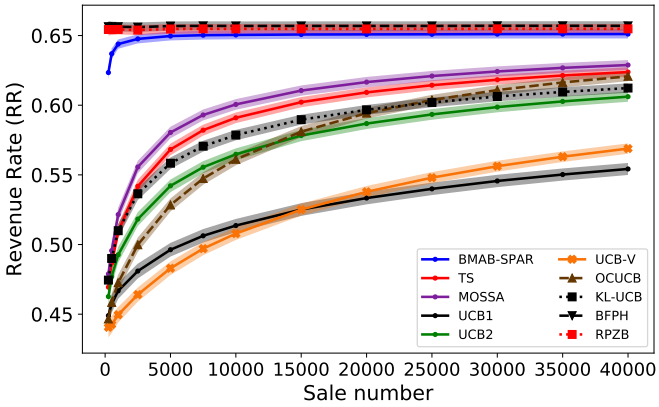


(a) $K = 30$.

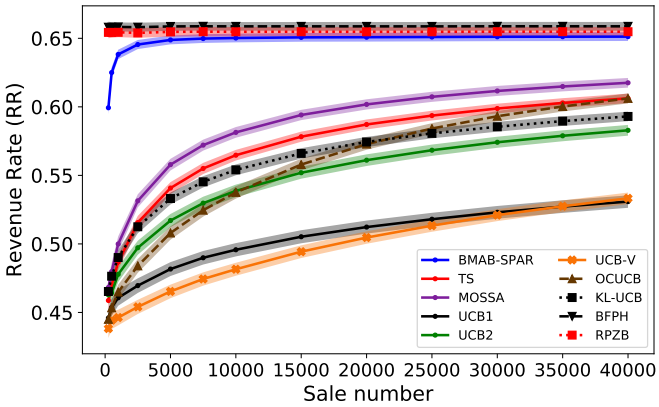


(b) $K = 60$.

Figure 3.3: Performance of the algorithms for dataset eBay-2, averaged over 200 runs. Lines indicate the mean and shaded region indicates 95% confidence interval.

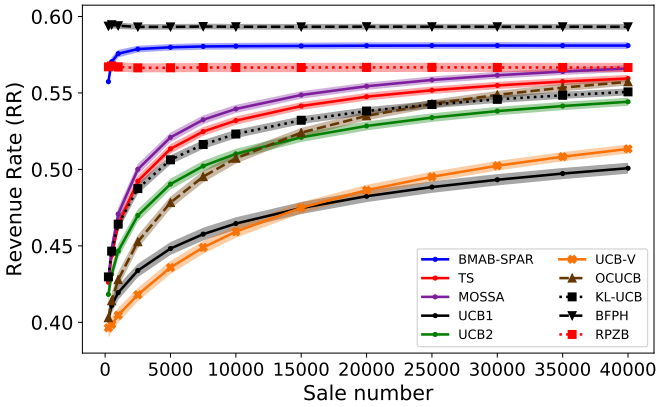


(a) $K = 30$.

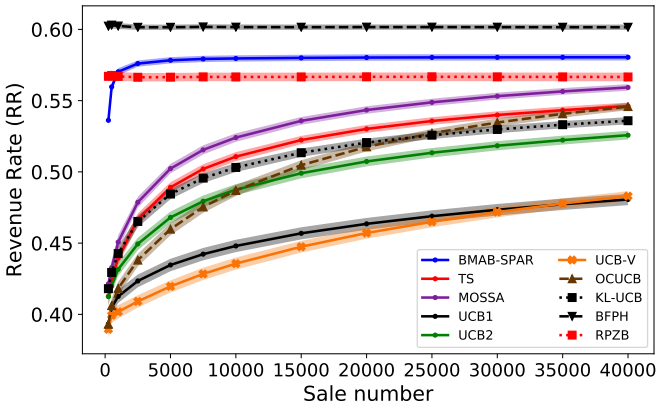


(b) $K = 60$.

Figure 3.4: Performance of the algorithms for dataset eBay-3, averaged over 200 runs. Lines indicate the mean and shaded region indicates 95% confidence interval.



(a) $K = 30$.



(b) $K = 60$.

Figure 3.5: Performance of the algorithms for dataset eBay-4, averaged over 200 runs. Lines indicate the mean and shaded region indicates 95% confidence interval.

3.7 Experimental analysis in non-stationary environments

In this section we perform experiments for non-stationary environments. Section 3.7.1 describes the data that is used. The benchmark algorithms are discussed in Section 3.7.2. Parameter settings and performance metrics are discussed in Section 3.7.3. Results of experiments are presented in Section 3.7.4.

3.7.1 Dataset Description

In order to evaluate our method we use real-life data from ad auction markets. We use header bidding (HB) data from a SME publisher that owns online gaming websites. Header bidding is an alternative way to sell impressions, where the publisher connects to multiple HB partners (these are SSPs) for a single impression and each HB partner submits a bid. The advertiser connected to the HB partner with the highest bid wins the impression.

The data is from February 29, 2020 for two websites (referred to as website A and B) and contains, for each impression, the highest bid and second highest bid among the HB partners, and the hour of the day. We use these bids as proxies for the top two bids in a second-price auction in order to construct the joint distribution of the bids for the piece-wise stationary environment as described in Section 3.5.1. We perform the following pre-processing steps on the raw data. Let L_{MAX} denote a list of numbers containing the maximum bid of all the HB partners for each auction. First, we determine the 95-th percentile q_{MAX} of the positive values in L_{MAX} and keep all the HB auctions where values in L_{MAX} are at most q_{MAX}. Next, to preserve proprietary information, we shift the bids of all HB partners by a small positive constant v. Finally, we normalize the bids of all the HB partners to the range $[0, 1]$ by dividing by $q_{MAX} + v$. After these steps, we end up with a list $L_{W,h}$ for website W and hour h. Each element in the list $L_{W,h}$ is a tuple (a, b) where a denotes the highest bid and b denotes the second highest bid.

Let $\mathcal{T} = \cup_{i=1}^{M} \mathcal{T}_i$ be a partition of $\mathcal{T}$ as described in Section 3.5.1. We use the following procedure to construct a time series of length T for the bids. First, we define $\mathcal{T}_i = \{(i-1) \cdot (T/M) + 1, \ldots, (T/M) + (i-1) \cdot (T/M)\}$ for segment $i \in \{1, \ldots, M\}$. Second, for round $t \in \mathcal{T}_i$, we sample a tuple (a_t, b_t) uniformly at random with replacement from $L_{W,h}$ and the highest bid equals $X_t = a_t$ and the second highest bid equals $Y_t = b_t$.

3.7.2 Benchmark algorithms

We compare the performance of BMAB-SPAR-NS with the following benchmark algorithms: MUCB, EXP3-S, EXP3-BGZ, EXP3-P and SHIFTBAND. These algorithms are designed for non-stationary environments and adversarial environments

and serve as natural benchmarks for our problem. MUCB was recently proposed by [39] for piece-wise stationary environments. EXP3-S and EXP3-P were proposed by [15] and are multi-armed bandit algorithms that have performance guarantees under adversarial environments. The authors in [26] develop a variant of EXP3-S (which we will refer to as EXP3-BGZ), which uses the concept of a variation budget that describes the degree of non-stationarity of the reward distributions of the arms over the horizon T. It is similar to EXP3-S but tuned in a different way based on the variation budget. SHIFTBAND was proposed by [16] and is designed for adversarial environments. In contrast to EXP3-S, SHIFTBAND uses upper confidence bounds in order to manage the exploration-exploitation trade-off.

3.7.3 Settings and Performance Metrics

We consider a horizon of $T = 450000$ and $M = 10$ segments. For both website A and B, we consider two settings for the non-stationary environment. More specifically, for website A, we use $L_{A,h}$ with $h \in \{1, 2, \ldots, 10\}$ (this is denoted as Set A1) and $L_{A,h}$ with $h \in \{13, 14, \ldots, 23\}$ (this is denoted as Set A2). In Set A1, $L_{A,1}$ is used in segment 1, $L_{A,2}$ is used in segment 2, etc. Similarly, in Set A2, $L_{A,13}$ is used in segment 1, $L_{A,14}$ is used in segment 2, etc. For website B, Set B1 and Set B2 are defined in a similar way. All of the aforementioned algorithms are run with $K = 15$ arms which are equally spaced in the interval $[0.0, 0.2]$. We use the same performance metrics as in the stationary case (see Section 3.6.3). All results are averaged over 200 independent runs. We run EXP3-BGZ with a variation budget of 0.8. Following [16], we run SHIFTBAND (using notation from the original paper) with $\delta = 0.05$. The parameters of all other benchmark algorithms are set according to the recommendations in the respective papers. We run BMAB-SPAR-NS with $(a_{j,0} = 1.0, b_{j,0} = 1.0)$ for all j, $w = 1000$, $q = 500$, $\kappa = 0.99$, $\tau = 0.003$ and $p^{ex} = 0.015$. A detailed sensitivity analysis with respect to the parameters can be found in the Appendix (Section 3.D).

3.7.4 Results

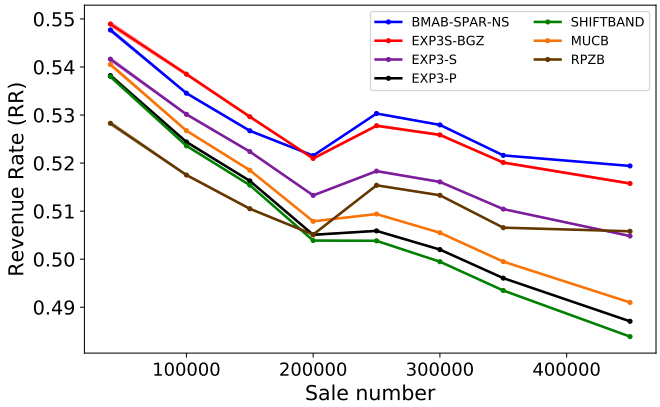
Figures 3.6-3.7 show the revenue rate (averaged over 200 independent runs) for website A and B. Results for the average cumulative return can be found in the Appedix).

In general, BMAB-SPAR-NS outperforms the benchmark algorithms. The difference in revenue rate relative to the best performing benchmark algorithm ranges from 0.5% to 2.5% depending on the number of elapsed rounds and the website. The best performing benchmark algorithms are EXP3-BGZ and EXP3-S. Note that, similar to BMAB-SPAR, BMAB-SPAR-NS is able to update information about arm j based on pulls from arms $k \neq j$. A possible explanation for the superior performance of BMAB-SPAR-NS could be related to this "cross-learning" feature. Due to the non-stationary environment, the learning problem becomes harder, and one would

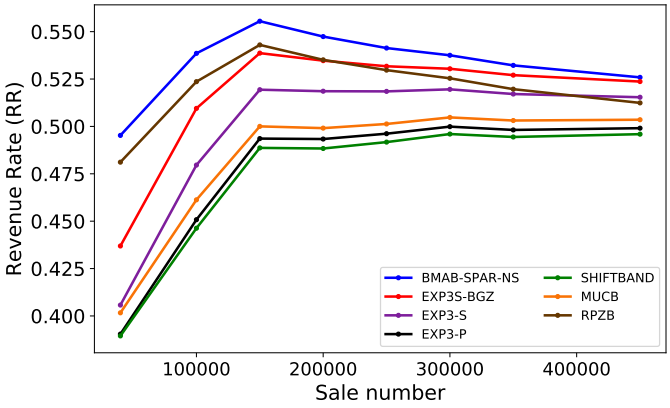
conjecture that this "cross-learning" feature has more added value since it allows the algorithm to quickly detect changes in the environment. The results appear to support this conjecture.

Recall that a piece-wise stationary environment is a special case of an adversarial environment. Another possible explanation for superior performance of BMAB-SPAR-NS could be that the bandit algorithms EXP3-BGZ, EXP3-S, EXP3-P and SHIFTBAND are too conservative because they assume that the environment is adversarial.

An explanation for the poor performance of MUCB could be related to the design of the algorithm and the piece-wise stationary environment itself. Due to the piece-wise stationary environment, the optimal reserve price can vary across segments. Furthermore, the expected return of the optimal reserve price may change, even if the optimal reserve price does not differ across segments. MUCB uses a change-point detection test, and each time that a change is detected, all arms get "reset" and an exploration phase with uniform sampling gets triggered. It could be that the piece-wise stationary environment for the reserve optimization problem leads to too many detected changes and to too many "resets", which in turn leads to poor performance of MUCB.

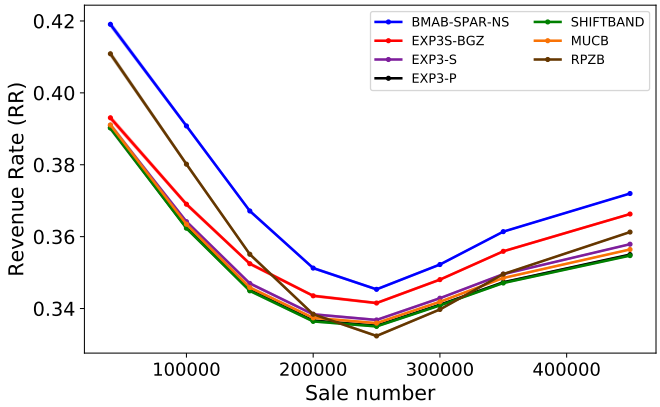


(a) Set A1.

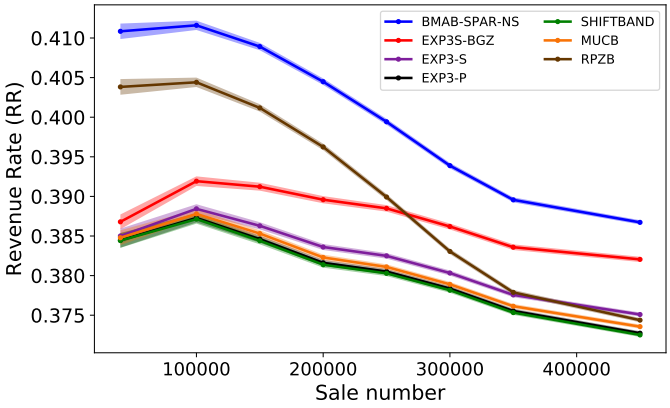


(b) Set A2.

Figure 3.6: Performance of the algorithms for website A, averaged over 200 runs. Shaded region indicates 95% confidence interval.



(a) Set B1.



(b) Set B2.

Figure 3.7: Performance of the algorithms for website B, averaged over 200 runs. Shaded region indicates 95% confidence interval.

3.8 Conclusion

The control of reserve prices in second-price auctions plays a key role in revenue optimization in online advertising. In this Chapter, we studied the reserve price optimization problem from the perspective of an online publisher. We considered a limited information setting where the values of the bids are not revealed to the publisher and no historical dataset containing the values of the bids is available. The main contribution of this Chapter is a method that incorporates knowledge about the rules of second-price auctions into a multi-armed bandit framework for optimizing reserve prices. Furthermore, the ideas behind our proposed method can be applied in both stationary and non-stationary environments. The experiments show that by incorporating the rules of second-price auctions, one can often improve upon the performance that can be obtained by traditional bandit algorithms.

In this Chapter we considered reserve price optimization on a single ad exchange. The next Chapter tackles the problem of reserve price optimization when the publisher can access multiple ad exchanges at the same time.

Appendix

3.A Additional results for Section 3.6

In this section we present tables that show the performance of the algorithms discussed in Section 3.6 of the main text.

Table 3.2: Performance of algorithms on eBay-1 dataset with $K = 30$.

BFPH												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.185	0.185	0.185	0.185	0.185	0.185	0.738	0.738	0.738	0.738	0.738	0.738
std	0.021	0.020	0.020	0.020	0.020	0.020	0.007	0.006	0.005	0.005	0.005	0.005
RPZB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.185	0.185	0.185	0.185	0.185	0.185	0.738	0.738	0.738	0.738	0.738	0.738
std	0.021	0.020	0.020	0.020	0.020	0.020	0.007	0.006	0.005	0.005	0.005	0.005
BMAB-SPAR												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.176	0.179	0.179	0.179	0.180	0.180	0.703	0.714	0.714	0.715	0.715	0.716
std	0.020	0.020	0.020	0.020	0.020	0.020	0.006	0.004	0.004	0.004	0.004	0.004
TS												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.120	0.152	0.158	0.162	0.169	0.174	0.476	0.604	0.629	0.644	0.673	0.694
std	0.022	0.021	0.021	0.021	0.020	0.020	0.035	0.017	0.014	0.012	0.009	0.007
MOSSA												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.129	0.157	0.162	0.165	0.172	0.176	0.510	0.624	0.644	0.658	0.684	0.702
std	0.022	0.021	0.021	0.021	0.020	0.020	0.029	0.014	0.011	0.009	0.006	0.004
UCB1												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.094	0.111	0.116	0.121	0.131	0.142	0.373	0.438	0.460	0.477	0.520	0.564
std	0.018	0.020	0.021	0.021	0.022	0.022	0.030	0.031	0.031	0.031	0.028	0.023
UCB2												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.109	0.136	0.143	0.148	0.158	0.167	0.430	0.539	0.568	0.587	0.629	0.663
std	0.021	0.022	0.022	0.022	0.021	0.021	0.034	0.025	0.022	0.020	0.013	0.008
UCB-V												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.089	0.111	0.119	0.125	0.141	0.156	0.352	0.437	0.470	0.496	0.561	0.620
std	0.016	0.019	0.020	0.020	0.021	0.020	0.027	0.028	0.027	0.025	0.018	0.010
OCUCB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.098	0.129	0.139	0.146	0.161	0.172	0.388	0.511	0.550	0.580	0.641	0.683
std	0.019	0.022	0.022	0.022	0.021	0.021	0.030	0.029	0.025	0.021	0.011	0.004
KL-UCB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.122	0.149	0.154	0.158	0.165	0.171	0.483	0.590	0.612	0.627	0.657	0.681
std	0.021	0.021	0.021	0.020	0.020	0.020	0.028	0.015	0.012	0.010	0.006	0.004

Table 3.3: Performance of algorithms on eBay-2 dataset with $K = 30$.

BFPH												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.196	0.195	0.195	0.195	0.195	0.195	0.554	0.553	0.553	0.553	0.553	0.553
std	0.024	0.024	0.024	0.024	0.024	0.024	0.012	0.011	0.011	0.010	0.010	0.010
RPZB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.177	0.177	0.177	0.177	0.177	0.177	0.500	0.500	0.500	0.500	0.500	0.500
std	0.025	0.025	0.025	0.025	0.025	0.025	0.022	0.021	0.021	0.021	0.021	0.021
BMAB-SPAR												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.187	0.188	0.189	0.189	0.188	0.189	0.529	0.533	0.534	0.534	0.534	0.534
std	0.025	0.025	0.025	0.025	0.025	0.025	0.016	0.016	0.015	0.015	0.015	0.015
TS												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.151	0.169	0.172	0.175	0.180	0.184	0.427	0.476	0.487	0.494	0.509	0.521
std	0.026	0.025	0.024	0.024	0.024	0.023	0.031	0.023	0.020	0.019	0.016	0.014
MOSSA												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.154	0.172	0.176	0.178	0.183	0.186	0.434	0.486	0.497	0.503	0.517	0.528
std	0.026	0.025	0.025	0.024	0.024	0.024	0.030	0.021	0.019	0.018	0.015	0.013
UCB1												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.136	0.147	0.150	0.152	0.158	0.164	0.384	0.413	0.423	0.429	0.447	0.464
std	0.025	0.025	0.025	0.025	0.026	0.026	0.034	0.031	0.030	0.029	0.027	0.025
UCB2												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.146	0.161	0.165	0.168	0.173	0.179	0.410	0.454	0.465	0.473	0.490	0.505
std	0.026	0.026	0.025	0.025	0.025	0.024	0.033	0.027	0.025	0.023	0.020	0.017
UCB-V												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.131	0.143	0.148	0.151	0.160	0.169	0.369	0.403	0.416	0.426	0.452	0.477
std	0.023	0.025	0.025	0.025	0.025	0.024	0.031	0.030	0.029	0.028	0.025	0.020
OCUCB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.139	0.157	0.163	0.167	0.175	0.183	0.392	0.443	0.459	0.471	0.496	0.518
std	0.025	0.026	0.026	0.025	0.025	0.024	0.032	0.028	0.026	0.024	0.019	0.015
KL-UCB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.152	0.166	0.170	0.172	0.177	0.181	0.429	0.470	0.480	0.486	0.500	0.512
std	0.025	0.025	0.024	0.024	0.024	0.024	0.029	0.022	0.020	0.019	0.017	0.015

Table 3.4: Performance of algorithms on eBay-3 dataset with $K = 30$.

BFPH												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.207	0.207	0.207	0.207	0.207	0.207	0.656	0.657	0.657	0.657	0.657	0.657
std	0.021	0.020	0.020	0.020	0.020	0.020	0.019	0.019	0.019	0.019	0.019	0.019
RPZB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.207	0.207	0.207	0.206	0.207	0.207	0.654	0.655	0.655	0.655	0.655	0.655
std	0.021	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.019	0.019	0.020	0.020
BMAB-SPAR												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.203	0.205	0.205	0.205	0.205	0.205	0.644	0.650	0.650	0.650	0.651	0.651
std	0.021	0.020	0.020	0.020	0.020	0.020	0.019	0.018	0.018	0.018	0.018	0.018
TS												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.161	0.179	0.184	0.187	0.192	0.197	0.509	0.568	0.582	0.591	0.609	0.624
std	0.021	0.020	0.020	0.020	0.020	0.020	0.030	0.024	0.022	0.022	0.021	0.019
MOSSA												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.165	0.183	0.187	0.190	0.195	0.198	0.522	0.580	0.593	0.601	0.617	0.629
std	0.021	0.020	0.020	0.020	0.020	0.020	0.029	0.023	0.022	0.022	0.021	0.020
UCB1												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.148	0.157	0.160	0.162	0.169	0.175	0.467	0.496	0.506	0.514	0.533	0.554
std	0.021	0.020	0.020	0.020	0.021	0.021	0.033	0.028	0.027	0.027	0.026	0.024
UCB2												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.156	0.171	0.176	0.178	0.185	0.191	0.493	0.542	0.556	0.565	0.587	0.606
std	0.022	0.021	0.021	0.021	0.020	0.020	0.032	0.026	0.025	0.024	0.022	0.021
UCB-V												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.142	0.153	0.157	0.160	0.170	0.180	0.450	0.483	0.497	0.508	0.538	0.569
std	0.020	0.020	0.020	0.020	0.020	0.020	0.031	0.028	0.027	0.026	0.024	0.021
OCUCB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.150	0.167	0.173	0.177	0.188	0.196	0.473	0.528	0.547	0.561	0.594	0.621
std	0.021	0.021	0.021	0.021	0.020	0.020	0.032	0.027	0.025	0.024	0.022	0.020
KL-UCB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.161	0.176	0.180	0.183	0.188	0.193	0.510	0.558	0.571	0.579	0.597	0.612
std	0.021	0.020	0.020	0.020	0.020	0.020	0.028	0.023	0.022	0.021	0.020	0.019

Table 3.5: Performance of algorithms on eBay-4 dataset with $K = 30$.

BFPH												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.200	0.199	0.199	0.199	0.199	0.199	0.594	0.593	0.593	0.593	0.593	0.593
std	0.019	0.019	0.019	0.019	0.019	0.019	0.012	0.010	0.010	0.011	0.011	0.011
RPZB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.191	0.190	0.190	0.190	0.190	0.190	0.567	0.566	0.567	0.567	0.567	0.567
std	0.020	0.020	0.020	0.020	0.020	0.020	0.018	0.017	0.017	0.017	0.017	0.017
BMAB-SPAR												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.194	0.195	0.195	0.195	0.195	0.195	0.576	0.580	0.580	0.581	0.581	0.581
std	0.020	0.019	0.019	0.019	0.019	0.019	0.014	0.012	0.012	0.012	0.012	0.012
TS												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.157	0.172	0.176	0.179	0.184	0.188	0.464	0.513	0.525	0.532	0.548	0.559
std	0.020	0.019	0.019	0.019	0.019	0.019	0.025	0.017	0.016	0.015	0.014	0.013
MOSSA												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.159	0.175	0.179	0.181	0.186	0.190	0.471	0.521	0.532	0.540	0.554	0.566
std	0.020	0.020	0.020	0.020	0.019	0.019	0.024	0.018	0.016	0.015	0.013	0.012
UCB1												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.142	0.151	0.154	0.156	0.162	0.168	0.419	0.448	0.458	0.465	0.482	0.501
std	0.020	0.020	0.020	0.020	0.020	0.020	0.027	0.024	0.024	0.023	0.022	0.020
UCB2												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.151	0.165	0.169	0.172	0.178	0.183	0.447	0.490	0.502	0.510	0.528	0.544
std	0.020	0.020	0.020	0.020	0.020	0.020	0.026	0.020	0.020	0.019	0.017	0.015
UCB-V												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.137	0.147	0.151	0.155	0.164	0.172	0.405	0.436	0.449	0.459	0.486	0.513
std	0.019	0.020	0.020	0.020	0.020	0.019	0.027	0.024	0.024	0.023	0.020	0.016
OCUCB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.145	0.161	0.167	0.171	0.180	0.187	0.428	0.478	0.495	0.507	0.535	0.557
std	0.020	0.020	0.021	0.020	0.020	0.019	0.028	0.023	0.021	0.020	0.016	0.014
KL-UCB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.157	0.170	0.174	0.176	0.181	0.185	0.464	0.506	0.516	0.523	0.538	0.551
std	0.020	0.019	0.019	0.019	0.019	0.019	0.024	0.017	0.016	0.015	0.014	0.013

Table 3.6: Performance of algorithms on eBay-1 dataset with $K = 60$.

BFPH												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.185	0.185	0.185	0.185	0.185	0.185	0.738	0.738	0.738	0.738	0.738	0.738
std	0.021	0.020	0.020	0.020	0.020	0.020	0.007	0.006	0.005	0.005	0.005	0.005
RPZB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.185	0.185	0.185	0.185	0.185	0.185	0.738	0.738	0.738	0.738	0.738	0.738
std	0.021	0.020	0.020	0.020	0.020	0.020	0.007	0.006	0.005	0.005	0.005	0.005
BMAB-SPAR												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.174	0.179	0.179	0.180	0.180	0.180	0.692	0.713	0.714	0.715	0.716	0.717
std	0.020	0.020	0.020	0.020	0.020	0.020	0.006	0.004	0.003	0.003	0.003	0.004
TS												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.106	0.137	0.145	0.150	0.160	0.168	0.419	0.543	0.574	0.594	0.636	0.668
std	0.020	0.021	0.021	0.021	0.020	0.020	0.033	0.024	0.021	0.018	0.011	0.007
MOSSA												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.115	0.146	0.152	0.156	0.165	0.171	0.455	0.579	0.605	0.621	0.656	0.682
std	0.021	0.022	0.021	0.021	0.021	0.021	0.031	0.021	0.017	0.015	0.009	0.006
UCB1												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.090	0.102	0.106	0.110	0.119	0.129	0.355	0.404	0.420	0.434	0.471	0.512
std	0.017	0.019	0.020	0.020	0.021	0.022	0.029	0.031	0.031	0.032	0.032	0.030
UCB2												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.099	0.122	0.129	0.134	0.146	0.156	0.393	0.482	0.510	0.530	0.577	0.621
std	0.019	0.022	0.022	0.022	0.022	0.021	0.034	0.031	0.029	0.027	0.021	0.015
UCB-V												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.085	0.098	0.104	0.109	0.123	0.139	0.336	0.388	0.411	0.430	0.486	0.550
std	0.015	0.018	0.019	0.019	0.021	0.021	0.022	0.028	0.029	0.029	0.027	0.020
OCUCB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.093	0.117	0.126	0.133	0.151	0.165	0.367	0.463	0.499	0.527	0.600	0.658
std	0.018	0.021	0.022	0.022	0.022	0.021	0.030	0.032	0.031	0.028	0.018	0.009
KL-UCB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.109	0.135	0.142	0.146	0.156	0.163	0.432	0.536	0.562	0.580	0.619	0.650
std	0.020	0.021	0.021	0.021	0.020	0.020	0.032	0.022	0.019	0.016	0.011	0.007

Table 3.7: Performance of algorithms on eBay-2 dataset with $K = 60$.

BFPH												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.199	0.198	0.199	0.198	0.198	0.198	0.563	0.563	0.563	0.563	0.563	0.562
std	0.024	0.023	0.023	0.023	0.023	0.023	0.010	0.009	0.009	0.009	0.008	0.008
RPZB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.177	0.177	0.177	0.177	0.177	0.177	0.500	0.500	0.500	0.500	0.500	0.500
std	0.025	0.025	0.025	0.025	0.025	0.025	0.022	0.021	0.021	0.021	0.021	0.021
BMAB-SPAR												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.185	0.188	0.188	0.188	0.188	0.189	0.524	0.532	0.533	0.533	0.534	0.534
std	0.025	0.025	0.025	0.025	0.025	0.025	0.018	0.017	0.017	0.017	0.017	0.017
TS												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.144	0.161	0.165	0.168	0.174	0.179	0.405	0.454	0.467	0.475	0.493	0.508
std	0.026	0.025	0.025	0.024	0.024	0.024	0.033	0.025	0.023	0.021	0.018	0.015
MOSSA												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.147	0.166	0.170	0.173	0.179	0.184	0.414	0.468	0.480	0.489	0.507	0.522
std	0.026	0.025	0.025	0.025	0.024	0.024	0.032	0.024	0.021	0.019	0.016	0.013
UCB1												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.134	0.142	0.145	0.147	0.152	0.158	0.377	0.400	0.408	0.413	0.428	0.445
std	0.025	0.025	0.025	0.025	0.026	0.026	0.033	0.031	0.031	0.031	0.030	0.028
UCB2												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.140	0.154	0.157	0.160	0.166	0.173	0.395	0.433	0.444	0.452	0.470	0.488
std	0.026	0.026	0.026	0.026	0.025	0.025	0.034	0.030	0.028	0.027	0.024	0.021
UCB-V												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.129	0.137	0.140	0.143	0.150	0.159	0.363	0.385	0.394	0.402	0.423	0.449
std	0.024	0.024	0.024	0.024	0.025	0.025	0.032	0.030	0.030	0.030	0.028	0.025
OCUCB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.136	0.151	0.156	0.160	0.170	0.179	0.383	0.425	0.440	0.452	0.480	0.507
std	0.025	0.026	0.026	0.026	0.025	0.024	0.032	0.030	0.029	0.027	0.022	0.016
KL-UCB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.146	0.160	0.163	0.166	0.171	0.176	0.410	0.451	0.461	0.468	0.484	0.499
std	0.025	0.025	0.025	0.025	0.024	0.024	0.031	0.025	0.024	0.022	0.019	0.017

Table 3.8: Performance of algorithms on eBay-3 dataset with $K = 60$.

BFPH												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.208	0.208	0.208	0.208	0.208	0.208	0.658	0.659	0.659	0.659	0.659	0.659
std	0.021	0.020	0.020	0.020	0.020	0.020	0.019	0.018	0.018	0.018	0.018	0.018
RPZB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.207	0.207	0.207	0.206	0.207	0.207	0.654	0.655	0.655	0.655	0.655	0.655
std	0.021	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.019	0.019	0.020	0.020
BMAB-SPAR												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.202	0.205	0.205	0.205	0.205	0.205	0.638	0.649	0.650	0.650	0.651	0.651
std	0.021	0.020	0.020	0.020	0.020	0.020	0.019	0.018	0.018	0.018	0.018	0.018
TS												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.154	0.171	0.175	0.178	0.185	0.191	0.487	0.541	0.555	0.565	0.587	0.606
std	0.021	0.020	0.020	0.020	0.020	0.020	0.031	0.024	0.023	0.022	0.021	0.020
MOSSA												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.158	0.176	0.181	0.184	0.190	0.195	0.500	0.558	0.572	0.581	0.602	0.618
std	0.021	0.020	0.020	0.020	0.020	0.020	0.031	0.024	0.023	0.022	0.021	0.020
UCB1												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.146	0.152	0.155	0.157	0.162	0.168	0.461	0.482	0.490	0.496	0.512	0.531
std	0.021	0.020	0.020	0.020	0.020	0.020	0.033	0.029	0.028	0.027	0.027	0.026
UCB2												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.151	0.163	0.167	0.170	0.177	0.184	0.478	0.517	0.530	0.539	0.561	0.583
std	0.021	0.021	0.021	0.021	0.021	0.020	0.032	0.027	0.027	0.026	0.024	0.022
UCB-V												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.141	0.147	0.150	0.152	0.160	0.169	0.446	0.465	0.474	0.482	0.505	0.533
std	0.020	0.019	0.019	0.019	0.020	0.020	0.032	0.028	0.027	0.027	0.026	0.024
OCUCB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.147	0.161	0.166	0.170	0.181	0.191	0.465	0.508	0.525	0.538	0.573	0.606
std	0.020	0.020	0.021	0.021	0.020	0.020	0.030	0.028	0.027	0.026	0.023	0.020
KL-UCB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.155	0.168	0.172	0.175	0.181	0.187	0.490	0.533	0.545	0.554	0.574	0.593
std	0.021	0.020	0.020	0.020	0.020	0.020	0.030	0.024	0.023	0.022	0.021	0.020

Table 3.9: Performance of algorithms on eBay-4 dataset with $K = 60$.

BFPH												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.203	0.202	0.202	0.202	0.202	0.202	0.602	0.601	0.602	0.602	0.602	0.602
std	0.019	0.018	0.019	0.019	0.018	0.019	0.012	0.011	0.011	0.011	0.011	0.011
RPZB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.191	0.190	0.190	0.190	0.190	0.190	0.567	0.566	0.567	0.567	0.567	0.567
std	0.020	0.020	0.020	0.020	0.020	0.020	0.018	0.017	0.017	0.017	0.017	0.017
BMAB-SPAR												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.192	0.194	0.194	0.195	0.195	0.195	0.570	0.578	0.579	0.580	0.580	0.580
std	0.019	0.019	0.019	0.019	0.019	0.019	0.014	0.012	0.012	0.013	0.013	0.013
TS												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.148	0.164	0.169	0.172	0.178	0.183	0.439	0.489	0.502	0.511	0.530	0.546
std	0.020	0.020	0.020	0.020	0.019	0.019	0.026	0.020	0.019	0.018	0.016	0.014
MOSSA												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.152	0.169	0.173	0.176	0.182	0.188	0.451	0.502	0.515	0.524	0.543	0.559
std	0.020	0.020	0.020	0.020	0.019	0.019	0.027	0.019	0.018	0.017	0.015	0.013
UCB1												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.139	0.146	0.149	0.151	0.156	0.162	0.413	0.435	0.442	0.448	0.464	0.481
std	0.019	0.020	0.020	0.020	0.020	0.020	0.026	0.025	0.025	0.025	0.024	0.022
UCB2												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.146	0.157	0.161	0.164	0.171	0.177	0.432	0.468	0.479	0.487	0.507	0.526
std	0.020	0.020	0.021	0.021	0.020	0.020	0.027	0.023	0.023	0.022	0.019	0.017
UCB-V												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.136	0.141	0.144	0.147	0.154	0.162	0.402	0.420	0.428	0.436	0.457	0.483
std	0.019	0.019	0.019	0.019	0.020	0.020	0.026	0.024	0.024	0.024	0.022	0.020
OCUCB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.141	0.155	0.160	0.164	0.174	0.183	0.418	0.460	0.475	0.487	0.518	0.546
std	0.019	0.020	0.021	0.021	0.020	0.019	0.026	0.024	0.023	0.022	0.018	0.014
KL-UCB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	1000	5000	7500	10000	20000	40000	1000	5000	7500	10000	20000	40000
mean	0.149	0.163	0.167	0.169	0.175	0.180	0.443	0.484	0.496	0.503	0.521	0.536
std	0.020	0.020	0.020	0.020	0.019	0.019	0.027	0.020	0.019	0.018	0.016	0.014

3.B Additional results for Section 3.7

In this section we present tables that show the performance of the algorithms discussed in Section 3.7 of the main text.

Table 3.10: Performance of algorithms for set A1.

BMAB-SPAR-NS												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.081	0.080	0.071	0.064	0.058	0.055	0.548	0.535	0.521	0.528	0.520	0.519
std	0.001	0.000	0.000	0.000	0.000	0.000	0.003	0.002	0.001	0.001	0.001	0.001
RPZB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.078	0.078	0.069	0.062	0.056	0.054	0.528	0.518	0.505	0.513	0.506	0.506
std	0.001	0.000	0.000	0.000	0.000	0.000	0.003	0.002	0.001	0.001	0.001	0.001
EXP3-BGZ												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.081	0.081	0.071	0.063	0.058	0.055	0.549	0.538	0.521	0.526	0.518	0.516
std	0.001	0.000	0.000	0.000	0.000	0.000	0.004	0.002	0.002	0.002	0.002	0.002
EXP3-S												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.080	0.080	0.070	0.062	0.057	0.054	0.542	0.530	0.513	0.516	0.508	0.505
std	0.001	0.000	0.000	0.000	0.000	0.000	0.003	0.002	0.001	0.001	0.001	0.001
EXP3-P												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.080	0.079	0.069	0.060	0.055	0.052	0.538	0.524	0.505	0.502	0.492	0.487
std	0.001	0.000	0.000	0.000	0.000	0.000	0.003	0.002	0.001	0.001	0.001	0.001
SHIFTBAND												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.080	0.079	0.069	0.060	0.055	0.051	0.538	0.524	0.504	0.499	0.489	0.484
std	0.001	0.000	0.000	0.000	0.000	0.000	0.003	0.002	0.001	0.001	0.001	0.001
MUCB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.080	0.079	0.069	0.061	0.055	0.052	0.541	0.527	0.508	0.505	0.496	0.491
std	0.001	0.000	0.000	0.000	0.000	0.000	0.003	0.001	0.001	0.001	0.001	0.001

Table 3.11: Performance of algorithms for set A2.

BMAB-SPAR-NS												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.033	0.041	0.054	0.061	0.062	0.064	0.495	0.538	0.547	0.538	0.528	0.526
std	0.000	0.000	0.000	0.000	0.000	0.000	0.004	0.002	0.001	0.001	0.001	0.001
RPZB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.032	0.039	0.053	0.059	0.060	0.062	0.481	0.524	0.535	0.525	0.515	0.512
std	0.000	0.000	0.000	0.000	0.000	0.000	0.004	0.002	0.002	0.001	0.001	0.001
EXP3-BGZ												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.029	0.038	0.053	0.060	0.062	0.064	0.437	0.509	0.535	0.530	0.524	0.524
std	0.000	0.000	0.000	0.000	0.000	0.000	0.004	0.003	0.002	0.002	0.001	0.001
EXP3-S												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.027	0.036	0.051	0.059	0.061	0.063	0.406	0.479	0.519	0.520	0.515	0.515
std	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.002	0.002	0.001	0.001	0.001
EXP3-P												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.026	0.034	0.049	0.057	0.058	0.061	0.391	0.451	0.493	0.500	0.497	0.499
std	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.002	0.001	0.001	0.001	0.001
SHIFTBAND												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.026	0.034	0.048	0.056	0.058	0.060	0.390	0.446	0.488	0.496	0.494	0.496
std	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.002	0.001	0.001	0.001	0.001
MUCB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.026	0.035	0.049	0.057	0.059	0.061	0.401	0.461	0.499	0.505	0.502	0.504
std	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.002	0.002	0.001	0.001	0.001

Table 3.12: Performance of algorithms for set B1.

BMAB-SPAR-NS												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.015	0.012	0.011	0.012	0.013	0.014	0.419	0.391	0.351	0.352	0.369	0.372
std	0.000	0.000	0.000	0.000	0.000	0.000	0.005	0.004	0.002	0.002	0.001	0.001
RPZB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.015	0.012	0.011	0.012	0.013	0.013	0.411	0.380	0.338	0.340	0.358	0.361
std	0.000	0.000	0.000	0.000	0.000	0.000	0.005	0.004	0.002	0.002	0.001	0.001
EXP3-BGZ												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.014	0.012	0.011	0.012	0.013	0.014	0.393	0.369	0.343	0.348	0.363	0.366
std	0.000	0.000	0.000	0.000	0.000	0.000	0.005	0.004	0.002	0.002	0.002	0.002
EXP3-S												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.014	0.012	0.011	0.012	0.013	0.013	0.391	0.364	0.338	0.343	0.355	0.358
std	0.000	0.000	0.000	0.000	0.000	0.000	0.005	0.003	0.002	0.002	0.001	0.001
EXP3-P												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.014	0.011	0.011	0.012	0.013	0.013	0.390	0.362	0.336	0.341	0.352	0.355
std	0.000	0.000	0.000	0.000	0.000	0.000	0.005	0.003	0.002	0.002	0.001	0.001
SHIFTBAND												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.014	0.011	0.011	0.012	0.013	0.013	0.390	0.362	0.336	0.341	0.352	0.355
std	0.000	0.000	0.000	0.000	0.000	0.000	0.005	0.003	0.002	0.002	0.001	0.001
MUCB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.014	0.011	0.011	0.012	0.013	0.013	0.391	0.363	0.337	0.342	0.353	0.356
std	0.000	0.000	0.000	0.000	0.000	0.000	0.005	0.003	0.002	0.002	0.001	0.001

Table 3.13: Performance of algorithms for set B2.

BMAB-SPAR-NS												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.016	0.016	0.016	0.015	0.015	0.015	0.411	0.412	0.404	0.394	0.387	0.386
std	0.000	0.000	0.000	0.000	0.000	0.000	0.005	0.003	0.002	0.002	0.002	0.002
RPZB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.015	0.015	0.015	0.015	0.014	0.014	0.404	0.404	0.396	0.383	0.375	0.374
std	0.000	0.000	0.000	0.000	0.000	0.000	0.005	0.003	0.002	0.002	0.002	0.001
EXP3-BGZ												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.015	0.015	0.015	0.015	0.015	0.014	0.387	0.392	0.390	0.386	0.383	0.382
std	0.000	0.000	0.000	0.000	0.000	0.000	0.005	0.003	0.003	0.002	0.002	0.002
EXP3-S												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.015	0.015	0.015	0.015	0.014	0.014	0.385	0.389	0.384	0.380	0.376	0.375
std	0.000	0.000	0.000	0.000	0.000	0.000	0.005	0.003	0.002	0.002	0.002	0.001
EXP3-P												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.015	0.015	0.015	0.015	0.014	0.014	0.385	0.387	0.382	0.378	0.374	0.373
std	0.000	0.000	0.000	0.000	0.000	0.000	0.005	0.003	0.002	0.002	0.002	0.001
SHIFTBAND												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.015	0.015	0.015	0.015	0.014	0.014	0.385	0.387	0.381	0.378	0.374	0.372
std	0.000	0.000	0.000	0.000	0.000	0.000	0.005	0.003	0.002	0.002	0.002	0.001
MUCB												
	Average Cumulative Return (ACR)						Revenue Rate (RR)					
horizon	40000	100000	200000	300000	400000	450000	40000	100000	200000	300000	400000	450000
mean	0.015	0.015	0.015	0.015	0.014	0.014	0.385	0.388	0.382	0.379	0.375	0.373
std	0.000	0.000	0.000	0.000	0.000	0.000	0.005	0.003	0.002	0.002	0.002	0.001

3.C Impact of alternative clustering

In this section we perform additional experiments related to Section 3.6 of the main text. In this section we study how the performance of the algorithms change if the clustering of the bids is performed in a different way. Similarly as in the main text, we define probability distributions over the clusters. However, in contrast with the main text, we do not use the relative gap between the second highest bid and the highest bid as a feature in the clustering.

The clustering is carried out as follows. First, we determine the 95-th percentile of $B_1 = \cup_{j=1}^{70213}\{b_1^j\}$. Denote the 95-th percentile by TP. Second, we remove outliers by removing pairs (b_1^j, b_2^j) for which the value of the top bid b_1^j exceeds the 95-th percentile TP, that is, we remove pairs j for which $b_1^j > TP$. We subsequently cluster the remaining bids (b_1^j, b_2^j) using the k-means clustering algorithm with $M = 6$ clusters. Next, we scale the remaining values of (b_1^j, b_2^j) by the maximum value of b_1^j the dataset so that all the bids are in the range $[0, 1]$. Figure 3.8 displays the resulting clustering of bids.

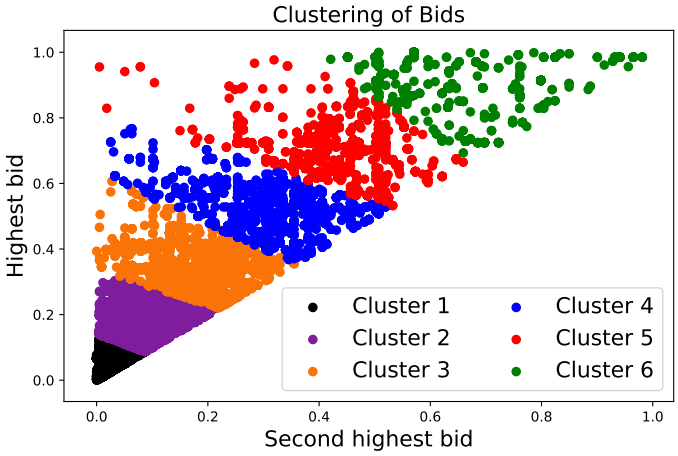


Figure 3.8: Clustering of bids in subset of eBay dataset.

The description of these probability distributions can be found in Table 3.14. In Table 3.14, η_i is defined as $\eta_i = \frac{\hat{\eta}_i}{\sum_{i=1}^{6}\hat{\eta}_i}$, where $\hat{\eta}_i$ is an i.i.d. draw from an uniform distribution on $[0.25, 0.75]$ for $i = 1, \ldots, 6$. In other words, in order to construct the distribution P_j, we sample the values of $\hat{\eta}_i$ for $i = 1, \ldots, 6$ and then we normalized these values so that they sum up 1. We perform 200 independent runs of the above procedure and in each run we sample new values for $\hat{\eta}_i$.

Dataset eBay-5 models a case where the joint distribution is (in expectation) more evenly spread across $[0, 1]$, dataset eBay-6 models a case where the joint distribution

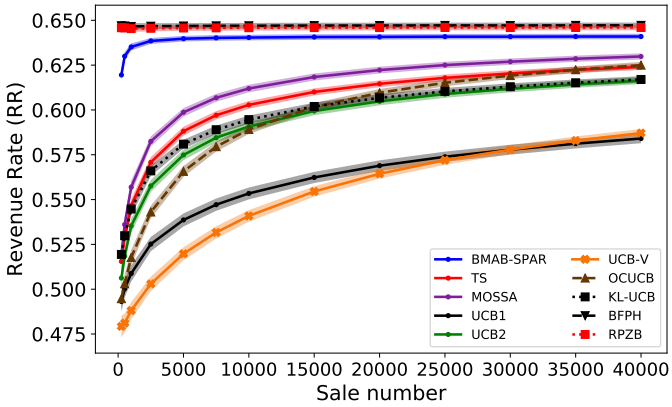
Table 3.14: Description of eBay datasets

dataset	distribution
eBay-5	$P_5 = (\eta_1, \eta_2, \eta_3, \eta_4, \eta_5, \eta_6)$
eBay-6	$P_6 = (\eta_1, \eta_2, 0.00, 0.00, 0.00, 0.00)$
eBay-7	$P_7 = (0.00, 0.00, 0.00, 0.00, \eta_1, \eta_2)$
eBay-8	$P_8 = (0.00, 0.00, \eta_1, \eta_2, 0.00, 0.00)$

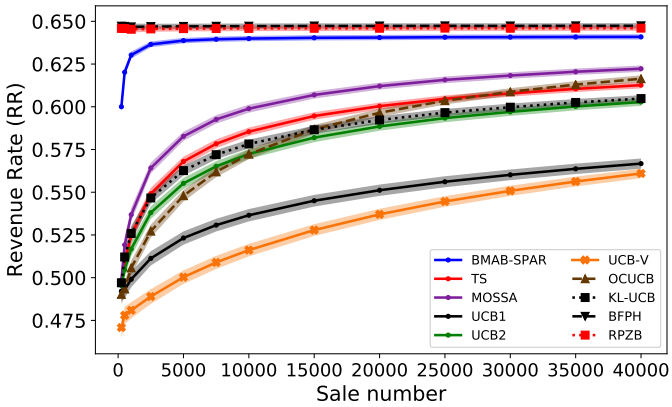
is concentrated in the lower part of $[0, 1]$, dataset eBay-7 models a case where the joint distribution is concentrated in the upper part of $[0, 1]$, and dataset eBay-8 models a case where the joint distribution is concentrated in the middle of $[0, 1]$.

We consider a horizon of $T = 40000$. We consider the same set of algorithms as in the main text and all of the algorithms are run with $K \in \{30, 60\}$ arms which are equally spaced in the interval $[0, 1]$. All results are averaged over 200 independent runs.

The results of the experiments can be found in Figures 3.9-3.12. The results are qualitatively similar to those reported in the main text. We again see that BMAB-SPAR tends to outperform the benchmark bandit algorithms. Furthermore, we again see that the revenue rate of BMAB-SPAR approaches that of BFPH rather quickly. Also, we see that the performance of BMAB-SPAR is not very sensitive with respect to the number of arms K.

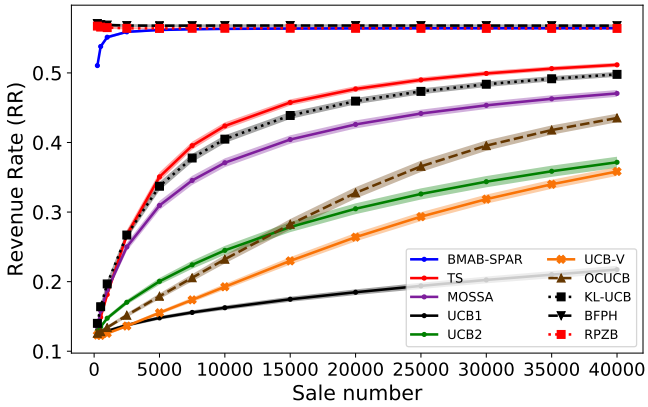


(a) $K = 30$.

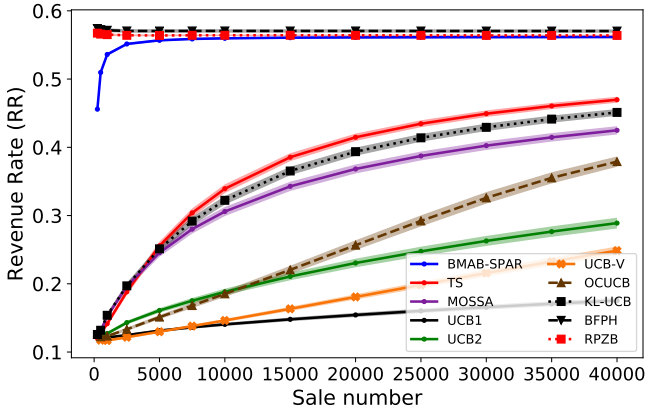


(b) $K = 60$.

Figure 3.9: Performance of the algorithms for dataset eBay-5, averaged over 200 runs. Lines indicate the mean and shaded region indicates 95% confidence interval.

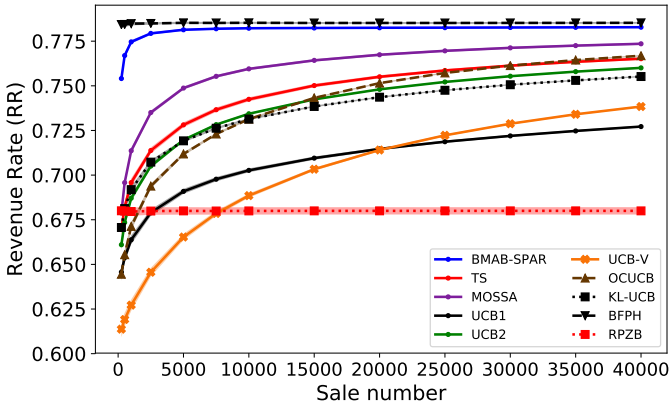


(a) $K = 30$.

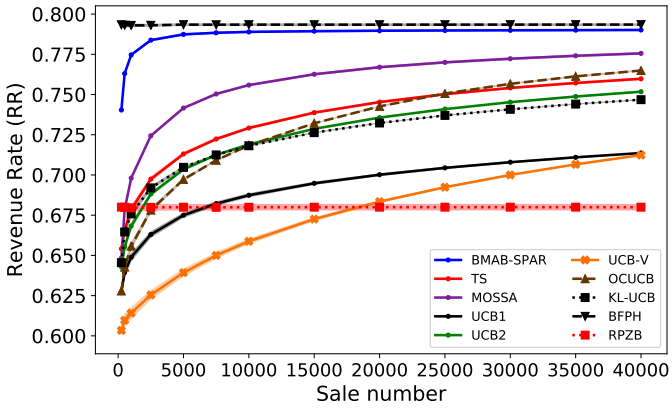


(b) $K = 60$.

Figure 3.10: Performance of the algorithms for dataset eBay-6, averaged over 200 runs. Lines indicate the mean and shaded region indicates 95% confidence interval.

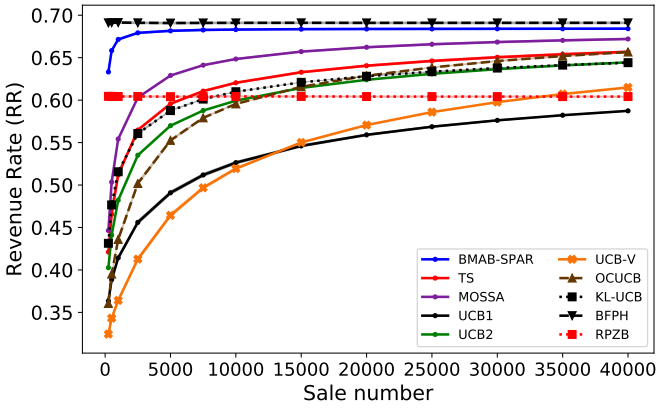


(a) $K = 30$.

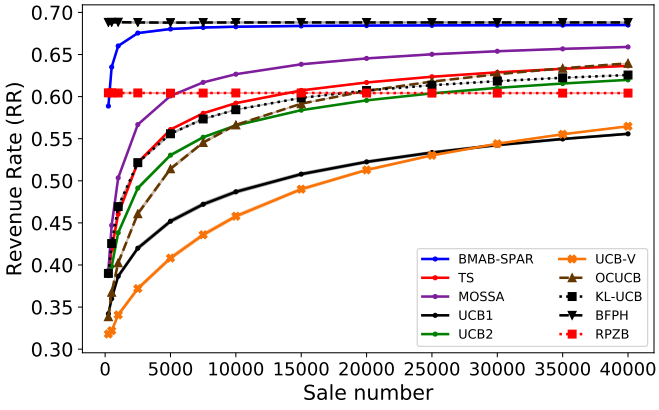


(b) $K = 60$.

Figure 3.11: Performance of the algorithms for dataset eBay-7, averaged over 200 runs. Lines indicate the mean and shaded region indicates 95% confidence interval.



(a) $K = 30$.



(b) $K = 60$.

Figure 3.12: Performance of the algorithms for dataset eBay-8, averaged over 200 runs. Lines indicate the mean and shaded region indicates 95% confidence interval.

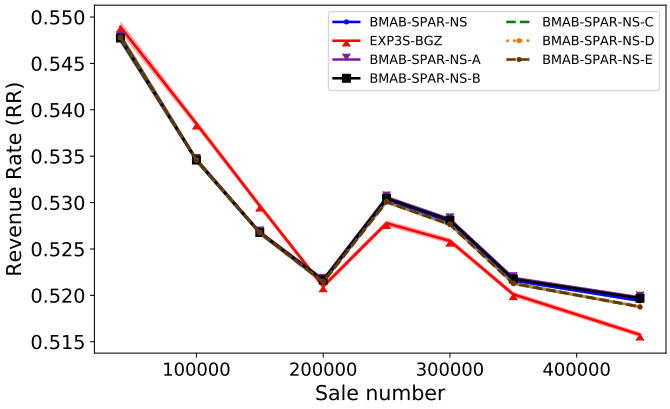
3.D Sensitivity analysis for BMAB-SPAR-NS

In this section we perform additional experiments related to Section 3.7 of the main text. In this section we present the results of a sensitivity analysis with respect to the parameters of BMAB-SPAR-NS. The experimental setting is identical to the setting in Section 3.7 of the main text, but we adjust several of the parameters of the BMAB-SPAR-NS algorithm in order to investigate the impact of these parameters on the performance. Table 3.15 below shows the parameter settings that are used in the sensitivity analysis and relates these to an abbreviation. In all of of the experiments we set $\kappa = 0.99$.

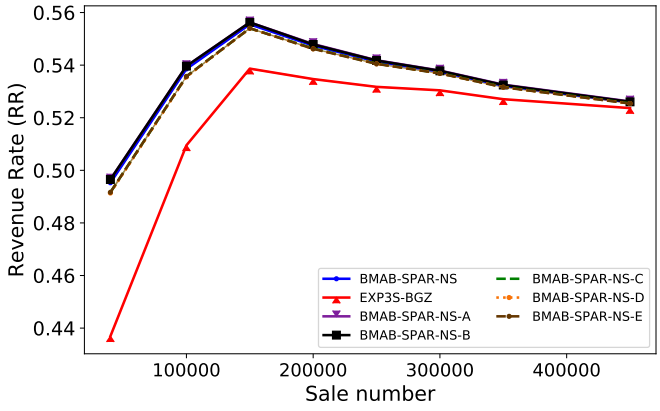
The results are presented in Figures 3.13 to 3.18. Figure 3.13 to 3.15 shows the performance for website A and Figure 3.16 to 3.18 shows the performance for website B. Overall, the results are qualitatively similar to those reported in the main text. The results indicate that, in general, the settings for BMAB-SPAR-NS lead to the best performance. Furthermore, we observe that performance of BMAB-SPAR-NS is competitive with the performance of the best performing benchmark algorithms for all of the parameter values considered.

Table 3.15: Parameter values for sensitivity analysis.

parameter values	Abbreviation
$p^{ex} = 0.015$, $\tau = 0.003$, $q = 500$, $w = 1000$	BMAB-SPAR-NS
$p^{ex} = 0.005$, $\tau = 0.005$, $q = 500$, $w = 1000$	BMAB-SPAR-NS-A
$p^{ex} = 0.005$, $\tau = 0.005$, $q = 1000$, $w = 2000$	BMAB-SPAR-NS-B
$p^{ex} = 0.05$, $\tau = 0.001$, $q = 500$, $w = 1000$	BMAB-SPAR-NS-C
$p^{ex} = 0.05$, $\tau = 0.001$, $q = 1000$, $w = 2000$	BMAB-SPAR-NS-D
$p^{ex} = 0.05$, $\tau = 0.001$, $q = 500$, $w = 2000$	BMAB-SPAR-NS-E
$p^{ex} = 0.005$, $\tau = 0.001$, $q = 500$, $w = 2000$	BMAB-SPAR-NS-F
$p^{ex} = 0.05$, $\tau = 0.005$, $q = 500$, $w = 2000$	BMAB-SPAR-NS-G
$p^{ex} = 0.05$, $\tau = 0.005$, $q = 1000$, $w = 3000$	BMAB-SPAR-NS-H
$p^{ex} = 0.005$, $\tau = 0.001$, $q = 750$, $w = 4000$	BMAB-SPAR-NS-I
$p^{ex} = 0.005$, $\tau = 0.001$, $q = 1000$, $w = 4000$	BMAB-SPAR-NS-J
$p^{ex} = 0.005$, $\tau = 0.001$, $q = 500$, $w = 4000$	BMAB-SPAR-NS-K
$p^{ex} = 0.005$, $\tau = 0.001$, $q = 1500$, $w = 4000$	BMAB-SPAR-NS-L
$p^{ex} = 0.05$, $\tau = 0.005$, $q = 1500$, $w = 4000$	BMAB-SPAR-NS-M
$p^{ex} = 0.05$, $\tau = 0.005$, $q = 1000$, $w = 4000$	BMAB-SPAR-NS-N

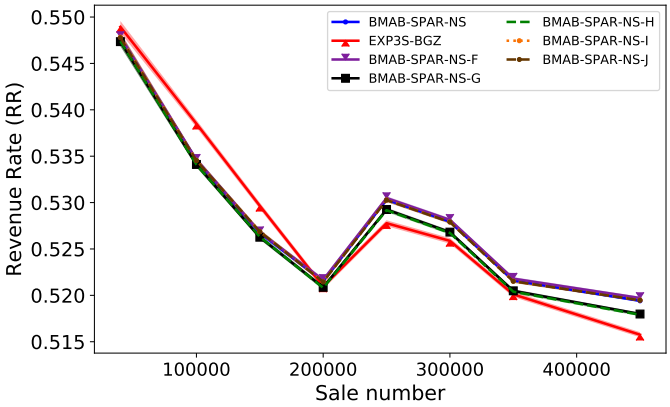


(a) Set A1.

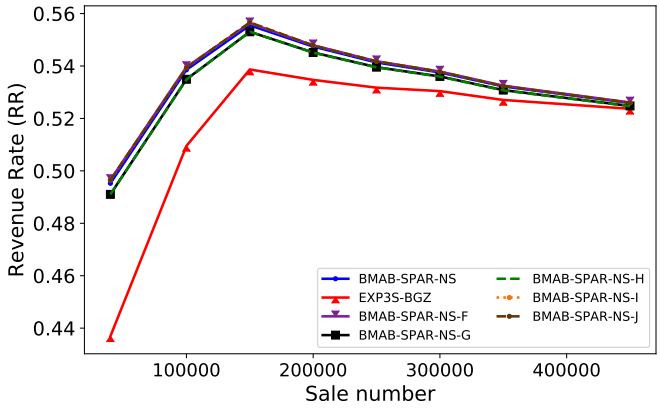


(b) Set A2.

Figure 3.13: Performance of the algorithms for website A, averaged over 200 runs. Lines indicate the mean and shaded region indicates 95% confidence interval.

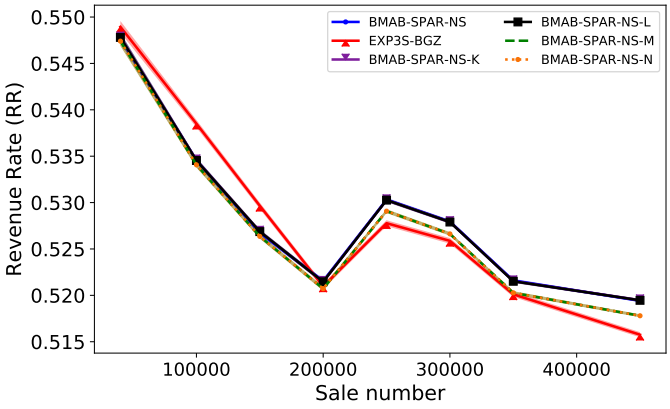


(a) Set A1.

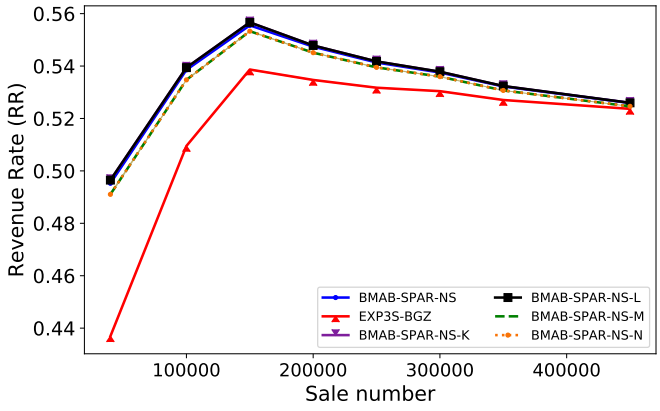


(b) Set A2.

Figure 3.14: Performance of the algorithms for website A, averaged over 200 runs. Lines indicate the mean and shaded region indicates 95% confidence interval.

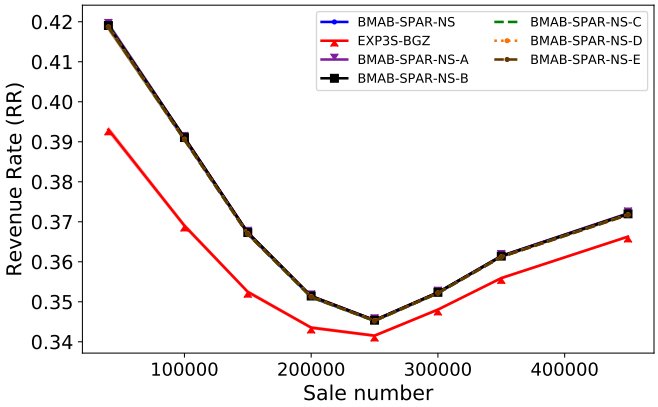


(a) Set A1.

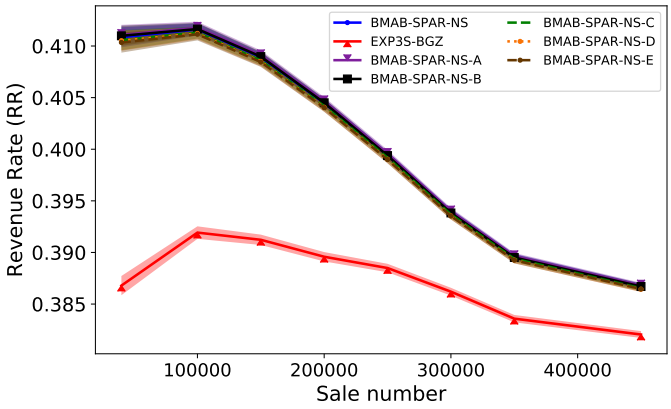


(b) Set A2.

Figure 3.15: Performance of the algorithms for website A, averaged over 200 runs. Lines indicate the mean and shaded region indicates 95% confidence interval.

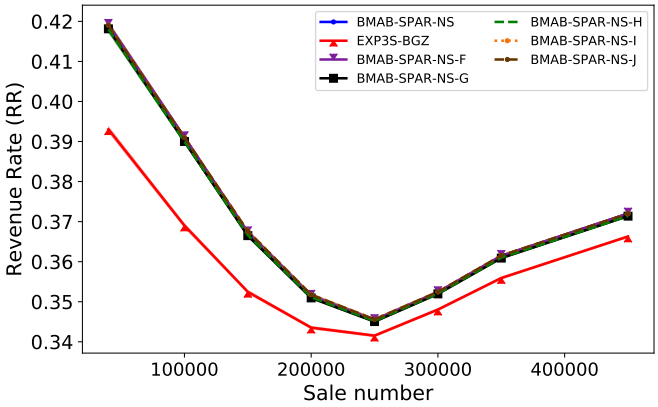


(a) Set B1.

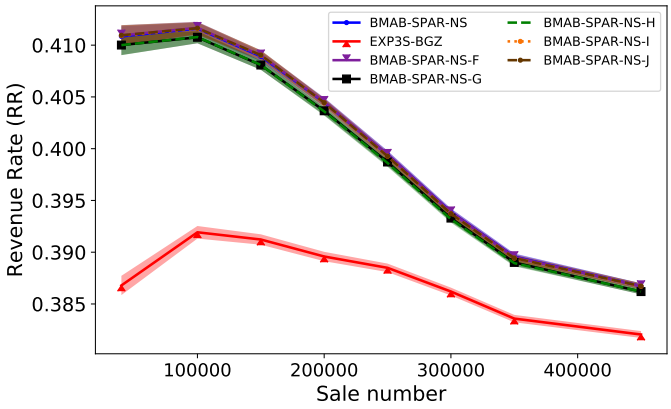


(b) Set B2.

Figure 3.16: Performance of the algorithms for website B, averaged over 200 runs. Lines indicate the mean and shaded region indicates 95% confidence interval.

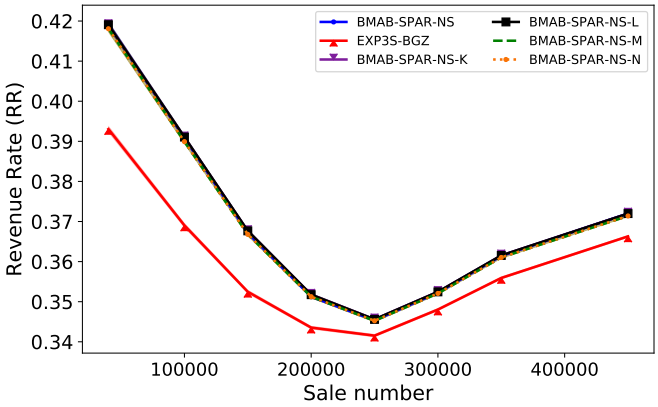


(a) Set B1.

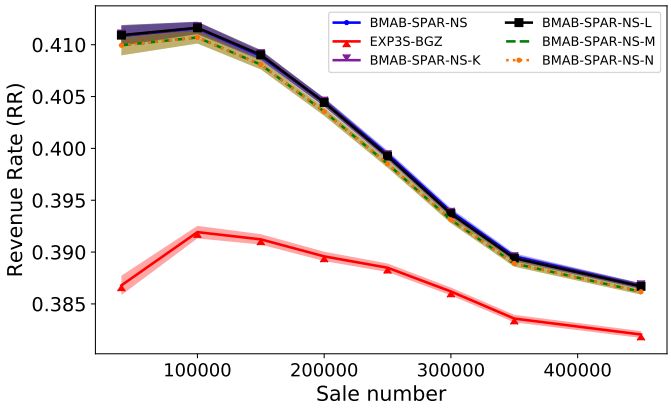


(b) Set B2.

Figure 3.17: Performance of the algorithms for website B, averaged over 200 runs. Lines indicate the mean and shaded region indicates 95% confidence interval.



(a) Set B1.



(b) Set B2.

Figure 3.18: Performance of the algorithms for website B, averaged over 200 runs. Lines indicate the mean and shaded region indicates 95% confidence interval.

Chapter 4

Slate bandits with non-separable reward functions*

This Chapter builds on Chapter 3 and studies reserve price optimization in the setting of header bidding. Note that, in the reserve price optimization problem of Chapter 3, the publisher received a single offer at a time for each impression. However, in this Chapter, we consider a publisher that receives multiple offers for a single impression and where each offer is the result of a second-price auction with a reserve price. The goal of the publisher is to learn the best reserve price for each second-price auction (i.e., a vector of reserve prices) in order to maximize his expected revenue.

---

*This chapter is based on Rhuggenaath et al. [139].

4.1 Introduction

In many practical problems an agent needs to choose an action from a set where each action leads to a random reward with the objective of maximizing expected cumulative rewards over a finite time horizon. Often the reward distribution is unknown, and as a consequence, the agent faces an exploration-exploitation trade-off. The multi-armed bandit problem [37] is a standard framework for studying such exploration-exploitation problems.

Many problems in the domain of web-services, such as e-commerce, online advertising and streaming, require the agent to select not only one but multiple actions at the same time. After the agent makes a choice, a collective reward characterizing the quality of the entire selection is observed. Problems of this type are typically referred to as slate bandits or combinatorial bandits [43, 59]. In a slate bandit problem, a slate consists of a number of slots and each slot has a number of base actions. Given a particular action for each slot, a reward function defined at the slate-level determines the reward for each slate.

One example of a slate bandit problem is when a seller can simultaneously access different markets in order to sell an item and accepts the best offer among the markets. The seller has to specify (and learn) the best price for each market and the revenue from a sale equals the maximum revenue over all markets. This can be interpreted as a slate bandit problem where the slots are the different markets and the base actions are the prices in each market. The goal is to select a set of prices such that the expected revenue is maximized. One example from the domain of online advertising which can be modeled as such a problem, is the reserve price optimization problem using header bidding (see Section 4.3.2 for more details).

Another example of a slate bandit problem, arises in the context of selection problems where fairness considerations are important. Suppose for example, that there are a number of distinct groups and that a decision-maker has to choose an option from a set for each group (each group has its own set of options) and where each option yields an uncertain reward for each group. Choosing the alternative with the highest expected reward for each group may not be desirable, since some groups might receive much higher expected rewards than others and which may be considered as unfair. One popular objective function that decision-maker can use that takes fairness into account is the max-min objective function, where each group receives a reward that equals the minimum reward over all groups. This selection problem faced by the decision-maker can be interpreted as a slate bandit problem where the slots are the different groups, the base actions are the options for each group, and the slate-level reward is the minimum reward over all groups.

Previous studies (see e.g. [70, 99, 132]) assume that the reward function at the slate level is additive or that the expected reward at the slate-level is a non-decreasing function of the expected rewards at the slot-level (this is also called the monotonicity assumption). This implies that the optimal action at the slate level can be found by

finding the optimal base action for each individual slot. Whether the monotonicity assumption holds or not, depends on the slate-level reward function and the slot-level distributions. In some applications the monotonicity assumption might be reasonable, but in some cases it might not hold (see Example 4.1 for more details). For example, if the slate-level reward is the maximum (or minimum) of the rewards at the slot level, then the monotonicity assumption does not hold in general. We refer to this as a *non-separable* slate-level reward function. The reserve price optimization problem mentioned above is thus a concrete example of a slate bandit problem with a non-separable reward function. The selection problem with fairness mentioned above is another example where non-separable reward functions arise. With the max-min objective function, the slate-level reward equals the minimum of the rewards at the slot level and the goal is to maximize the expected value. However, there are many more possible aggregation functions (many of which are non-linear) that can be used in order to construct the slate-level reward (see e.g. [25]), and these functions generally do not satisfy the monotonicity assumption.

In this Chapter we study slate bandits with non-separable reward functions. To the best of our knowledge, this variant of the slate bandit problem has not been studied before and existing algorithms either cannot be applied or do not have performance guarantees for our problem. We are mainly concerned with cases where the number of slates is large relative to the time horizon, so that trying each slate as a separate arm in a traditional multi-armed bandit, would not be efficient. In such cases it is not immediately clear whether sub-linear regret is possible, and therefore we study the design of algorithms that have sub-linear regret. We summarize the main contributions of this Chapter as follows:

- To the best of our knowledge, we are the first to study slate bandits with non-separable reward functions.
- We provide a theoretical analysis and derive problem-dependent and problem-independent regret bounds. We provide algorithms that have sub-linear regret with respect to the time horizon.
- Experimental results on simulated data and using real-world data show that our proposed method outperforms popular benchmark bandit algorithms.

The remainder of this Chapter is organized as follows. In Section 4.2 we discuss the related literature. Section 4.3 provides a formal description of the problem. In Section 4.4 we present the our proposed algorithms for the slate bandit problem and provide a theoretical analysis. In Section 4.5 we perform experiments and compare our method with baseline strategies in order to assess the quality of our proposed algorithms. Section 4.6 concludes our work and provides some interesting directions for further research.

4.2 Related Literature

The slate bandit problem has been studied before in multiple prior papers and these papers study different variants of the problem and make different assumptions. The main variants of the slate bandit problem center around three properties of the problem: (i) whether the slot-level rewards in the slate are observed or not (the situation where the slot-level rewards are observed is often referred to as semi-bandit feedback in the literature); (ii) whether the function that determines the slate-level reward is known or not; (iii) the structural properties of the function that determines the slate-level reward.

In [43, 52, 59, 99, 110, 112, 132, 164, 165] slate bandits with semi-bandit feedback are studied. In [99, 110, 152, 165] it is assumed that the slate-level reward is an additive function of the rewards of the individual slots. In [152] the slot-level rewards are assumed to be unobserved, while [99] assumes that slate-level reward function is known. Some papers make other structural assumptions about the slate-level reward function. In [52, 53, 112, 132] two key structural assumptions are made: a monotonicity assumption and a bounded smoothness (or Lipschitz continuity) assumption. In addition, [52, 112, 132] do not assume that the slate-level reward function is known. Instead, they assume that an α-approximation oracle is available.

In related work [70] do not assume that the slot-level rewards are observed and that the slate-level reward function is known. They exploit a monotonicity assumption (similar to [52, 53, 112, 132]) that relates the slot-level rewards to the slate-level rewards and propose a heuristic based on Thompson sampling in order to balance exploration and exploitation. However, they do not provide performance guarantees for their algorithms.

Similar to previous works, we also consider a setting with semi-bandit feedback. The main difference between the work in this Chapter and the aforementioned works, is that we do not assume that the slate-level reward is additive or that the expected reward at the slate-level is a non-decreasing function of the expected rewards at the slot-level. However, unlike in [70], we assume that the slate-level reward function is known. Furthermore, we do not make use of approximation oracles as in [52, 53, 112, 132].

To the best of our knowledge, this variant of the slate bandit problem has not been studied before and existing algorithms either cannot be applied or do not have performance guarantees for our problem.

4.3 Problem formulation

4.3.1 Problem definition and notation

We consider a slate bandit problem that is similar to [70]. The set of actions (the slates) is given by $\mathcal{B}$ with $|\mathcal{B}| = \bar{K}$. If action $b \in \mathcal{B}$ is selected, then the reward is a

random variable $Y(b)$. A slate consists of $M \in \mathbb{N}$ slots, where $M > 1$. Each action is a vector in $\mathbb{R}^M$. That is, $b \in \mathbb{R}^M$ for all $b \in \mathcal{B}$. Slot $i \in \{1, \dots, M\}$ has a set of base actions $\mathcal{B}_i$ with $|\mathcal{B}_i| = K_i$. The set of slates $\mathcal{B}$ is given by $\mathcal{B} = \mathcal{B}_1 \times \mathcal{B}_2 \times \cdots \times \mathcal{B}_M$. We make the following assumptions regarding the slot-level action sets.

Assumption 4.1. *Without loss of generality we assume that* $\mathcal{B}_i = \{1, \dots, K_i\}$ *for* $i = 1, \dots, M$ *and* $|\mathcal{B}_i| = K_i = K$ *for* $i = 1, \dots, M$.

Given an action $b \in \mathcal{B}$ the random variable $Y(b)$ satisfies $Y(b) = f(Y_1(b_1), \dots, Y_M(b_M))$, where $b_i \in \mathbb{R}$ is the i-th element of action b and where $Y_i(b_i)$ for $i = 1, \dots, M$ is a random variable. We make the following assumptions regarding the slate-level reward function.

Assumption 4.2. *Let* $b \in \mathcal{B}$ *and* $Y(b) = f(Y_1(b_1), \dots, Y_M(b_M))$. *Then,* $Y_i(b_i)$ *is independent from* $Y_j(b_j)$ *for all* $j \neq i$.

Assumption 4.3. *The function* f *is known and satisfies* $f : \mathbb{R}^M \to [0, 1]$.

For $b \in \mathcal{B}$ define the quantity $\mu(b) = \mathbb{E}\{Y(b)\}$ and let $b^* = \operatorname{argmax}_{b \in \mathcal{B}} \mu(b)$. The optimality gap for action $b \in \mathcal{B}$ is defined as $\Delta(b) = \mu(b^*) - \mu(b)$. Define $\Delta_{\min} = \min\{\Delta(b) | b \in \mathcal{B}, b \neq b^*\}$. Here $\Delta_{\min}$ measures the optimality gap between the best action and the second-best action. We assume that the optimality gaps satisfy $\Delta_{\min} \geq \varepsilon > 0$ for some $\varepsilon \in \mathbb{R}$. This assumption enforces that the optimality gap is bounded from below and ensures that the notion of 'the best action' and 'the second-best action' is well-defined.

We assume that decisions need to be made for $T \geq 2$ rounds. We assume that the decisions are implemented according to the following online protocol: for each round $t \in \{1, \dots, T\}$

1. the agent selects a slate $b \in \mathcal{B}$.
2. for each $i \in \{1, \dots, M\}$, the agent observes an i.i.d. (independent over rounds) realization r^i from the distribution of $Y_i(b_i)$. The agent receives r_t where $r_t = f(r^1, \dots, r^M)$. That is, $r_t \sim Y(b)$ where $Y(b) = f(Y_1(b_1), \dots, Y_M(b_M))$. The rewards r_t are independent over the rounds.

For a fixed sequence $i_1, \dots, i_T$ of selected actions, the pseudo-regret over T rounds is defined as $R_T = T \cdot \mu^* - \sum_{t=1}^{T} \mu(i_t)$. The expected pseudo-regret is defined as $\mathcal{R}_T = \mathbb{E}\{R_T\}$, where the expectation is taken with respect to possible randomization in the selection of the actions $i_1, \dots, i_T$.

The slate bandit problem is challenging due to the number of actions growing exponentially in M, and due to the non-separable reward function which implies that $\mathbb{E}\{Y(b)\}$ cannot necessarily be maximized by choosing the action with the highest expected reward at the slot-level for each slot individually. Note that we allow for an arbitrary function f in Assumption 4.3 and that the reward distributions at the slot-level can also be arbitrary (as long as they are mapped to values in $[0, 1]$). Existing

papers [51, 52, 53, 70, 112, 132] assume that f is an additive function or that $\mathbb{E}\{Y(b)\}$ satisfies a monotonicity property. In Example 4.1 below we give a concrete example that shows that this monotonicity property does not hold in all instances of the slate bandit problem. The example shows that, if an instance of the slate bandit problem is not compatible with the monotonicity assumption, the optimal action may not always be correctly identified by an algorithm that exploits the monotonicity assumption. As a consequence, existing algorithms that rely on the monotonicity assumption may fail to identify the optimal action, and are therefore in general not guaranteed to solve our problem. Furthermore, Example 4.1 also shows that the performance guarantees of algorithms that rely on the monotonicity assumption do not hold for all instances of the slate bandit problem. The example illustrates that, whether the monotonicity assumption holds or not, depends on the slate-level reward function and the slot-level distributions. As the reward distributions are unknown, it is valuable to have algorithms (like the one proposed in this Chapter) with performance guarantees that do not depend on prior knowledge about the reward distributions and do not rely on the monotonicity assumption. Assumption 4.2 is a common assumption (see e.g., [51, 163]) and may seem restrictive, but even under this assumption, this problem is still non-trivial and, to the best of our knowledge, existing algorithms are not able to solve this problem. Example 4.1 shows that, even under Assumption 4.2, existing algorithms can fail to learn the best slate.

Example 4.1. *Consider a simple instance of the slate bandit problem where there are $M = 2$ slots. Let $\mathcal{B}_1 = \{a, b\}$, $\mathcal{B}_2 = \{c, d\}$. Let $Y_1(a) \sim U(0.4, 0.5)$, $Y_1(b) \sim U(0.0, 0.1)$, $Y_2(c) \sim U(0.4, 0.5)$, $Y_2(d) \sim U(0.15, 0.7)$. Here $U(v, w)$ denotes a uniform distribution on $[v, w]$. For each slot, there are 2 actions. There are 4 slates in total and the slates are given by $\mathcal{B} = \{\{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}\}$.*
The rewards at the slate level are given by:

$$Y(\{a,c\}) = \max\{Y_1(a), Y_2(c)\}, \qquad Y(\{a,d\}) = \max\{Y_1(a), Y_2(d)\},$$
$$Y(\{b,c\}) = \max\{Y_1(b), Y_2(c)\}, \qquad Y(\{b,d\}) = \max\{Y_1(b), Y_2(d)\}.$$

Let $\mu_a = \mathbb{E}\{Y_1(a)\}$, $\mu_b = \mathbb{E}\{Y_1(b)\}$, $\mu_c = \mathbb{E}\{Y_2(c)\}$ and $\mu_d = \mathbb{E}\{Y_2(d)\}$.
Existing algorithms [51, 52, 53, 70, 112, 132] make a monotonicity assumption. This assumption states that if the vector of mean rewards of the slots in a slate (say slate A) dominates the vector of mean rewards of the slots in another slate (say slate B), then the expected reward of slate A is at least as high as the expected reward of slate B. A vector $W \in \mathbb{R}^n$ dominates a vector $Z \in \mathbb{R}^n$ if, for all $i = 1, \ldots, n$, the i-th component of W is at least as large as the i-th component of Z. In this example the monotonicity assumption implies that, if $\mu_c \geq \mu_d$, then it must be that $\mathbb{E}\{Y(\{a,d\})\} \leq \mathbb{E}\{Y(\{a,c\})\}$.
Note that from the properties of the uniform distribution we have that $\mathbb{E}\{Y_1(a)\} = \mathbb{E}\{Y_2(c)\} > \mathbb{E}\{Y_2(d)\}$. Therefore, the monotonicity assumption implies that we should have $\mathbb{E}\{Y(\{a,d\})\} \leq \mathbb{E}\{Y(\{a,c\})\}$. However, it can be shown that

$\mathbb{E}\{Y(\{a,d\})\} > \mathbb{E}\{Y(\{a,c\})\}$ *in this example (a proof can be found in the Appendix). Therefore, the monotonicity assumption implies that slate* $\{a,c\}$ *has an expected reward that is at least as high as the expected reward of slate* $\{a,d\}$ *and this implication is false. Note that the vector of expected rewards of slate* $\{a,c\}$ *dominates the vector of expected rewards of all other slates. Therefore, under the monotonicity assumption, the slate* $\{a,c\}$ *is actually the optimal slate and thus the best action. In this example, the optimal action is thus not correctly identified. Existing algorithms that rely on the monotonicity assumption are therefore not guaranteed to learn the best action in this slate bandit problem.* □

4.3.2 Example application: reserve price optimization and header bidding

One of the main mechanisms that web publishers use in online advertising in order to sell their advertisement space is the real-time bidding (RTB) mechanism [162]. In RTB there are three main platforms: supply side platforms (SSPs), demand side platforms (DSPs) and an ad exchange (ADX) which connects SSPs and DSPs. The SSPs collect inventory of different publishers and thus serve the supply side of the market. Advertisers which are interested in showing online advertisements are connected to DSPs. A real-time auction decides which advertiser is allowed to display its ad and the amount that the advertiser needs to pay. Most of the ad inventory is sold via second-price auctions with a reserve price [125, 146, 162]. In this auction, the publisher specifies a value p_t (the reserve price) which represent the minimum price that he wants for the impression. The revenue for the publisher (at a particular reserve price) is random and depends on the highest bid (X_t) and second highest bid (W_t) in the auction. The revenue of the publisher in round t is given by $R_t(p_t) = \mathbb{I}\{p_t \leq X_t\} \cdot \max\{W_t, p_t\}$.

In header bidding (see e.g. [97]), the publisher can simultaneously connect to multiple header bidding partners (these are SSPs and ad exchanges but for simplicity we call each partner an SSP) for a single impression. The publisher specifies a reserve price for each SSP. Each SSP is involved in a separate auction and reports a value (a bid) indicating the revenue for the publisher. The publisher observes the individual revenues and subsequently chooses a winner among the SSPs (the SSP with the highest revenue wins). The slate bandit problem studied in this Chapter can be used to model a reserve price optimization problem with header bidding. The connection is as follows. There are M SSPs and in every round t the publisher needs to choose a vector of reserve prices from the set $\mathcal{B}$. The revenue from header bidding when action $b \in \mathcal{B}$ chosen is given by $Y(b) = f(Y_1(b_1), \ldots, Y_M(b_M)) = \max\{Y_1(b_1), \ldots, Y_M(b_M)\}$. Note that Assumption 4.2 is reasonable in this setting since (i) the pool of advertisers and their bidding strategies can differ across DSPs, (ii) advertisers do not observe the bids (of their competitors) on other DSPs, and (iii) SSPs can be connected to different DSPs.

4.4 Algorithm and Analysis

4.4.1 The ETC-SLATE algorithm

In this section we discuss our proposed algorithm. We refer to our algorithm as ETC-SLATE (Explore then Commit slate bandit algorithm). The main idea that is used in our proposed algorithm relies on exploiting Assumption 4.2. This is best illustrated using an example.

Example 4.2. *Consider a simple instance of the slate bandit problem where there are $M = 3$ slots. Assume that $\mathcal{B}_1 = \{x_1, x_2\}$, $\mathcal{B}_2 = \{y_1, y_2\}$, $\mathcal{B}_3 = \{z_1, z_2\}$. Therefore, we have that*

$$\begin{aligned}\mathcal{B} = \{&(x_1, y_1, z_1), (x_1, y_1, z_2),\\ &(x_1, y_2, z_1), (x_1, y_2, z_2),\\ &(x_2, y_1, z_1), (x_2, y_1, z_2),\\ &(x_2, y_2, z_1), (x_2, y_2, z_2)\}.\end{aligned}$$

Suppose that, for every $b \in \mathcal{B}$, we want to have N i.i.d. (independent and identically distributed) samples from the distribution of $Y(b) = f(Y_1(b_1), \ldots, Y_3(b_3))$ where $b_i \in \mathcal{B}_i$. The straightforward way to do this is to collect N i.i.d. samples from the distribution of $Y(b)$ by selecting every action $b \in \mathcal{B}$ exactly N times. Thus you would need $N \cdot |\mathcal{B}|$ samples in total.

A more efficient approach is simply to sample action (x_1, y_1, z_1) and action (x_2, y_2, z_2) exactly N times and save the values of $Y_1(x_1)$, $Y_1(x_2)$, $Y_2(y_1)$, $Y_2(y_2)$, $Y_3(z_1)$, $Y_3(z_2)$. By Assumption 4.2, we can use these samples to obtain N i.i.d. samples from the distribution of $Y(b)$ for all $b \in \mathcal{B}$. To get an i.i.d. sample from the distribution of $Y((x_2, y_1, z_2)) = f(Y_1(x_2), Y_2(y_1), Y_3(z_2))$, we simply use a sample from the distribution of $Y_1(x_2)$, $Y_2(y_1)$ and $Y_3(z_2)$. Note that this approach only requires $N \cdot 2$ samples in total and this is less than the $N \cdot |\mathcal{B}|$ samples of the previous approach. Note in particular that this approach allows us to obtain samples for actions $b \in \mathcal{B}$ that have not been selected. In our example above, action (x_2, y_1, z_2) was not selected. However, by selecting action (x_1, y_1, z_1) and action (x_2, y_2, z_2) we do obtain the necessary information that allows us to construct an artificial i.i.d. sample from the distribution of $Y((x_2, y_1, z_2))$. □

The pseudo-code for ETC-SLATE is given by Algorithm 4.1. The main idea is to divide the horizon T into two phases. The first phase (the exploration phase) has length $N = \hat{N}K$ and the second phase (the commit phase) has length $T - N$. In the first phase, the algorithm determines the best action $\hat{b}$ in action set $\mathcal{B}$. In the second phase, the algorithm commits to using action $\hat{b}$ in each round.

In the first phase, the algorithm takes a subset $\mathcal{B}^F = \{\cup_{l=1}^{K}(l, \ldots, l) | (l, \ldots, l) \in \mathcal{B}\}$ of actions from the set $\mathcal{B}$ and selects each action in this subset $\hat{N}$ times. Each time

that action $b \in \mathcal{B}^F$ is selected, the rewards of the slots are observed (Line 6) and stored for later use (Line 7). In Lines 11-16, the stored rewards for the slots are used in order to generate $\hat{N}$ i.i.d. samples from the distribution of the random variable $Y(b)$ which are given by $\hat{Y}^1(b), \ldots, \hat{Y}^{\hat{N}}(b)$. In Line 17, the empirical mean of the $\hat{N}$ values $\hat{Y}^1(b), \ldots, \hat{Y}^{\hat{N}}(b)$ is determined for each action $b \in \mathcal{B}$. The action $\hat{b}$ is then chosen as the action $b \in \mathcal{B}$ with the highest empirical mean. The value of $\hat{N}$ is determined by the following parameters: the horizon T, κ, γ, and action set $\mathcal{B}$. In Section 4.4.2 and 4.4.3 we will show that this choice for $\hat{N}$ leads to sub-linear regret for suitably chosen values of κ and γ.

Algorithm 4.1 ETC-SLATE

Require: horizon T, κ, γ, action sets $\mathcal{B}$.

1: Set $\hat{N} = \left\lceil \frac{2}{\kappa^2} \cdot (\log(|\mathcal{B}|) - \log(\gamma)) \right\rceil$. Set $t = 1$.
2: Set $\mathcal{V}_{i,j} = \varnothing \ \forall \ i \in \{1, \ldots, M\}$ and $j \in \mathcal{B}_i$.
Explore Phase.
3: **for** $l \in \{1, \ldots, K\}$ **do**
4: **for** $n \in \{1, \ldots, \hat{N}\}$ **do**
5: Select action $(l, \ldots, l) \in \mathcal{B}$.
6: Observe rewards $z_{l,i,n} \sim Y_i(l)$ for $i = 1, \ldots, M$.
7: Set $\mathcal{V}_{i,l} = \mathcal{V}_{i,l} \cup \{z_{l,i,n}\}$ for $i = 1, \ldots, M$.
8: Set $t = t + 1$.
9: **end for**
10: **end for**
11: **for** $b = (l^1, \ldots, l^M) \in \mathcal{B}$ **do**
12: **for** $n \in \{1, \ldots, \hat{N}\}$ **do**
13: Select $z_{l^i,i,n} \in \mathcal{V}_{i,l^i}$ for $i = 1, \ldots, M$.
14: Set $\hat{Y}^n(b) = f(z_{l^1,1,n}, \ldots, z_{l^M,M,n})$.
15: **end for**
16: **end for**
Find best arm in $\mathcal{B}$.
17: Find $\hat{b} \in \mathcal{B}$ such that $\sum_{n=1}^{\hat{N}} \hat{Y}^n(\hat{b}) \frac{1}{\hat{N}} \geq \sum_{n=1}^{\hat{N}} \hat{Y}^n(b) \frac{1}{\hat{N}}$ for all $b \neq \hat{b}$.
Commit Phase.
18: **for** $t \in \{\hat{N}K + 1, \ldots, T\}$ **do**
19: Play action $\hat{b}$.
20: **end for**

4.4.2 Problem-dependent regret bounds

Lemma 4.1 ([88])**.** *Let $X_1, \ldots, X_n$ be independent random variables such that $X_i \in [a, b]$ for $i = 1, \ldots, n$. Let $\bar{X} = \frac{1}{n} \cdot \sum_{i=1}^{n} X_i$, and let $\epsilon \geq 0$. Then, $\mathbb{P}\left\{\bar{X} - \mathbb{E}\left\{\bar{X}\right\} \geq \epsilon\right\} \leq \exp\left\{\frac{-2\epsilon^2 n^2}{n(b-a)^2}\right\}$.*

Proposition 4.1. *Let $Y^1(b), \ldots, Y^n(b)$ be n i.i.d. draws from the distribution of $Y(b)$ for an action $b \in \mathcal{B}$. Assume that $Y^j(b)$ and $Y^k(l)$ are independent if $b \neq l$*

and $j \neq k$. Let $\bar{\mu}(b^*) = \sum_{i=1}^{n} Y^i(b^*) \cdot \frac{1}{n}$ and $\bar{\mu}(b) = \sum_{i=1}^{n} Y^i(b) \cdot \frac{1}{n}$ for $b \neq b^*$. Let $\hat{b} = \text{argmax}_{b \in \mathcal{B}} \bar{\mu}(b)$, where ties are broken arbitrarily if there are multiple candidates for $\hat{b}$. Then, $\mathbb{P}\left\{\hat{b} \neq b^*\right\} \leq \bar{K} \exp\left\{-\frac{1}{2} n (\Delta_{\min})^2\right\}$.

Proof. Define $\mathcal{B}^- = \{b \in \mathcal{B} | \bar{\mu}(b) \geq \bar{\mu}(b^*), b \neq b^*\}$. Then we have that,

$$\mathbb{P}\left\{\hat{b} \neq b^*\right\} = \mathbb{P}\left\{\hat{b} \in \mathcal{B}^-\right\} \overset{(a)}{\leq} \sum_{b \in \mathcal{B}^-} \mathbb{P}\left\{\hat{b} = b\right\} \overset{(b)}{\leq} \sum_{b \in \mathcal{B}^-} \mathbb{P}\left\{\bar{\mu}(b^*) \leq \bar{\mu}(b)\right\}$$
$$\overset{(c)}{\leq} \sum_{b \in \mathcal{B}^-} \exp\left\{-\frac{n}{2} (\Delta(b))^2\right\} \overset{(d)}{\leq} \bar{K} \exp\left\{-\frac{n}{2} (\Delta_{\min})^2\right\}.$$

Inequality (a) follows from applying a union bound over the set $\mathcal{B}^-$. Inequality (b) follows from the fact that $\mathbb{I}\left\{\hat{b} = b\right\} = 1 \Rightarrow \mathbb{I}\left\{\bar{\mu}(b^*) \leq \bar{\mu}(b)\right\} = 1$. Inequality (c) follows from applying Lemma 4.1 to the differences $Y^i(b^*) - Y^i(b)$ for $i = 1, \ldots, n$. Inequality (d) follows from the fact that $|\mathcal{B}^-| \leq \bar{K} = |\mathcal{B}|$ and $(\Delta_{\min})^2 \leq (\Delta(b))^2$ for $b \in \mathcal{B}$. This completes the proof. □

Recall that $\hat{b}$ denotes the action in $\mathcal{B}$ that is identified as b^* by Algorithm 4.1. The following proposition bounds the probability that b^* is incorrectly identified.

Proposition 4.2. *Let $m > 0$. Let $\hat{b}$ denote the action in $\mathcal{B}$ that is identified as b^* by Algorithm 4.1. If Algorithm 4.1 is run with the inputs: T, $\kappa = \Delta_{\min}$, $\gamma = \frac{1}{T^m}$, and action set $\mathcal{B}$, then $\mathbb{P}\left\{\hat{b} \neq b^*\right\} \leq \gamma$.*

Proof. From the description of Algorithm 4.1, it follows that $\hat{N} = 2(\Delta_{\min})^{-2} \cdot (\log(\bar{K}) - \log(\gamma))$. Given this choice of $\hat{N}$, we are able to generate $\hat{N}$ i.i.d. draws from the distribution of $Y(b)$ for each $b \in \mathcal{B}$. Let $\hat{b}$ denote the action that has the highest empirical mean based on the $\hat{N}$ samples and recall that b^* is the action with the highest expected return. By Proposition 4.1 it follows that $\mathbb{P}\left\{\hat{b} \neq b^*\right\} \leq \bar{K} \exp\left\{-\frac{\hat{N}}{2} (\Delta_{\min})^2\right\} = \frac{1}{T^m} = \gamma$. This completes the proof. □

We can now state the main result of this subsection (Theorem 4.1).

Theorem 4.1. *Let $m > 0$. If Algorithm 4.1 is run with inputs: T, $\kappa = \Delta_{\min}$, $\gamma = \frac{1}{T^m}$, and action set $\mathcal{B}$, then $\mathcal{R}_T \leq \frac{2K}{(\Delta_{\min})^2} \cdot (\log(\bar{K}) + m \log(T)) + T^{1-m}$ with $\bar{K} = |\mathcal{B}|$.*

Proof. Note that the regret $\mathcal{R}_T$ can be decomposed as $\mathcal{R}_T = \mathcal{R}_N + \mathcal{R}_{T-N}$. Here $\mathcal{R}_N$ denotes the regret over the first N rounds and $\mathcal{R}_{T-N}$ denotes the regret over the last $T - N$ rounds. In order to bound $\mathcal{R}_T$ it suffices to bound each term.

Note that $\mathcal{R}_N$ is trivially bounded by $N \cdot 1$ since by Assumption 4.3 the regret for any period is at most 1. From the description of Algorithm 4.1, it follows

that $\hat{N} = \frac{2}{(\Delta_{\min})^2} \cdot (\log(\bar{K}) - \log(\gamma))$. Given $\hat{N}$, it follows that Phase I has length $N = K \cdot \hat{N}$. By substituting the quantities for γ and κ, we conclude that $\mathcal{R}_N \leq \frac{2K}{(\Delta_{\min})^2} \cdot (\log(\bar{K}) + m\log(T))$.

We decompose $\mathcal{R}_{T-N}$ according to two cases:

(i) $\hat{b} \neq b^*$. If case (i) occurs, then $\mathcal{R}_{T-N}$ is trivially bounded by $(T-N) \cdot 1$. Therefore we conclude that in case (i) $\mathcal{R}_{T-N} \leq T - N$.

(ii) $\hat{b} = b^*$. If case (ii) occurs, then $\mathcal{R}_{T-N} = 0$. This follows from the fact that, $\Delta(b^*) = 0$.

By combining the results for the two cases above and noting that by Proposition 4.2 we have $\mathbb{P}\left\{\hat{b} \neq b^*\right\} \leq \gamma$, we obtain

$$\mathcal{R}_{T-N} \leq (T-N) \cdot \mathbb{P}\left\{\hat{b} \neq b^*\right\} + \mathbb{P}\left\{\hat{b} = b^*\right\} \cdot 0 \leq (T-N) \cdot \frac{1}{T^m} \leq T \cdot \frac{1}{T^m} = T^{1-m}.$$

This completes the proof. □

Corollary 4.1. *Let $\bar{K} = |\mathcal{B}| \leq T$ and $m = 1$. Suppose that Algorithm 4.1 is run with inputs: T, $\kappa = \Delta_{\min}$, $\gamma = \frac{1}{T^m}$, and action set $\mathcal{B}$. Then, $\mathcal{R}_T \leq \frac{2K}{(\Delta_{\min})^2} \cdot (2\log(T)) + 1$.*

If $\Delta_{\min}$ is not precisely known, we can still run Algorithm 4.1 using a lower bound for $\Delta_{\min}$ if this is available. In Theorem 4.1, the dependence of regret on T would then still be logarithmic in T but with a different problem-dependent constant.

4.4.3 Problem-independent regret bounds

The results of the previous section show that expected regret of order $O(\log T)$ is possible if the gaps are known. However, as $\Delta_{\min} \to 0$, the regret bounds in Theorem 4.1 and Corollary 4.1 becomes vacuous. Therefore, it is useful to study whether sub-linear regret is possible when the gaps are unknown. In this section we prove problem-independent regret bounds and show that sub-linear regret is still achievable.

Theorem 4.2. *Let $m > 0$. If Algorithm 4.1 is run with inputs: T, $\kappa = T^{-1/3}\sqrt{K}\sqrt{\log(T)}$, $\gamma = \frac{1}{T^m}$, and action set $\mathcal{B}$, then $\mathcal{R}_T \leq \frac{2T^{2/3}}{\log(T)} \cdot (\log(\bar{K}) + m\log(T)) + T^{1-m} + T^{2/3}\sqrt{K}\sqrt{\log(T)}$ with $\bar{K} = |\mathcal{B}|$.*

Proof. The proof uses similar arguments as in Proposition 4.2 and Theorem 4.1. Define the set $\mathcal{B}^H = \{b \in \mathcal{B} | \Delta(b) \geq \kappa\}$. Let $\hat{b}$ denote the action in $\mathcal{B}$ that is identified as b^* by Algorithm 4.1. Using similar arguments as in the proof of Proposition 4.2, we conclude that $\mathbb{P}\left\{\hat{b} \in \mathcal{B}^H\right\} \leq \gamma$. We again decompose the regret $\mathcal{R}_T$ as $\mathcal{R}_T = \mathcal{R}_N + \mathcal{R}_{T-N}$.

Note that $\mathcal{R}_N$ is trivially bounded by $N \cdot 1$ since by Assumption 4.3 the regret for any period is at most 1. From the description of Algorithm 4.1, it follows that $\hat{N} = 2\kappa^{-2} \cdot (\log(\bar{K}) - \log(\gamma))$. Given $\hat{N}$, it follows that Phase I has length $N = K \cdot \hat{N}$. By substituting the quantities for γ and κ, we conclude that $\mathcal{R}_N \leq \frac{2T^{2/3}}{\log(T)} \cdot (\log(\bar{K}) + m\log(T))$.

We decompose $\mathcal{R}_{T-N}$ according to two cases:

(i) $\hat{b} \in \mathcal{B}^H$. If case (i) occurs, then $\mathcal{R}_{T-N}$ is trivially bounded by $(T-N) \cdot 1$. Therefore, we conclude that in case (i) $\mathcal{R}_{T-N} \leq T - N$.

(ii) $\hat{b} \notin \mathcal{B}^H$. If $\hat{b} \notin \mathcal{B}^H$, then from the definition of $\mathcal{B}^H$, it follows that $\Delta(\hat{b}) \leq \kappa$.

By combining the results for the two cases above and noting that $\mathbb{P}\left\{\hat{b} \in \mathcal{B}^H\right\} \leq \gamma$, we obtain

$$\begin{aligned}\mathcal{R}_{T-N} &\leq (T-N) \cdot \mathbb{P}\left\{\hat{b} \in \mathcal{B}^H\right\} + \mathbb{P}\left\{\hat{b} \notin \mathcal{B}^H\right\} \cdot (T-N)\kappa \\ &\leq T \cdot \frac{1}{T^m} + T\kappa = T^{1-m} + T\kappa \\ &\leq T^{1-m} + T \cdot T^{-1/3}\sqrt{K}\sqrt{\log(T)}.\end{aligned}$$

Putting everything together, we obtain

$$\mathcal{R}_T \leq \frac{2T^{2/3}}{\log(T)} \cdot (\log(\bar{K}) + m\log(T) + T^{1-m} + T^{2/3}\sqrt{K}\sqrt{\log(T)}.$$

This completes the proof. □

Corollary 4.2. *Let $\bar{K} = |\mathcal{B}| \leq T$ and $m = 1$. Suppose that Algorithm 4.1 is run with inputs: T, $\kappa = T^{-1/3}\sqrt{K}\sqrt{\log(T)}\sqrt{(1+m)}$, $\gamma = \frac{1}{T^m}$, and action set $\mathcal{B}$. Then, $\mathcal{R}_T \leq T^{2/3} \cdot (2 + \sqrt{2K\log(T)}) + 1$.*

It is useful to compare the obtained bounds with previously known results. If we consider every slate as a separate action in a standard multi-armed bandit algorithm such as UCB1, then regret of order $O(\sqrt{T\log(T)}\sqrt{K^M})$ is possible [37]. If we compare this with Corollary 4.2, then we have a worse dependence on T (we have $T^{2/3}\sqrt{\log(T)}$ instead of $\sqrt{T\log(T)}$) but a better dependence on K.

When $\bar{K}$ is large compared to T (the case we are interested in) UCB1 will perform very poorly. Also, UCB1 can only be applied if $\bar{K} \leq T$ since it needs to sample each of the slates are least once. In contrast, our algorithm can even be applied when $\bar{K} > T$ and it will still have $T^{2/3}$ regret (see Corollary 4.3 and 4.4), since it does not need sample each slate. Therefore, our algorithm provides a substantial improvement relative to what is possible based on the current state-of-the-art (e.g. can also be used when UCB1 cannot). It is an open problem whether the dependence on T can be improved further without needing to sample each slate at least once.

Corollary 4.3. *Let $\bar{K} = |\mathcal{B}| > T \geq 3$ and $m = 1$. Suppose that Algorithm 4.1 is run with inputs: T, $\kappa = T^{-1/3}\sqrt{K}\sqrt{\log(T)}\sqrt{(M+m)}$, $\gamma = \frac{1}{T^m}$, and action set $\mathcal{B}$. Suppose that $T \geq K2\kappa^{-2} \cdot (\log(\bar{K}) - \log(\gamma))$. Then, $\mathcal{R}_T \leq T^{2/3} \cdot (\sqrt{2} + 2\sqrt{M}\log(K) + \sqrt{(M+1)K\log(T)}) + 1$.*

Proof. The proof uses similar arguments as in Theorem 4.2. Define the set $\mathcal{B}^H = \{b \in \mathcal{B} | \Delta(b) \geq \kappa\}$. Let $\hat{b}$ denote the action in $\mathcal{B}$ that is identified as b^* by Algorithm 4.1. Using similar arguments as in the proof of Proposition 4.2, we conclude that $\mathbb{P}\left\{\hat{b} \in \mathcal{B}^H\right\} \leq \gamma$. We again decompose the regret $\mathcal{R}_T$ as $\mathcal{R}_T = \mathcal{R}_N + \mathcal{R}_{T-N}$.

Note that $\mathcal{R}_N$ is trivially bounded by $N \cdot 1$ since by Assumption 4.3 the regret for any period is at most 1. From the description of Algorithm 4.1, it follows that $\hat{N} = 2\kappa^{-2} \cdot (\log(\bar{K}) - \log(\gamma))$. Given $\hat{N}$, it follows that Phase I has length $N = K \cdot \hat{N}$. Note that by assumption we have $T \geq K2\kappa^{-2} \cdot (\log(\bar{K}) - \log(\gamma))$, so that $N \leq T$. Therefore, Phase I can be executed. By substituting the quantities for γ and κ, we conclude that

$$\begin{aligned} N = K \cdot \hat{N} &= \frac{K2T^{2/3}}{K\log(T)\sqrt{M+m}} \cdot (M\log(K) + m\log(T)) \\ &= \frac{K2T^{2/3}}{K\log(T)\sqrt{M+m}} \cdot M\log(K) + \frac{K2T^{2/3}}{K\log(T)\sqrt{M+m}} \cdot m\log(T) \\ &\overset{(a)}{\leq} 2T^{2/3}\sqrt{M}\log(K) + \frac{2T^{2/3}}{\sqrt{M+1}} \\ &\overset{(b)}{\leq} 2T^{2/3}\sqrt{M}\log(K) + \sqrt{2}T^{2/3}. \end{aligned}$$

Inequality (a) follows from the fact that $\log(T) \geq 1$ if $T \geq 3$, and by using that $m = 1$, and from the fact that $\frac{M}{\sqrt{M+1}} \leq \frac{M}{\sqrt{M}} = \sqrt{M}$. Inequality (b) follows from the fact that $M \geq 1$.
Therefore, we conclude that $\mathcal{R}_N \leq N \leq 2T^{2/3}\sqrt{M}\log(K) + \sqrt{2}T^{2/3}$.

We decompose $\mathcal{R}_{T-N}$ according to two cases:

(i) $\hat{b} \in \mathcal{B}^H$. If case (i) occurs, then $\mathcal{R}_{T-N}$ is trivially bounded by $(T-N) \cdot 1$. Therefore, we conclude that in case (i) $\mathcal{R}_{T-N} \leq T - N$.

(ii) $\hat{b} \notin \mathcal{B}^H$. If $\hat{b} \notin \mathcal{B}^H$, then from the definition of $\mathcal{B}^H$, it follows that $\Delta(\hat{b}) \leq \kappa$.

By combining the results for the two cases above and noting that $\mathbb{P}\left\{\hat{b} \in \mathcal{B}^H\right\} \leq \gamma$, we obtain

$$\begin{aligned}\mathcal{R}_{T-N} &\leq (T-N)\cdot \mathbb{P}\left\{\hat{b}\in\mathcal{B}^H\right\} + \mathbb{P}\left\{\hat{b}\notin\mathcal{B}^H\right\}\cdot(T-N)\kappa \\ &\leq T\cdot\frac{1}{T^m} + T\kappa \\ &\leq T^{1-m} + T\cdot T^{-1/3}\sqrt{K}\sqrt{\log(T)}\sqrt{(M+m)} \\ &\leq 1 + T^{2/3}\sqrt{K}\sqrt{\log(T)}\sqrt{(M+1)}.\end{aligned}$$

Putting everything together, we obtain

$$\mathcal{R}_T \leq T^{2/3}\cdot(\sqrt{2} + 2\sqrt{M}\log(K) + \sqrt{(M+1)K\log(T)}) + 1.$$

This completes the proof. □

Corollary 4.4. *Let $\bar{K} = \beta\cdot T$ with $\beta > 1$ and let $m = 1$. Suppose that Algorithm 4.1 is run with inputs: T, $\kappa = T^{-1/3}\sqrt{K}\sqrt{\log(T)}\sqrt{(1+m)}$, $\gamma = \frac{1}{T^m}$, and action set $\mathcal{B}$. Suppose that $T \geq K2\kappa^{-2}\cdot(\log(\bar{K}) - \log(\gamma))$.*
Then, $\mathcal{R}_T \leq T^{2/3}\cdot(2 + \log(\beta) + \sqrt{2K\log(T)}) + 1$.

Proof. The proof follows the same steps as in Corollary 4.3. Define the set $\mathcal{B}^H = \{b\in\mathcal{B}|\Delta(b)\geq\kappa\}$. Let $\hat{b}$ denote the action in $\mathcal{B}$ that is identified as b^* by Algorithm 4.1. Using similar arguments as in the proof of Proposition 4.2, we conclude that $\mathbb{P}\left\{\hat{b}\in\mathcal{B}^H\right\}\leq\gamma$. We again decompose the regret $\mathcal{R}_T$ as $\mathcal{R}_T = \mathcal{R}_N + \mathcal{R}_{T-N}$.

Note that $\mathcal{R}_N$ is trivially bounded by $N\cdot 1$ since by Assumption 4.3 the regret for any period is at most 1. From the description of Algorithm 4.1, it follows that $\hat{N} = 2\kappa^{-2}\cdot(\log(\bar{K}) - \log(\gamma))$. Given $\hat{N}$, it follows that Phase I has length $N = K\cdot\hat{N}$. Note that by assumption we have $T \geq K2\kappa^{-2}\cdot(\log(\bar{K}) - \log(\gamma))$, so that $N \leq T$. Therefore, Phase I can be executed. By substituting the quantities for γ and κ, we conclude that

$$\begin{aligned}N = K\cdot\hat{N} &= \frac{K2T^{2/3}}{K\log(T)(1+m)}\cdot(\log(\beta) + \log(T) + m\log(T)) \\ &= \frac{K2T^{2/3}}{K\log(T)(1+m)}\cdot\log(\beta) + \frac{2KT^{2/3}}{K\log(T)(1+m)}\cdot((1+m)\log(T)) \\ &\overset{(a)}{=} \frac{2T^{2/3}}{\log(T)(1+m)}\cdot\log(\beta) + 2T^{2/3} \\ &\overset{(b)}{\leq} \frac{2T^{2/3}}{(1+m)}\cdot\log(\beta) + 2T^{2/3} \\ &\overset{(c)}{\leq} T^{2/3}\log(\beta) + 2T^{2/3}.\end{aligned}$$

Inequality (b) follows from the fact that $T \geq N$ implies that $\log(T) \geq 1$. Too see

this, note that the assumption that $T \geq N$ implies that $T \geq 2T^{2/3}$ (this follows from equality (a)). Since $T \geq 2$, we have that $T \geq 2T^{2/3} > 2.72$. Therefore $\log(T) \geq 1$. Inequality (c) follows from the fact that $m = 1$.
Therefore, we conclude that $\mathcal{R}_N \leq N \leq T^{2/3}\log(\beta) + 2T^{2/3}$.

We decompose $\mathcal{R}_{T-N}$ according to two cases:

(i) $\hat{b} \in \mathcal{B}^H$. If case (i) occurs, then $\mathcal{R}_{T-N}$ is trivially bounded by $(T-N)\cdot 1$. Therefore, we conclude that in case (i) $\mathcal{R}_{T-N} \leq T - N$.

(ii) $\hat{b} \notin \mathcal{B}^H$. If $\hat{b} \notin \mathcal{B}^H$, then from the definition of $\mathcal{B}^H$, it follows that $\Delta(\hat{b}) \leq \kappa$.

By combining the results for the two cases above and noting that $\mathbb{P}\left\{\hat{b} \in \mathcal{B}^H\right\} \leq \gamma$, we obtain

$$\begin{aligned}
\mathcal{R}_{T-N} &\leq (T-N)\cdot \mathbb{P}\left\{\hat{b} \in \mathcal{B}^H\right\} + \mathbb{P}\left\{\hat{b} \notin \mathcal{B}^H\right\}\cdot (T-N)\kappa \\
&\leq T \cdot \frac{1}{T^m} + T\kappa \\
&\leq T^{1-m} + T\cdot T^{-1/3}\sqrt{K}\sqrt{\log(T)}\sqrt{(1+m)} \\
&\leq 1 + T^{2/3}\sqrt{K}\sqrt{\log(T)}\sqrt{2}.
\end{aligned}$$

Putting everything together, we obtain

$$\mathcal{R}_T \leq T^{2/3}\cdot(2 + \log(\beta) + \sqrt{2K\log(T)}) + 1.$$

This completes the proof. □

4.5 Experiments

In this section we conduct experiments in order to test the performance of our proposed algorithm. We conduct experiments using both simulated data and real-world data.

4.5.1 Experiments using simulated data

The main purposes of the experiments with simulated data are to verify the theoretical results that were derived, and to investigate the effects of ignoring the non-separability of the slate-level reward function on the regret.

Experimental settings

In the experiments we set $M = 5$ and $\mathcal{B}_i = \{1, \ldots, 10\}$ for $i = 1, \ldots, M$. Let $\max\{x, y\} = x \vee y$. We consider three choices for the slate-level reward function.

These choices are:
$f_1 = \frac{1}{4}(Y_1(b_1) \vee Y_2(b_2)) + \frac{1}{4}(Y_2(b_2) \vee Y_3(b_3)) + \frac{1}{4}(Y_3(b_3) \vee Y_4(b_4)) + \frac{1}{4}(Y_4(b_4) \vee Y_5(b_5))$
$f_2 = \frac{1}{4}(Y_1(b_1) \vee Y_2(b_2)) + \frac{1}{4}Y_3(b_3) + \frac{1}{4}Y_4(b_4) + \frac{1}{4}(Y_4(b_4) \vee Y_5(b_5))$
$f_3 = \frac{1}{4}(Y_1(b_1) \vee Y_2(b_2)) + \frac{1}{4}(Y_1(b_1) \vee Y_3(b_3)) + \frac{1}{4}(Y_1(b_1) \vee Y_4(b_4)) + \frac{1}{4}(Y_1(b_1) \vee Y_5(b_5))$.

In our experiments the rewards for $b \in \mathcal{B}_i$ follow a uniform distribution on $[a - c, a + c]$ where a is chosen uniformly from $[0.4, 0.6]$ independently for $i = 1, \ldots, M$ and for all $b \in \mathcal{B}_i$, and c is chosen uniformly from $[0.1, 0.3]$ independently from a. In total we have three experimental settings: Exp1, Exp2, Exp3. The abbreviation Exp1 means that f_1 is used. The other abbreviations have a similar interpretation.

The main motivation for the choice of slate-level reward functions and the reward distributions is that, the slate-level reward functions are non-separable, but since the reward distributions are uniform, the optimal slate and the regret can still be calculated analytically. In order to measure the performance of the methods, we also look at the per period reward, which is defined as $PPR(T) = \sum_{t=1}^{T} \hat{R}_t/T$. Here $\hat{R}_t$ is the observed reward in round t.

Benchmarks

To the best of our knowledge, there are no existing algorithms (with performance guarantees) for our slate bandit problem with non-separable rewards. For this reason we use the following two benchmarks. First, we run a standard multi-armed bandit algorithm on the base actions at the slot-level (for each slot independently), and we then combine the base actions chosen by these independent bandits in order to form the action at the slate-level. This is a reasonable benchmark, in the sense that assuming a non-decreasing reward function at the slate-level (i.e., a function that satisfies the monotonicity assumption in [52, 53, 70, 112, 132]), this should allow this benchmark to learn the optimal action over time. In the experiments we use the UCB1 [17] and Thompson sampling (TS) with Gaussian priors [8] as the multi-armed bandit algorithms at the slot-level. The second benchmark is the marginal posterior sampling (MPS) algorithm proposed in [70]. There are no formal performance guarantees for MPS, but under the monotonicity assumption, the authors in [70] show that MPS performs well in experiments. While the monotonicity assumption does not hold in general in our setting (as we consider non-separable reward functions), MPS can still be implemented and may perform differently compared to UCB1 and TS because MPS only uses slate-level rewards in order to determine with action to select.

In the experiments, ETC-SLATE is tuned according to Corollary 4.2, as this requires the least information about the problem instance. MPS is implemented using Thompson sampling with Gaussian priors, as recommended by [70].

Results

In Figure 4.1 the cumulative regret is shown for different experimental settings and different values for the problem horizon. Each point in the graph shows the cumulative

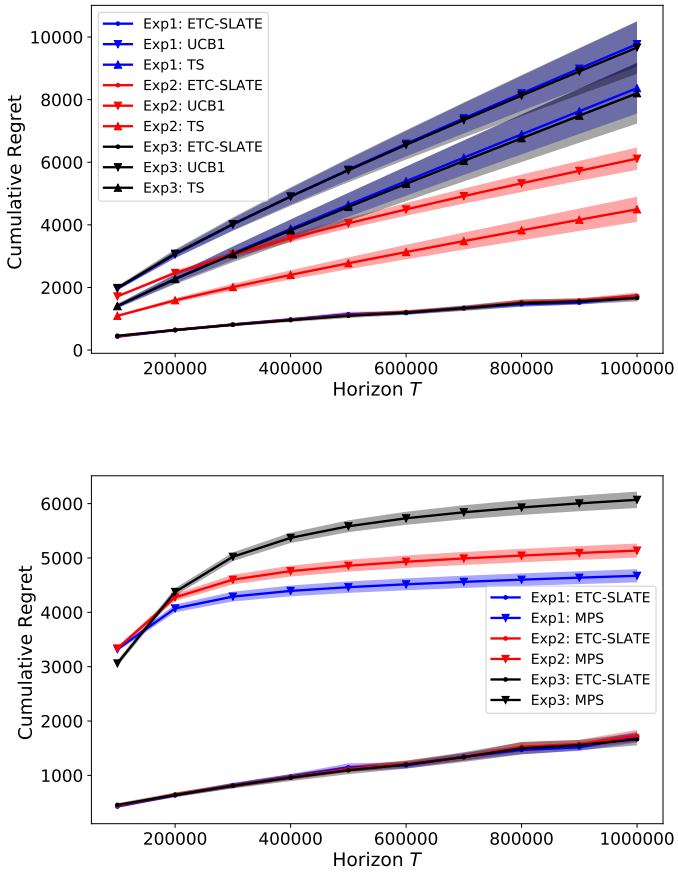


Figure 4.1: Performance of algorithms averaged over 200 runs. Lines indicate the mean and shaded region indicates 95% confidence interval.

regret over T rounds for a slate bandit problem of horizon T averaged over 200 simulations. The results indicate that ETC-SLATE clearly outperforms the benchmarks. The regret of UCB1 is at least twice as high as ETC-SLATE. TS tends to outperform UCB1, but the regret is still at least 50% higher compared to ETC-SLATE for short horizons, and twice as high for large horizons. MPS outperforms UCB1 and TS for longer horizons, but performs worse for short horizons. The regret of MPS is at least 2.5 times as high as the regret of ETC-SLATE. Also, we note that ETC-SLATE performs similarly on all the test functions, but for UCB1 and TS the performance in Exp1 and Exp3 differs from Exp2.

Figure 4.2 shows the per period reward. Here we again observe that ETC-SLATE outperforms the benchmarks. For Exp1 and Exp3, the per period reward for ETC-

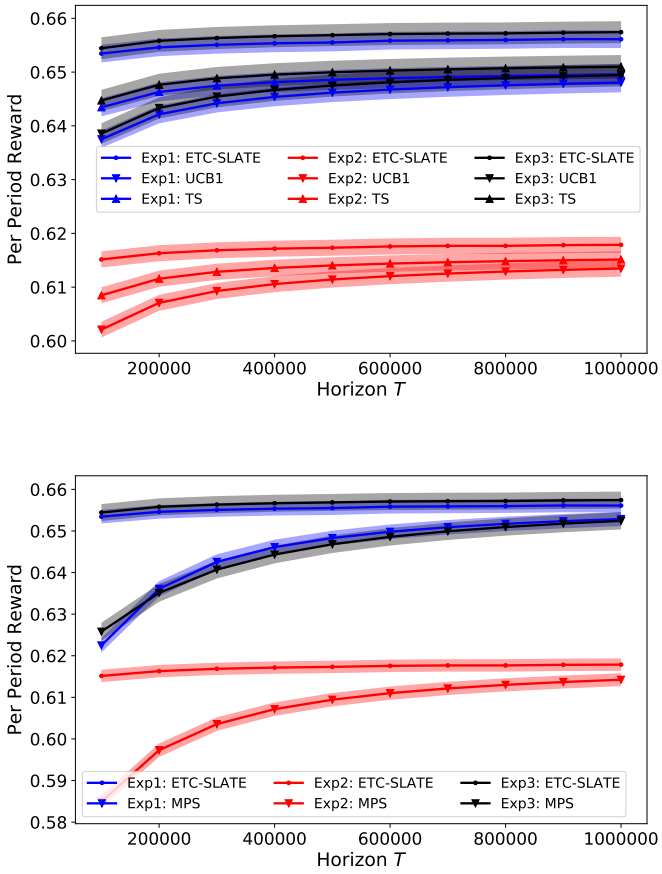


Figure 4.2: Performance of algorithms averaged over 200 runs. Lines indicate the mean and shaded region indicates 95% confidence interval.

SLATE is about 2%-2.5% higher compared to UCB1 and TS for the shorter horizons and about 1% higher for longer horizons. For Exp2 the differences are smaller: the per period reward for ETC-SLATE is about 1% higher compared to UCB1 and TS for the shorter horizons and about 0.5% higher for longer horizons. MPS performs worse than UCB1 and TS for short horizons and performs similarly for large horizons. The results show that small differences in per period reward can be associated with large differences in regret.

Finally, the results in Figure 4.1 also confirm that the regret bound from Corollary 4.2 indeed holds. However, by comparing the regret curve with the expression for the regret bound, it appears that the bound is not tight and this suggests that the bound could be improved further.

4.5.2 Experiments using real-world data

In this section we perform experiments on the reserve price optimization problem with header bidding. In this problem, there are M SSPs on the header bidding platform. In every round t, the publisher needs to choose a reserve price b_i from the set $\mathcal{B}_i$. The revenue on the header bidding platform when action $b \in \mathcal{B}$ chosen is given by $Y(b) = f(Y_1(b_1), \ldots, Y_M(b_M)) = \max\{Y_1(b_1), \ldots, Y_M(b_M)\}$.

Dataset description

In order to evaluate our method we use real-life data from ad auction markets from the publicly available iPinYou dataset [170]. It contains bidding information from the perspective of nine advertisers on a Demand Side Platform (DSP) during a week. The dataset contains information about the top bid and the second bid if the advertiser wins an auction. We use the iPinYou dataset to construct synthetic data for the top bid and second bid in order to test our proposed approach.

We use data from the advertisers to model the bids from an SSP. Fix an advertiser (say advertiser m) and fix an hour of the day (say hour h). For advertiser m we take the values of the second highest bid in hour h and we filter these values by the ad exchange (there are two ad exchanges) on which the bids were placed. Next, we sample (with replacement) 10000 values for each ad exchange to approximate the distribution of the second bid. After these steps we end up with 2 lists $L_{m,h,1}$ and $L_{m,h,2}$ of size 10000 for each ad exchange for advertiser m in hour h. Define $L_{m,h}^{max}$ as the maximum value of all values in $L_{m,h,1}$ and $L_{m,h,2}$. We use the following procedure to construct the bids for a horizon of length T. For round $t \in \{1, \ldots, T\}$ we draw A_t uniformly at random from $L_{m,h,1}$ and B_t uniformly at random from $L_{m,h,2}$. The highest bid in round t is given by $X_t = \max\{A_t, B_t\}/L_{m,h}^{max}$ and the second-highest bid is given by $Y_t = \min\{A_t, B_t\}/L_{m,h}^{max}$. Denote the resulting joint distribution by $D_{m,h}$.

Experimental settings

In the experiments we assume that there are $M = 4$ SSPs. The action sets $\mathcal{B}_i$ for SSP i is given by $K = 15$ reserve prices which are equally spaced in the interval $[0.1, 0.8]$. We consider three experimental settings and in each setting the distributions $D_{m,h}$ are different. The different experimental settings are summarized as follows: (i) in setting Exp1 we use data from advertisers 1458, 3358, 3386 and 3427 on day 2 and from hour 18; (ii) in setting Exp2 we use data from advertisers 1458, 3358, 3386 and 3427 on day 2 and from hour 15; (iii) in setting Exp3 we use data from advertisers 1458, 2261, 2821 and 3427 on day 3 and from hour 18. In the experiments, ETC-SLATE is tuned according to Corollary 4.2 and we use the same benchmarks as in the previous experiments.

Results

Figure 4.3 shows the per period reward. We again see that ETC-SLATE outperforms all of the benchmarks. From this figure we observe that the difference in performance is quite substantial as ETC-SLATE has a reward that is on average 10% higher than the UCB1 and TS benchmarks. MPS tends to outperform UCB1 and TS, and the performance is better for large horizons T. The per period reward for ETC-SLATE is about 2%-2.5% higher than MPS for the shorter horizons (depending on the specific experimental setting) and about 1%-1.5% higher than MPS for longer horizons. As small differences in per period reward can be associated with large differences in regret (see Figure 4.1 and Figure 4.2), the results indicate that the benefits of using ETC-SLATE can be substantial.

Furthermore, the results indicate that the difference in performance is not sensitive with respect to the underlying distributions at the slot-level.

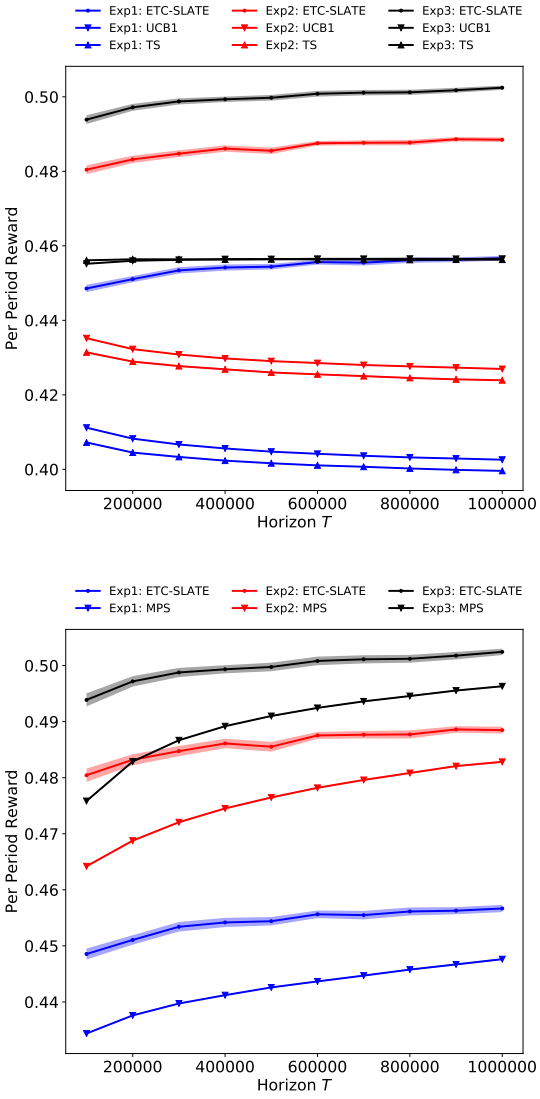


Figure 4.3: Performance of algorithms averaged over 200 runs. Lines indicate the mean and shaded region indicates 95% confidence interval.

4.6 Conclusion

In this Chapter we study slate bandits with a non-separable reward function at the slate-level. In a slate bandit problem, a slate consists of a number of slots and each slot has a number of base actions. At the slot level, every base action leads to a reward. The slate-level reward is a function (a combination) of the rewards at the

slot level. The non-separability property of the reward function implies that choosing the base action with the highest expected reward at the slot level and combining these base actions to form the action at the slate level, does not necessarily lead to the highest expected reward at the slate-level. Previous papers have only considered the case where the reward function satisfies a monotonicity property. We show that existing approaches that rely on the monotonicity property are not suitable for the slate bandit problem considered in this Chapter. We provide a theoretical analysis of our proposed algorithms and derive problem-dependent and problem-independent regret bounds. We show that our proposed algorithms have sub-linear regret with respect to the time horizon. In addition, we show that our algorithms can even be applied when the number of slates is larger than the horizon and that they will still have regret of order $T^{2/3}$. This is in contrast with benchmark algorithms such as UCB1, which cannot be applied in that case. Our solution therefore provides a substantial improvement relative to what is possible based on the current state-of-the-art.

The work presented in this Chapter can be extended in a number of ways. In our analysis we made the assumption that the slot-level rewards are independent from each other. It is not clear how to tackle the slate bandit problem when this independence assumption is relaxed and future research can be directed towards deriving sub-linear regret bounds for this case. Our algorithms have an explore-then-commit type of structure. Another interesting question is whether techniques such as Thompson Sampling can be used to guide the exploration-exploitation trade-off.

Appendix

4.A Proofs for Example 4.1

In this section we provide some missing details related to Example 4.1 in the main text. For convenience, Example 4.1 is restated below as Example 4.3.

Example 4.3. *Consider a simple instance of the slate bandit problem where there are $M = 2$ slots. Let $\mathcal{B}_1 = \{a, b\}$, $\mathcal{B}_2 = \{c, d\}$. Let $Y_1(a) \sim U(0.4, 0.5)$, $Y_1(b) \sim U(0.0, 0.1)$, $Y_2(c) \sim U(0.4, 0.5)$, $Y_2(d) \sim U(0.15, 0.7)$. Here $U(v, w)$ denotes a uniform distribution on $[v, w]$. For each slot, there are 2 actions. There are 4 slates in total and the slates are given by $\mathcal{B} = \{\{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}\}$.*
The rewards at the slate level are given by:

$$Y(\{a, c\}) = \max\{Y_1(a), Y_2(c)\}, \qquad Y(\{a, d\}) = \max\{Y_1(a), Y_2(d)\},$$
$$Y(\{b, c\}) = \max\{Y_1(b), Y_2(c)\}, \qquad Y(\{b, d\}) = \max\{Y_1(b), Y_2(d)\}.$$

Let $\mu_a = \mathbb{E}\{Y_1(a)\}$, $\mu_b = \mathbb{E}\{Y_1(b)\}$, $\mu_c = \mathbb{E}\{Y_2(c)\}$ and $\mu_d = \mathbb{E}\{Y_2(d)\}$.

Existing algorithms in [52, 53, 70, 112, 132] make a monotonicity assumption. This assumption states that if the vector of mean rewards of the slots in a slate (say slate A) dominates the vector of mean rewards of the slots in another slate (say slate B), then the expected reward of slate A is at least as high as the expected reward of slate B. A vector $W \in \mathbb{R}^n$ dominates a vector $Z \in \mathbb{R}^n$ if, for all $i = 1, \ldots, n$, the i-th component of W is at least as large as the i-th component of Z. In this example the monotonicity assumption implies that, if $\mu_c \geq \mu_d$, then it must be that $\mathbb{E}\{Y(\{a,d\})\} \leq \mathbb{E}\{Y(\{a,c\})\}$.
Note that from the properties of the uniform distribution we have that $\mathbb{E}\{Y_1(a)\} = \mathbb{E}\{Y_2(c)\} > \mathbb{E}\{Y_2(d)\}$. Therefore, the monotonicity assumption implies that we should have $\mathbb{E}\{Y(\{a,d\})\} \leq \mathbb{E}\{Y(\{a,c\})\}$. However, it can be shown that $\mathbb{E}\{Y(\{a,d\})\} > \mathbb{E}\{Y(\{a,c\})\}$ in this example. Therefore, the monotonicity assumption implies that slate $\{a,c\}$ has an expected reward that is at least as high as the expected reward of slate $\{a,d\}$ and this implication is false. Note that the vector of expected rewards of slate $\{a,c\}$ dominates the vector of expected rewards of all other slates. Therefore, under the monotonicity assumption, the slate $\{a,c\}$ is actually the optimal slate and thus the best action. In this example, the optimal action is thus not correctly identified. Existing algorithms that rely on the monotonicity assumption are therefore not guaranteed to learn the best action in this slate bandit problem. □

The claim that was not proven in the main text is given by Proposition 4.3 below.

Proposition 4.3. *In Example 4.3 we have $\mathbb{E}\{Y(\{a,d\})\} > \mathbb{E}\{Y(\{a,c\})\}$.*

The proof of Proposition 4.3 makes use of Fact 4.1, which is stated below.

Fact 4.1. *If X is a non-negative random variable, then $\mathbb{E}\{X\} = \int_0^\infty \mathbb{P}\{X \geq z\}\, \mathrm{d}z$.*

The proof of Proposition 4.3 makes use of Lemma 4.2 and Lemma 4.3, which are proven below. The proof of Proposition 4.3 will then follow from these Lemmas.

Lemma 4.2. *Let $X \sim U(a,b)$ and $Y \sim U(a,b)$ with X independent of Y, then $\mathbb{E}\{\max\{X,Y\}\} = a + \frac{2}{3}(b-a)$.*

Proof. Let $Z = \max\{X,Y\}$. Note that $\mathbb{P}\{Z \leq z\} = \mathbb{P}\{X \leq z\} \cdot \mathbb{P}\{Y \leq z\}$, because X and Y are independent.

Using the properties of the Uniform distribution we get that, $\mathbb{P}\{Y \leq z\} = \frac{z-a}{b-a}$ and $\mathbb{P}\{X \leq z\} = \frac{z-a}{b-a}$, if $a \leq z \leq b$.
If $z \leq a$, we have $\mathbb{P}\{X \leq z\} = \mathbb{P}\{Y \leq z\} = 0$.
If $z \geq b$, we have $\mathbb{P}\{X \leq z\} = \mathbb{P}\{Y \leq z\} = 1$.

Using Fact 4.1, we obtain $\mathbb{E}\{Z\} = \int_0^\infty 1 - \mathbb{P}\{Z \leq z\}\, \mathrm{d}z = \int_0^a 1\, \mathrm{d}z + \int_a^b 1 - \left(\frac{z-a}{b-a}\right)^2 \mathrm{d}z$.
This yields $\mathbb{E}\{Z\} = a + \frac{2}{3}(b-a)$. This completes the proof. □

Lemma 4.3. *Let $X, W \sim U(0.4, 0.5)$ and $Y \sim U(0.15, 0.7)$ with X, W and Y all independent of each other, then $\mathbb{E}\{\max\{X, Y\}\} > \mathbb{E}\{\max\{X, W\}\}$.*

Proof. Let $Z = \max\{X, Y\}$. We can write:

$$\mathbb{E}\{Z\} = \mathbb{E}\{Z|Y \leq 0.4\} \cdot \mathbb{P}\{Y \leq 0.4\} + \mathbb{E}\{Z|Y \geq 0.4\} \cdot \mathbb{P}\{Y \geq 0.4\}.$$

Now, for $\mathbb{E}\{Z|Y \geq 0.4\}$ we can write:

$$\begin{aligned}\mathbb{E}\{Z|Y \geq 0.4\} = {} & \mathbb{E}\{Z|0.4 \leq Y \leq 0.5, Y \geq 0.4\} \cdot \mathbb{P}\{0.4 \leq Y \leq 0.5|Y \geq 0.4\} \\ & + \mathbb{E}\{Z|0.5 \leq Y \leq 0.7, Y \geq 0.4\} \cdot \mathbb{P}\{0.5 \leq Y \leq 0.7|Y \geq 0.4\}.\end{aligned}$$

Note that conditional on the event $\{0.4 \leq Y \leq 0.5\}$, we have that $Y \sim U(0.4, 0.5)$.
Using Lemma 4.2, we get $\mathbb{E}\{Z|0.4 \leq Y \leq 0.5, Y \geq 0.4\} = 0.4 + \frac{2}{3}(0.5 - 0.4) = 0.4 + \frac{0.2}{3}$.
Note that conditional on the event $\{0.5 \leq Y \leq 0.7\}$, we have that $Z = Y$.
Therefore, we obtain $\mathbb{E}\{Z|0.5 \leq Y \leq 0.7, Y \geq 0.4\} = 0.6$.
This yields:

$$\begin{aligned}\mathbb{E}\{Z|Y \geq 0.4\} = {} & \mathbb{E}\{Z|0.4 \leq Y \leq 0.5, Y \geq 0.4\} \cdot \mathbb{P}\{0.4 \leq Y \leq 0.5|Y \geq 0.4\} \\ & + \mathbb{E}\{Z|0.5 \leq Y \leq 0.7, Y \geq 0.4\} \cdot \mathbb{P}\{0.5 \leq Y \leq 0.7|Y \geq 0.4\} \\ = {} & \left(0.4 + \frac{0.2}{3}\right) \cdot \frac{1}{3} + 0.6 \cdot \frac{2}{3}\end{aligned}$$

Note that conditional on the event $\{Y \leq 0.4\}$, we have that $Z = X$ with $X \sim U(0.4, 0.5)$.
Therefore, we obtain $\mathbb{E}\{Z|Y \leq 0.4\} = \mathbb{E}\{X\} = 0.45$.

Putting everything together we obtain:

$$\mathbb{E}\{Z\} = 0.45 \cdot \frac{0.25}{0.55} + \left[\left(0.4 + \frac{0.2}{3}\right) \cdot \frac{1}{3} + 0.6 \cdot \frac{2}{3}\right] \cdot \frac{0.30}{0.55} \approx 0.5076.$$

Now, using Lemma 4.2, we obtain $\mathbb{E}\{\max\{X, W\}\} = 0.4 + \frac{0.2}{3} \approx 0.4667 < \mathbb{E}\{Z\}$.
This completes the proof. □

Proof (of Proposition 4.3). The assertion in the statement of Proposition 4.3 follows directly from Lemma 4.3. □

Chapter 5

Maximizing revenue for publishers using header bidding and ad exchange auctions*

In practice publishers may have access to multiple selling mechanisms. More specifically, some publishers use both header bidding and an ad exchange in order to sell impressions. This Chapter builds on the ideas and problem settings of Chapter 3 and 4 and studies how publishers should should set their floor prices in order to maximize revenues when they have access to both header bidding and an ad exchange.

*This chapter is based on Rhuggenaath et al. [134].

5.1 Introduction

One of the main selling channels in online display advertising is the Real-Time Bidding (RTB) selling channel, where impressions are sold in real-time via auctions when users visit websites the are owned by publishers [162]. In this Chapter, we focus on two popular selling mechanisms [73, 97] that can be used to sell impressions via RTB: (i) ad exchange (ADX) auctions, and (ii) Header Bidding (HB).

In RTB there are three main platforms: supply side platforms (SSPs), demand side platforms (DSPs) and an ad exchange which connects SSPs and DSPs. The SSPs (e.g., DoubleClick for Publishers, Rubicon Project for Sellers and MoPub) collect inventory of different publishers and thus serve the supply side of the market. Advertisers which are interested in showing online advertisements are connected to DSPs. ADX auctions proceed as follows. When a user loads a page, the publisher's ad server calls (typically, the publisher uses an SSP for this purpose) the ad exchange. The DSPs (e.g., MediaMath and Criteo) receive the bid requests from the ad exchanges they are connected to, and bid on behalf of the advertisers. Once the ad exchange has received the bids from the DSPs, it sends the clearing price back to the publisher.

Header bidding (e.g., [73, 97]) is a relatively new mechanism, where the publisher can simultaneously connect to multiple header bidding partners (these are SSPs and ad exchanges) for a single impression. Each header bidding partner is involved in a separate auction and reports a value (a bid) indicating the revenue for the publisher. The publisher observes the individual revenues and subsequently chooses a winner among the header bidding partners (the partner with the highest revenue wins).

Publishers typically specify a floor price (a minimum amount they want to receive for the impression) when selling on the RTB market. If the price returned by the ADX auction or HB auction is lower than the floor price, the impression is not sold. Floor prices allow publisher some control on the revenues they receive: if a publisher thinks its inventory is undervalued (and it does not want to sell) it can enforce this via the floor price.

Publishers typically have access to both selling mechanisms: they first observe the offered price from header bidding and can accept or reject this price [73, 100, 133]. If the price is rejected, they can try to sell the impression on an ADX (typically via their primary/ default SSP). In this Chapter, we study how publishers should set their floor prices in order to maximize expected revenues when they have access to both selling mechanisms. In particular, we study (i) when a publisher should sell via header bidding and when he should use the ADX; and (ii) if the publisher uses ADX, which floor price it should use on the ADX. We summarize the main contributions of this Chapter as follows:

- To the best of our knowledge, we are the first to study the joint optimization of revenues for publishers that use both a header bidding platform and ADX auctions. We show how floor prices should be adjusted in order to carefully manage the exploration-exploitation trade-offs.

- We propose two algorithms and provide a theoretical analysis that shows that they have sub-linear regret with respect to the time horizon.
- We perform experiments using simulated data and using real-world data in order to validate our proposed algorithms.

The remainder of this Chapter is organized as follows. In Section 5.2 we discuss the related literature. Section 5.3 provides a formulation of the problem. In Section 5.4 we present our proposed algorithms and provide a theoretical analysis. In Section 5.5 we perform experiments in order to assess the quality of our proposed algorithms. Section 5.6 concludes our work.

5.2 Related Literature

In the last decade there has been a lot of research on online auctions and revenue management in online advertising, see e.g. [46, 87, 114, 162, 168]. However, to the best of our knowledge, we are the first to study the joint optimization of revenues for publishers that use both a header bidding platform and ad exchange auctions. Previous works typically focus on a single selling mechanism (ad exchanges that use second-price auctions [18, 41, 124, 146, 171]), or focus on the perspective of the ad exchange or header bidding partner [97, 133]. Revenue optimization in second-price auctions (using reserve prices) has been studied in [18, 41, 124, 146, 171]. These studies either assume that the bids are observed or that the number of bidders is known, and attempt to use prediction models from machine learning or use online learning techniques in order to learn the optimal reserve price. These papers, however, do not consider header bidding, and therefore, do not specify how publishers should make decisions when they have access to both an ad exchange and Header Bidding. In [100] both ad exchange auctions and header bidding are studied, however, the focus is not on revenue maximization. Revenue optimization using header bidding has been studied before by [97, 133], but these papers consider the perspective of the ad exchange or an SSP and not the perspective of the publisher.

In terms of methodology, the work in this Chapter is related to works that combine techniques from the literature of multi-armed bandits (MAB) [37, 148, 154] with online advertising auctions. Some papers [72, 81, 155] study bidding strategies or buying decisions for advertisers. For example, in [72, 155] the focus is on buying decision in order to maximize clicks when click-through-rates are unknown and typically with budget constraints. There is also a stream of literature that uses multi-armed bandit algorithms in order to design (auction) mechanisms, see e.g. [22, 32, 67, 68, 84]. The goal in such studies is to design (truthful) mechanisms that either maximize revenue of the seller or welfare, when decisions are made based on low-regret algorithms. This is not the focus of this Chapter as the publisher is not a mechanism designer.

We refer the reader to the papers cited above for additional references. The main difference between the work in this Chapter and the aforementioned works, is that (i)

we focus on revenue optimization when the publisher has access to both ad exchange auctions and a header bidding platform; and (ii) we focus on the publisher perspective (and not the perspective of the buyer or ad exchange or mechanism designer).

5.3 Problem Formulation

We consider a publisher that owns a single advertisement slot and sequentially sells impressions arriving over time. There are a number of *rounds* and in each round one impression becomes available (i.e., every time a user loads a webpage an impression becomes available and a new round begins). The total number of rounds is denoted by $T \in \mathbb{N}$. In each round $t \in \{1, \dots, T\}$ the publisher has access to an ADX and access to HB in order to sell this impression. In each round t, the publisher specifies a floor price $h_t \in \mathbb{R}$ to decide whether an impression is sold via HB. If the impression is not sold via HB, then the publisher specifies a floor price $f_t \in \mathcal{P}$ for ADX, where $\mathcal{P}$ is the set of admissible prices.

We assume that the decisions and sequence of events proceed according to the following online protocol. In each round $t \in \{1, \dots, T\}$:

1. the publisher specifies a floor price $h_t \in \mathbb{R}$ for the HB auction.
2. the publisher observes the result (i.e., the revenue m_t) of the HB auction.
3. the publisher decides to accept or reject the offer from the HB auction. If $m_t \geq h_t$, the impression is sold via HB and the revenue for the publisher equals m_t, and round t ends. Otherwise, the HB offer is rejected and the impression is not sold via HB. If the HB offer is rejected, then go to step 4.
4. if the offer from HB was rejected, the publisher selects a floor price $f_t \in \mathcal{P}$ and offers the impression for sale on ADX with a floor price f_t.
5. the publisher observes the result (i.e., the revenue v_t) of the ADX auction. The observed revenue v_t satisfies the following relations: either $v_t = 0$, or $v_t \geq f_t$.

We make the following assumptions regarding the revenue distributions.

Assumption 5.1. *We assume that* $2 \leq |\mathcal{P}| = K \leq T$ *and that* $0 \leq p^1 < p^2 < \cdots < p^K \leq 1$ *with* $p^k \in \mathcal{P}$ *for* $k = 1, \dots, K$.

Assumption 5.2. *The revenues are bounded such that* $m_t \in [0, 1]$ *and* $v_t \in [0, 1]$ *for all* t.

Assumption 5.3. *If floor price* $f_t = p^k$ *is selected in round* t*, then* v_t *is an i.i.d. draw from the distribution* X_k *and this is denoted by* $v_t \sim X_k$.

In our formulation, the sequence $m_1, \dots, m_T$ can be any arbitrary sequence such that $m_t \in [0, 1]$. Let $\mathcal{K} = \{1, \dots, K\}$. For $p^k \in \mathcal{P}$ define the quantity $\mu_k = \mathbb{E}\{X_k\}$,

let $k^* = \text{argmax}_{k \in \mathcal{K}} \mu_k$ and let $\mu^* = \mu_{k^*}$. In every round t, the publisher makes a decision $a_t = (h_t, i_t)$, where $h_t \in \mathbb{R}$ denotes the floor price for the HB auction, and $i_t \in \mathcal{K}$ denotes the index of the floor price for the ADX auction.

For a fixed sequence $\overrightarrow{m} = m_1, \ldots, m_T$ of observed revenues and a fixed sequence of decisions $a_1, \ldots, a_T$ by the publisher, the pseudo-regret over T rounds is defined as $R_T(\overrightarrow{m}) = \sum_{t=1}^{T} \max\{\mu^*, m_t\} - \sum_{t=1}^{T} \mathbb{I}\{h_t > m_t\} \cdot \mu_{i_t} - \sum_{t=1}^{T} \mathbb{I}\{h_t \leq m_t\} \cdot m_t$. The first term represents the highest expected reward that can be obtained in round t, the second term represents the expected reward if the publisher uses ADX, and the third term represents the expected reward if the publisher accepts the HB offer. An algorithm $\mathcal{A}$ is a (possibly randomized) decision rule that, to every past decisions $a_1, \ldots, a_{t-1}$ and observed values of $v_1, \ldots, v_{t-1}$ and $m_1, \ldots, m_{t-1}$ associates the next choice a_t (note that v_t is not observed if $m_t \geq h_t$).

The expected pseudo-regret over T rounds is defined as $\mathcal{R}_T(\overrightarrow{m}) = \mathbb{E}\{R_T(\overrightarrow{m})\}$, where the expectation is taken with respect to possible randomness in the selected actions $a_1, \ldots, a_T$ using $\mathcal{A}$. In the remainder, the expected pseudo-regret will simply be referred to as the regret. The notation using $\overrightarrow{m}$ makes it clear that the regret depends on the sequence of observed revenues. We will omit this dependence when the meaning is clear from the context or when a relation is understood to hold for all possible revenue sequences. For example, we write $\mathcal{R}_T \leq O(\sqrt{T})$ when $\mathcal{R}_T(\overrightarrow{m}) \leq O(\sqrt{T})$ for all possible values of $\overrightarrow{m}$.

The objective of the publisher is to devise an algorithm that makes decisions such that the regret is as low as possible. Intuitively, the publisher only wants to accept the revenue from the HB auction if $m_t \geq \mu^*$. Otherwise, the publisher want to use ADX with floor price equal to p^{k^*}, since this yields the highest expected reward μ^*.

In our formulation, the ADX auction is essentially treated as a black-box that takes the floor price as input, and where each floor price has its own revenue distribution. This is motivated by the following (combination of) reasons: (i) the ADX auction format is possibly unknown to the publisher (in our formulation the format can be arbitrary); (ii) impressions with different floor prices may get allocated to auctions of different quality by the ADX.

The online protocol captures a number of salient features of the selling process that are relevant for publishers that are small and medium sized enterprises (SMEs). The first feature relates to the timing of events. Publishers usually first receive an offer from the HB auction, and then need to decide if they accept or reject this offer[5]. If the offer is rejected, then the ad server attempts to sell the impression via an ADX auction. The second feature relates to the limited feedback that publishers receive. If the publisher never sells via ADX auctions, he cannot estimate the expected reward from ADX auctions. Moreover, if the publisher uses an ADX auction, he can only observe the revenue for the chosen floor price. Also, in ADX auctions, the publisher

[5]We consider a setting where the header bidding auction takes place directly inside the browser of the website visitor (this is called client-side header bidding [129]). In this setting, the publisher observes the outcome of the header bidding auction (m_t) and applies the floor price to this outcome.

does not observe the actual offer made (bids placed) but only observes the end result of the auction[6]. Finally, publishers need to make decisions under uncertainty. In the protocol, the revenue distributions and the sequence of revenues from the HB auction are not known in advance. As a consequence, the publisher needs to devise an algorithm that learns these unknown quantities over time and makes decisions based on the accumulated information.

Our main contribution is that we propose algorithms that have regret bounds that are sub-linear in T and K and that hold for all possible values of $\overrightarrow{m}$. To the best of our knowledge, we are the first to derive such regret bounds for our problem.

5.4 Algorithms and Analysis

In this section we discuss our proposed algorithms and provide a theoretical analysis.

5.4.1 Proposed algorithms

Let $n_{t,k}$ denote the number of times that floor price p^k has been used before the start of round t. Let $\bar{v}_{t,k}$ denote the empirical mean of observed revenues for floor price p^k after selecting it $n_{t,k}$ times. Let $\mathcal{B}(x, y)$ denote a Beta distribution with mean $x/(x + y)$ and let $\mathcal{BN}(x)$ denote a Bernoulli distribution with success probability x.

Our first algorithm, referred to as UCB-HB-ADX, is given by Algorithm 5.1. The main idea in the UCB-HB-ADX algorithm is as follows. At the ADX auction level, the choices are made based on a UCB-type algorithm based on the UCB1 algorithm by [17]. The floor price h_t for the HB auction is then set equal to the highest index (or upper confidence bound) in the UCB algorithm at the ADX level. The algorithm is initialized by using each floor price p^k once in the first K rounds.

Our second algorithm, TS-HB-ADX, is based on Thompson Sampling [7, 154] and is given by Algorithm 5.2. It has a similar structure as UCB-HB-ADX, but instead of using upper confidence bounds, it samples from a posterior distribution and uses the highest sampled value as the floor price h_t.

5.4.2 Regret bounds

In this section we provide a theoretical analysis of the algorithms proposed in the previous section. First, we discuss a lower bound on the regret (Proposition 5.1). Next, we provide upper bounds for the regret of UCB-HB-ADX (Proposition 5.2) and TS-HB-ADX (Proposition 5.3), and relate these to the lower bound.

Proposition 5.1. *Suppose that $\mathcal{A}$ is an algorithm that always accepts the offer of the HB auction when $m_t \geq \mu^*$ and always rejects the offer when $m_t < \mu^*$. Then*

[6] In ADX auctions the publisher needs to send the value of the floor price to the ADX, the auction takes place on the ADX, and the ADX applies the floor price to the auction outcome. The publisher only observes the feedback as described in step 5 of the protocol.

Algorithm 5.1 UCB-HB-ADX

Require: horizon T.
Set $t = 1$. Set $n_{t,k} = \bar{v}_{t,k} = 0 \ \forall k \in \mathcal{K}$.
for $k \in \{1, \dots, K\}$ **do**
  Use $f_t = p^k$ on ADX auction and observe v_t.
  Set $\bar{v}_{t,k} = (n_{t,k} \cdot \bar{v}_{t,k} + v_t)/(n_{t,k} + 1)$.
  Set $n_{t,k} = n_{t,k} + 1$. Set $t = t + 1$.
end for
for $t \in \{K + 1, \dots, T\}$ **do**
  Set $r_{t,k} = \sqrt{(1.5 \log t)/n_{t,k}}$. Set $I_{t,k} = \bar{v}_{t,k} + r_{t,k}$.
  Set $\hat{k} = \text{argmax}_{k \in \mathcal{K}}\{I_{t,k}\}$. Set $h_t = I_{t,\hat{k}}$.
  if $m_t \geq h_t$ **then**
    Accept offer from HB auction at price m_t.
  else
    Reject offer from HB auction at price m_t.
    Use $f_t = p^{\hat{k}}$ on ADX auction and observe v_t.
    Set $\bar{v}_{t,\hat{k}} = (n_{t,\hat{k}} \cdot \bar{v}_{t,\hat{k}} + v_t)/(n_{t,\hat{k}} + 1)$.
    Set $n_{t,\hat{k}} = n_{t,\hat{k}} + 1$.
  end if
end for

Algorithm 5.2 TS-HB-ADX

Require: horizon T.
Set $t = 1$. Set $\alpha_k = \beta_k = 1 \ \forall k \in \mathcal{K}$.
for $k \in \{1, \dots, K\}$ **do**
  Use $f_t = p^k$ on ADX auction and observe v_t.
  Draw $x \sim \mathcal{BN}(v_t)$.
  Set $\alpha_k = \alpha_k + x$. Set $\beta_k = \beta_k + (1 - x)$.
  Set $t = t + 1$.
end for
for $t \in \{K + 1, \dots, T\}$ **do**
  Draw $I_{t,k}$ from $\mathcal{B}(\alpha_k, \beta_k)$.
  Set $\hat{k} = \text{argmax}_{k \in \mathcal{K}}\{I_{t,k}\}$. Set $h_t = I_{t,\hat{k}}$.
  if $m_t \geq h_t$ **then**
    Accept offer from HB auction at price m_t.
  else
    Reject offer from HB auction at price m_t.
    Use $f_t = p^{\hat{k}}$ on ADX auction and observe v_t.
    Draw $x \sim \mathcal{BN}(v_t)$.
    Set $\alpha_{\hat{k}} = \alpha_{\hat{k}} + x$. Set $\beta_{\hat{k}} = \beta_{\hat{k}} + (1 - x)$.
  end if
end for

there exists a sequence of revenues $\overrightarrow{m} = m_1, \dots, m_T$ *such that* $\mathcal{R}_T(\overrightarrow{m}) \geq \Omega(\sqrt{KT})$ *for algorithm* $\mathcal{A}$.

Proof. Let $\overrightarrow{m} = m_1, \dots, m_T$ be such that $m_t < \mu^*$ for all t. In this case, the optimal

decision is to go to ADX in all rounds. The problem then reduces to a standard stochastic multi-armed bandit problem for which a lower bound of $\mathcal{R}_T \geq \Omega(\sqrt{KT})$ is known [148]. □

Proposition 5.2. *If Algorithm 5.1 is used on a problem instance with horizon T, then $\mathcal{R}_T \leq O(\sqrt{TK \log(T)})$.*

Proof. If $\mathbb{I}\{\mu^* > m_t \geq h_t\} = 1$ then the publisher did not go to the ADX when instead he should have. Similarly, if $\mathbb{I}\{\mu^* < m_t < h_t\} = 1$, then the publisher did not accept the revenue of the HB auction when instead he should have accepted it.

Define $A = \sum_{t=1}^{T} \mathbb{E}\{(\mu^* - m_t) \cdot \mathbb{I}\{\mu^* > m_t \geq h_t\}\}$ and define
$B = \sum_{t=1}^{T} \mathbb{E}\{(m_t - \mu^*) \cdot \mathbb{I}\{\mu^* < m_t < h_t\}\}$.
Note that we can bound the regret as follows $\mathcal{R}_T \leq K + A + B$. The first term reflects the fact that Algorithm 5.1 plays each floor price once in the first K rounds and that the regret in these rounds is bounded by $K \cdot 1$ by Assumption 5.2.

We will bound each term separately. Define the events, $F_t = \{\mu^* > m_t \geq h_t\}$, $E_t = \{h_t > \mu^*\}$, $H_t = \{\forall k : |\bar{v}_{t,k} - \mu_k| \leq \sqrt{\frac{1.5 \log T}{n_{t,k}}}\}$, and $H_t^C = \{\exists k : |\bar{v}_{t,k} - \mu_k| > \sqrt{\frac{1.5 \log T}{n_{t,k}}}\}$.

For term A we have,

$$\begin{aligned} A &\leq \sum_{t=1}^{T} \mathbb{E}\{(\mu^* - m_t) \cdot \mathbb{I}\{F_t\}\} \leq \sum_{t=1}^{T} \mathbb{E}\{1 \cdot \mathbb{I}\{F_t\}\} \\ &\leq \sum_{t=1}^{T} \mathbb{P}\{F_t\} \leq \sum_{t=1}^{T} \mathbb{P}\{\mu^* > h_t\} \leq \sum_{t=1}^{T} \mathbb{P}\{\mu^* > I_{t,k^*}\}. \end{aligned}$$

Using Hoeffding's inequality [89] and a union bound we obtain
$\mathbb{P}\{\mu^* > I_{t,k^*}\} \leq \frac{2}{t^2}$. This yields $\sum_{t=1}^{T} \mathbb{P}\{\mu^* > I_{t,k^*}\} \leq \sum_{t=1}^{\infty} \frac{2}{t^2} = 2 \cdot \frac{\pi^2}{6}$.
Therefore, we conclude that $\sum_{t=1}^{T} \mathbb{P}\{\mu^* > I_{t,k^*}\} \leq \frac{\pi^2}{3}$.

Define $\bar{\mathcal{T}} = \{t \in \mathcal{T} \mid h_t > m_t\}$. For term B we have,

$$\begin{aligned} B &\leq \sum_{t \in \bar{\mathcal{T}}} \mathbb{E}\{(m_t - \mu^*) \cdot \mathbb{I}\{\mu^* < m_t < h_t\}\} \\ &\leq \sum_{t \in \bar{\mathcal{T}}} \mathbb{E}\{(h_t - \mu^*) \cdot \mathbb{I}\{E_t\}\} \leq B_1 + B_2, \end{aligned}$$

where $B_1 = \sum_{t \in \bar{\mathcal{T}}} \mathbb{E}\{(h_t - \mu^*) \cdot \mathbb{I}\{E_t \cap H_t\}\}$ and where
$B_2 = \sum_{t \in \bar{\mathcal{T}}} \mathbb{E}\{(h_t - \mu^*) \cdot \mathbb{I}\{E_t \cap H_t^C\}\}$.

Let $\hat{k}_t = \text{argmax}_{k\in\mathcal{K}}\{I_{t,k}\}$ and note that $h_t = I_{t,\hat{k}_t}$.
We bound B_1 as follows.

$$\begin{aligned}
B_1 &\leq \sum_{t\in\bar{\mathcal{T}}} \mathbb{E}\left\{|h_t - \mu^*| \cdot \mathbb{I}\left\{H_t\right\}\right\} \\
&\leq \sum_{t\in\bar{\mathcal{T}}} \mathbb{E}\Big\{ |h_t - \mu^*| \;\Big|\; H_t \Big\} \cdot \mathbb{P}\left\{H_t\right\} \\
&\leq \sum_{t\in\bar{\mathcal{T}}} \mathbb{E}\Big\{ |h_t - \mu^*| \;\Big|\; H_t \Big\} \leq \sum_{t\in\bar{\mathcal{T}}} \mathbb{E}\Big\{ |I_{t,\hat{k}_t} - \mu_{\hat{k}_t}| \;\Big|\; H_t \Big\} \\
&\leq \sum_{t\in\bar{\mathcal{T}}} 2\sqrt{\frac{3\log T}{2n_{t,\hat{k}_t}}} \leq \sum_{t\in\mathcal{T}} 2\sqrt{\frac{3\log T}{2n_{t,\hat{k}_t}}} \leq \sum_{k=1}^{K}\sum_{t\in\mathcal{T}_k} 2\sqrt{\frac{3\log T}{2n_{t,k}}} \\
&\leq \sum_{k=1}^{K} 2\cdot 2\sqrt{|\mathcal{T}_k|\frac{3\log T}{2}} \overset{(a)}{\leq} 2\cdot 2\sqrt{TK\frac{3\log T}{2}},
\end{aligned}$$

where $\mathcal{T}_k = \{t : \hat{k}_t = k\}$.
Inequality (a) follows from $\sum_{k=1}^{K}\frac{1}{K}\sqrt{|\mathcal{T}_k|} \overset{(b)}{\leq} \sqrt{\sum_{k=1}^{K}\frac{1}{K}\cdot|\mathcal{T}_k|} = \sqrt{\frac{1}{K}\sum_{k=1}^{K}|\mathcal{T}_k|} \overset{(c)}{\leq} \sqrt{\frac{1}{K}T}$. Inequality (b) follows from noting that $\sqrt{x}$ is a concave function and by applying Jensen's inequality and inequality (c) follows from the fact that $\sum_{k=1}^{K}|\mathcal{T}_k| \leq T$.

For B_2 we have, $B_2 \leq \sum_{t\in\bar{\mathcal{T}}}\mathbb{P}\left\{H_t^C\right\} \leq T\cdot\frac{2}{T}$.

Putting everything together we obtain $\mathcal{R}_T \leq K + \frac{\pi^2}{3} + 4\sqrt{1.5\log T}\sqrt{TK} + T\cdot\frac{2}{T}$. Therefore, we conclude that $\mathcal{R}_T \leq O(K + \sqrt{KT\log T})$. Assuming that $T \geq K$, we have that $\mathcal{R}_T \leq O(\sqrt{KT\log T})$. □

Proposition 5.3 bounds the regret of TS-HB-ADX. It uses Lemma 5.1 and Lemma 5.2, which are presented below. Lemma is Fact 1 in [6] and Lemma 5.2 is based on Lemma 1 in [104].

Lemma 5.1. *Let $F^{Beta}_{a,b}(\cdot)$ denote the cumulative distribution function (CDF) of a Beta distribution with parameters a and b. Let $F^{B}_{v,w}(\cdot)$ denote the CDF of a Binomial distribution with v trials and success parameter w. Then, $F^{Beta}_{a,b}(y) = 1 - F^{B}_{a+b+1,y}(a-1)$.*

Lemma 5.2. *Suppose that Algorithm 5.2 is used on a problem instance with $T \geq 2$ and let $n_{t,k}$ denote the number of times that floor price k has been used before round t. If $t \geq K$, then $\mathbb{P}\left\{|\mu_k - I_{t,k}| \geq \sqrt{\frac{32\log T}{n_{t,k}}}\right\} \leq \frac{2}{T^3}$.*

Proof. Let $t \geq K$. In this case, each floor price on ADX has been used at least once. Let $\alpha_{t,k}$ denote the value of α_k before round t. Let $\mathcal{BIN}(n,p)$ denote a Binomial distribution with n trials and success probability p. Define the following

events $A_t = \{n_{t,k} \geq 1\}$, $B_t = \{I_{t,k} \leq \mu_k - \sqrt{\frac{32\log T}{n_{t,k}}}\}$ and $C_t = \{I_{t,k} \geq \mu_k + \sqrt{\frac{32\log T}{n_{t,k}}}\}$. Let (U_t) denote a sequence of i.i.d. uniform random variables. Now in round $t \geq K$,

$$\begin{aligned}
\mathbb{P}\{B_t\} &= \mathbb{P}\left\{U_t \leq F^{Beta}_{\alpha_k+1, n_{t,k}-\alpha_k+1}(\mu_k - \sqrt{\frac{32\log T}{n_{t,k}}})\right\} \\
&\overset{(a)}{=} \mathbb{P}\left\{\left\{U_t \leq 1 - F^{B}_{n_{t,k}+1, \mu_k - \sqrt{\frac{32\log T}{n_{t,k}}}}(\alpha_k)\right\} \cap A_t\right\} \\
&= \mathbb{P}\left\{\left\{F^{B}_{n_{t,k}+1, \mu_k - \sqrt{\frac{32\log T}{n_{t,k}}}}(\alpha_k) \leq U_t\right\} \cap A_t\right\} \\
&\leq \mathbb{P}\left\{\exists s \in \{1,\dots,t\} : F^{B}_{s+1, \mu_k - \sqrt{\frac{32\log T}{s}}}(\alpha_{s,k}) \leq U_t\right\} \\
&\overset{(b)}{\leq} \sum_{s=1}^{t} \mathbb{P}\left\{\alpha_{s,k} \leq (F^{B})^{-1}_{s+1, \mu_k - \sqrt{\frac{32\log T}{s}}}(U_t)\right\}.
\end{aligned}$$

Here $(F)^{-1}$ denotes the inverse CDF of the CDF F. Equality (a) follows from Lemma 5.1 and inequality (b) follows from a union bound. Note that $V_s = (F^B)^{-1}_{s+1, \mu_k - \sqrt{\frac{32\log T}{s}}}(U_t) \sim \mathcal{BIN}(s+1, \mu_k - \sqrt{\frac{32\log T}{s}})$ and is independent from $\alpha_{s,k} \sim \mathcal{BIN}(s, \mu_k)$.

For a fixed value of s, define two i.i.d. sequences of Bernoulli random variables $W_{1,l} \sim \mathcal{BN}(\mu_k - \sqrt{\frac{32\log T}{s}})$ and $W_{2,l} \sim \mathcal{BN}(\mu_k)$ and let $Z_l = W_{2,l} - W_{1,l}$ with $\mathbb{E}\{Z_l\} = \sqrt{\frac{32\log T}{s}}$.

Now,

$$\begin{aligned}
\mathbb{P}\{\alpha_{s,k} \leq V_s\} &= \mathbb{P}\left\{\sum_{l=1}^{s} W_{2,l} \leq \sum_{l=1}^{s+1} W_{1,l}\right\} \\
&\leq \mathbb{P}\left\{\sum_{l=1}^{s} Z_l \leq W_{1,s+1}\right\} \\
&\leq \mathbb{P}\left\{\sum_{l=1}^{s} Z_l \leq 1\right\} \\
&= \mathbb{P}\left\{\sum_{l=1}^{s} (Z_l - \sqrt{\frac{32\log T}{s}}) \leq -(\sqrt{s32\log T} - 1)\right\} \\
&\overset{(c)}{\leq} \mathbb{P}\left\{\sum_{l=1}^{s} (Z_l - \sqrt{\frac{32\log T}{s}}) \leq -\sqrt{s8\log T}\right\} \\
&\overset{(d)}{\leq} \exp\{-2\frac{(\sqrt{8s\log T})^2}{4s}\} = \frac{1}{T^4}.
\end{aligned}$$

Inequality (c) follows from $0.5\sqrt{s32\log T} \geq \sqrt{s8\log T} > 1$ for $T \geq 2$ and $s \geq 1$. Inequality (d) follows from Hoeffding's inequality. Therefore, it follows that $\mathbb{P}\{B_t\} \leq \sum_{s=1}^{t} \mathbb{P}\{\alpha_{s,k} \leq V_s\} \leq \frac{T}{T^4} = \frac{1}{T^3}$.

Define $W_{1,l}^{+} \sim \mathcal{BN}(\mu_k + \sqrt{\frac{32\log T}{s}})$ and let $Z_l^{+} = W_{1,l}^{+} - W_{2,l}$. Using a similar approach one can show that $\mathbb{P}\{C_t\} \leq \sum_{s=1}^{t} \mathbb{P}\{\alpha_{s,k} \geq V_s^{+}\} \leq \frac{T}{T^4} = \frac{1}{T^3}$, where $V_s^{+} = (F^B)^{-1}_{s+1,\mu_k+\sqrt{\frac{32\log T}{s}}}(U_t) \sim \mathcal{BIN}(s+1, \mu_k + \sqrt{\frac{32\log T}{s}})$. By taking a union bound, we obtain $\mathbb{P}\left\{|\mu_k - I_{t,k}| \geq \sqrt{\frac{32\log T}{n_{t,k}}}\right\} \leq \frac{2}{T^3}$. □

Proposition 5.3. *If Algorithm 5.2 is used on a problem instance with horizon T, then $\mathcal{R}_T \leq O(\sqrt{KT\log T})$).*

Proof. The proof is similar to the proof of Proposition 5.2.
Define $A = \sum_{t=1}^{T} \mathbb{E}\{(\mu^* - m_t) \cdot \mathbb{I}\{\mu^* > m_t \geq h_t\}\}$ and define
$B = \sum_{t=1}^{T} \mathbb{E}\{(m_t - \mu^*) \cdot \mathbb{I}\{\mu^* < m_t < h_t\}\}$.
Note that we can bound the regret as follows $\mathcal{R}_T \leq K + A + B$.

We will bound each term separately. Let $n_{t,k}$ denote the number of times that floor price k has been used before round t. Define the events, $F_t = \{\mu^* > m_t \geq h_t\}$, $E_t = \{h_t > \mu^*\}$, $H_t = \{\forall k : |\mu_k - I_{t,k}| \leq \sqrt{\frac{32\log T}{n_{t,k}}}\}$ and $H_t^C = \{\exists k : |\mu_k - I_{t,k}| > \sqrt{\frac{32\log T}{n_{t,k}}}\}$. Note that, by Lemma 5.2 and by using a union bound over the K floor prices and using the fact that $K \leq T$, we have that

$\mathbb{P}\left\{H_t^C\right\} \leq \frac{2}{T^2}$.

Define $\hat{\mathcal{T}} = \{t \in \mathcal{T} \mid h_t \leq m_t\}$. For term A we have,

$$A \leq \sum_{t\in\hat{\mathcal{T}}} \mathbb{E}\left\{(\mu^* - m_t)\cdot \mathbb{I}\left\{F_t\right\}\right\} \leq \sum_{t\in\hat{\mathcal{T}}} \mathbb{E}\left\{|\mu^* - I_{t,k^*}|\cdot \mathbb{I}\left\{F_t\right\}\right\} \leq A_1 + A_2,$$

where $A_1 = \sum_{t\in\hat{\mathcal{T}}} \mathbb{E}\left\{|\mu^* - I_{t,k^*}|\cdot \mathbb{I}\left\{F_t \cap H_t\right\}\right\}$ and where
$A_2 = \sum_{t\in\hat{\mathcal{T}}} \mathbb{E}\left\{|\mu^* - I_{t,k^*}|\cdot \mathbb{I}\left\{F_t \cap H_t^C\right\}\right\}$.

For A_1 we have,

$$\begin{aligned} A_1 &\leq \sum_{t\in\hat{\mathcal{T}}} \mathbb{E}\left\{|\mu^* - I_{t,k^*}|\cdot \mathbb{I}\left\{H_t\right\}\right\} \\ &\leq \sum_{t\in\hat{\mathcal{T}}} \mathbb{E}\Big\{ |\mu^* - I_{t,k^*}| \;\Big|\; H_t \Big\} \\ &\leq \sum_{t\in\hat{\mathcal{T}}} \sqrt{\frac{32\log T}{n_{t,\hat{k}_t}}} \overset{(a1)}{\leq} 2\sqrt{KT32\log T}. \end{aligned}$$

Here inequality (a1) follows from similar arguments used for inequality (a) in Proposition 5.2.

For A_2 we have, $A_2 \leq \sum_{t\in\hat{\mathcal{T}}} \mathbb{P}\left\{H_t^C\right\} \leq T\cdot\frac{2}{T^2}$.

Define $\bar{\mathcal{T}} = \{t \in \mathcal{T} \mid t > N, h_t > m_t\}$. For term B we have,

$$\begin{aligned} B &\leq \sum_{t\in\bar{\mathcal{T}}} \mathbb{E}\left\{(m_t - \mu^*)\cdot \mathbb{I}\left\{\mu^* < m_t < h_t\right\}\right\} \\ &\leq \sum_{t\in\bar{\mathcal{T}}} \mathbb{E}\left\{(h_t - \mu^*)\cdot \mathbb{I}\left\{E_t\right\}\right\} \leq B_1 + B_2, \end{aligned}$$

where $B_1 = \sum_{t\in\bar{\mathcal{T}}} \mathbb{E}\left\{(h_t - \mu^*)\cdot \mathbb{I}\left\{E_t \cap H_t\right\}\right\}$ and where
$B_2 = \sum_{t\in\bar{\mathcal{T}}} \mathbb{E}\left\{(h_t - \mu^*)\cdot \mathbb{I}\left\{E_t \cap H_t^C\right\}\right\}$.

Let $\hat{k}_t = \text{argmax}_{k\in\mathcal{K}}\{I_{t,k}\}$ and note that $h_t = I_{t,\hat{k}_t}$. Following similar steps as in the

proof of Proposition 5.2, we obtain

$$\begin{aligned}
B_1 &\leq \sum_{t\in\bar{\mathcal{T}}} \mathbb{E}\left\{|h_t - \mu^*| \cdot \mathbb{I}\left\{H_t\right\}\right\} \\
&\leq \sum_{t\in\bar{\mathcal{T}}} \mathbb{E}\Big\{ |h_t - \mu^*| \;\Big|\; H_t \Big\} \cdot \mathbb{P}\left\{H_t\right\} \\
&\leq \sum_{t\in\bar{\mathcal{T}}} \mathbb{E}\Big\{ |h_t - \mu^*| \;\Big|\; H_t \Big\} \leq \sum_{t\in\bar{\mathcal{T}}} \mathbb{E}\Big\{ |I_{t,\hat{k}_t} - \mu_{\hat{k}_t}| \;\Big|\; H_t \Big\} \\
&\leq \sum_{t\in\bar{\mathcal{T}}} \sqrt{\frac{32\log T}{n_{t,\hat{k}_t}}} \overset{(a2)}{\leq} 2\sqrt{KT32\log T}.
\end{aligned}$$

Here inequality (a2) follows from inequality (a1).

For B_2 we have, $B_2 \leq \sum_{t\in\bar{\mathcal{T}}} \mathbb{P}\left\{H_t^C\right\} \leq T \cdot \frac{2}{T^2}$.

Putting everything together we obtain $\mathcal{R}_T \leq K + 4\sqrt{32KT\log T} + T \cdot \frac{4}{T^2}$. Therefore, we conclude that $\mathcal{R}_T \leq O(K + \sqrt{KT\log T})$. Assuming that $T \geq K$, we have that $\mathcal{R}_T \leq O(\sqrt{KT\log T})$. □

If we compare the upper bounds on the regret with the lower bound, then we observe the following. Proposition 5.1 shows that, in general, the problem considered in this paper is in a sense at least as hard as a standard stochastic multi-armed bandit problem. In Proposition 5.2 and 5.3 the regret scales with $\sqrt{TK\log T}$ which matches the lower bound of Proposition 5.1 up to logarithmic factors. Therefore, our proposed algorithms are optimal up to logarithmic factors.

Proposition 5.2 and 5.3 show that our TS-based and UCB-based algorithms have similar regret guarantees (up to constant factors), but it is still useful to consider both, because TS-based approaches have often been reported to perform better (e.g. [104]). In our experiments, TS-HB-ADX outperforms UCB-HB-ADX, which is consistent with the empirical performance of TS-based approaches in the literature (e.g. [104]).

5.5 Experiments

In this section we conduct experiments in order to test the performance of our proposed algorithms. We conduct experiments using both simulated data and real-world data.

5.5.1 Experiments using simulated data

Experimental settings

We consider a setting with $K = 5$ floor prices for the ADX auction. In the experiments, X_k follows a uniform distribution on $[\mu_k - x, \mu_k + x]$ with x chosen at random from $\{0.1, 0.2, 0.3\}$. We set $\mu_1 = 0.5$, $\mu_2 = 0.6$, $\mu_3 = 0.7$, $\mu_4 = 0.65$ and $\mu_5 = 0.6$. With these choices, we have that $\mu^* = 0.7$. We consider three settings for the process that generates the revenues from the HB auction. In setting A, m_t is an i.i.d. draw from $\mathcal{B}(u, v)$, where $u = 2 \cdot w$ and $v = 2 \cdot (1 - w)$ and $w = \mu^* + 0.1$. Setting B is the same as setting A, except that $w = \mu^* - 0.1$. Setting C is the same as setting A, except that $w = \mu^* - 0.1$ for rounds $t \leq T/2$ and $w = \mu^* + 0.1$ for rounds $t > T/2$.

Setting A represents a scenario where the average revenue from header bidding is on average higher than on the ad exchange. In setting B, the average revenue from header bidding is lower on average. Finally, in setting C, the average revenue from header bidding is initially lower than on the ad exchange, but afterwards, a change occurs and the average revenue from is on average higher than on the ad exchange. By analyzing these three settings, we can test whether the results depend heavily on the process the generates the revenues from header bidding.

We run TS-HB-ADX and UCB-HB-ADX for a horizon of $T = 50000$.

Results

In Figure 5.1 the cumulative regret is shown for different experimental settings and different values for the problem horizon. As the algorithms do not require knowledge of T, each point in the graph with round number $t \leq T$ can be interpreted as the cumulative regret after t rounds for a problem of horizon t averaged over 200 simulations. In all figures the shaded region indicates the 95% confidence interval, but since it is too small in some figures, some intervals are not visible. The results indicate that the regret is sub-linear in the horizon T for both algorithms. The results show that TS-HB-ADX generally outperforms UCB-HB-ADX, as the regret is much lower for all experimental settings. In Figure 5.2 the fraction of rounds in which the optimal action is taken is displayed. The results confirm what we already saw in Figure 5.1, namely that as the number of rounds increases, the optimal action is played more often. Moreover, we again observe that TS-HB-ADX generally outperforms UCB-HB-ADX.

5.5.2 Experiments using real-world data

Dataset

We use header bidding data from a SME publisher that publishes gaming content. The data is from February 22, 2020 (day 1) and February 21, 2020 (day 2) and contains the bids placed by the Header Bidding partners and the final result of the

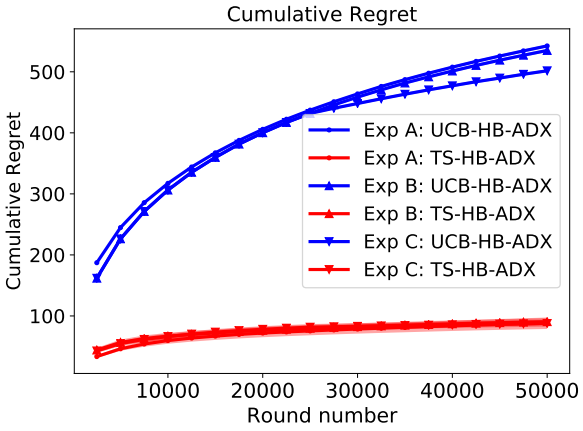


Figure 5.1: Cumulative regret using UCB-HB-ADX and TS-HB-ADX.

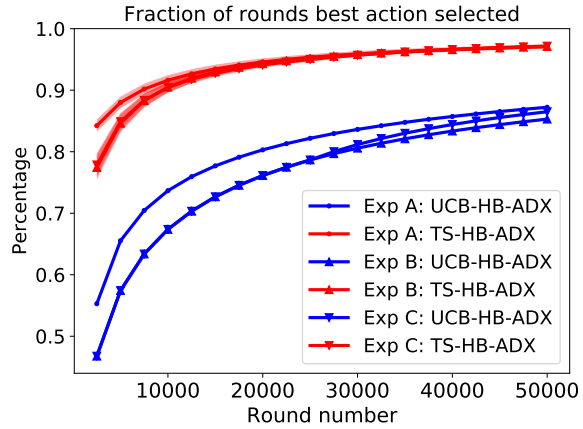


Figure 5.2: Fraction of rounds best action selected using UCB-HB-ADX and TS-HB-ADX.

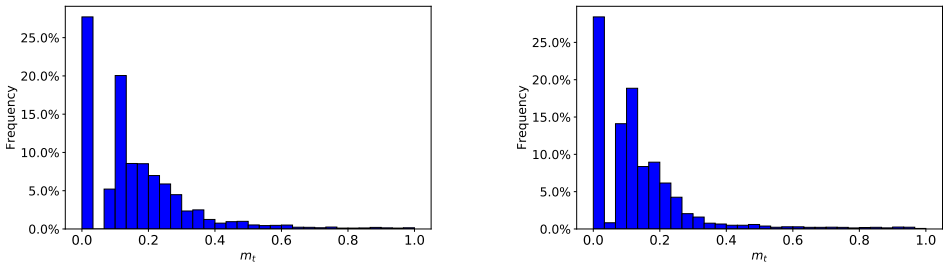
HB auction (maximum of the bids). We use this data to construct the distributions X_k for the revenue on the ADX auction.

We perform the following pre-processing steps on the raw data. Let L_{MAX} denote a list of numbers containing the observed results of the HB auctions. First, we determine the 95-th percentile q_{MAX} of the positive values in L_{MAX} and keep all the HB auctions where values in L_{MAX} are at most q_{MAX}. Next, to preserve proprietary information, we shift the bids of all HB partners by a small positive constant c. Finally, we normalize the bids of all the HB partners to the range $[0, 1]$ by dividing by $q_{MAX} + c$. Let M denote the resulting empirical distribution of the values L_{MAX}

(after normalizing). Let HB_1 and HB_2 denote the empirical distributions of the bids from Header Bidding partner 1 and 2 (after normalizing), respectively.

Experimental settings

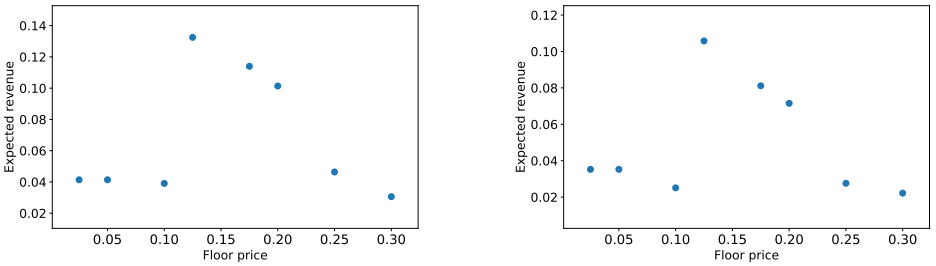
The set of floor prices for the ADX auction is $\mathcal{P} = \{0.025, 0.05, 0.1, 0.125, 0.175, 0.2, 0.25, 0.3\}$. In the experiments we assume that v_t is an i.i.d. draw: (i) from HB_1 if $p^k \leq 0.1$; (ii) from M if $0.1 < p^k \leq 0.2$; from (iii) HB_2 if $0.2 < p^k \leq 0.3$. In the experiments m_t is an i.i.d. draw from M. With these settings $\mathbb{E}\{m_t\} \approx 0.1583$ and $\mu^* \approx 0.1325$ for day 1, and $\mathbb{E}\{m_t\} \approx 0.1349$ and $\mu^* \approx 0.1058$ for day 2. Figure 5.3 displays the distribution for m_t for both days. Figure 5.4 displays the expected reward for each floor price on the ADX for both days. We run TS-HB-ADX and UCB-HB-ADX for a horizon of $T = 100000$.



(a) Distribution of m_t for Day 1.

(b) Distribution of m_t for Day 2.

Figure 5.3: Distribution of m_t.



(a) Expected rewards for Day 1.

(b) Expected rewards for Day 2.

Figure 5.4: Expected rewards for floor prices.

Results

The results are displayed in Figures 5.5 to 5.7. The results in Figure 5.5 and 5.6 are qualitatively similar to those reported in Figure 5.1 and 5.2: we again observe that regret is sub-linear in T and that TS-HB-ADX generally outperforms UCB-HB-ADX.

Figure 5.7 displays the average cumulative revenue for both TS-HB-ADX and UCB-HB-ADX. The results show that the average cumulative revenue, for both algorithms and for both days, exceeds both μ^* and $\mathbb{E}\{m_t\}$ as the number of rounds increases. This shows that optimization of revenues that takes both selling mechanisms into account leads to a substantial improvement over the expected revenue that can be obtained by using just one selling mechanism.

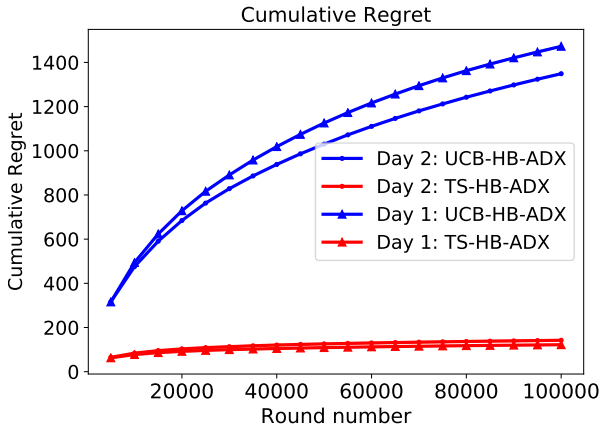


Figure 5.5: Cumulative regret using UCB-HB-ADX and TS-HB-ADX.

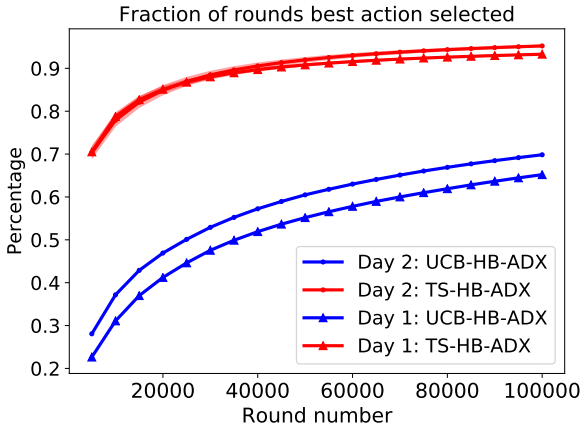


Figure 5.6: Fraction of rounds best action selected using UCB-HB-ADX and TS-HB-ADX.

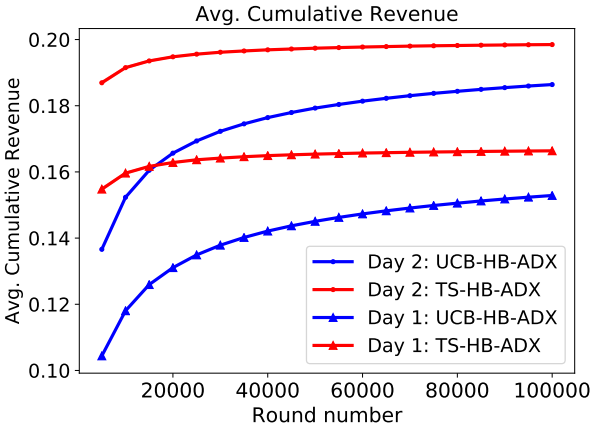


Figure 5.7: Average cumulative revenue using UCB-HB-ADX and TS-HB-ADX.

5.6 Conclusion

In this Chapter, we study how publishers should set their floor prices in order to maximize expected revenues when they have access to both an ad exchange and header bidding in order to sell impressions on the Real-Time Bidding market. We formulate the problem as a regret minimization problem and develop algorithms based on techniques from the multi-armed bandits literature. We provide a theoretical analysis of our algorithms and verify our results on simulated data and real-world data. Our experiments show that optimization of revenues that takes both selling mechanisms

into account leads to a substantial improvement over the expected revenue that can be obtained by using just one selling mechanism.

We present algorithms with regret guarantees that hold for all possible revenue sequences on the header bidding platform. Our analysis shows that our proposed algorithms match a lower bound up to logarithmic factors. Therefore, our proposed algorithms are optimal up to logarithmic factors. Future work can be directed towards algorithms that take additional information into account in the form of features, similar to for example contextual bandits. It would be interesting to study whether optimal algorithms exist and which assumptions are needed in order to derive such performance guarantees. Another direction would be to investigate whether it is possible to derive performance guarantees when the rewards on the ad exchange are adversarial instead of stochastic.

Chapter 6

Algorithms for strategic buyers with unknown valuations in repeated posted-price auctions*

The previous Chapters of this thesis have studied revenue management problems from the perspective of the seller. This Chapter switches perspectives and instead focuses on buying decisions in online advertising auctions. More specifically, we consider buying decisions in repeated posted-price auction where a seller repeatedly interacts with a buyer for a number of time periods and where the buyer wants to maximize his expected utility over time. In this problem, the buyer needs to learn at what prices it is worthwhile to purchase an item when her valuation for the item is unknown.

---

*This chapter is based on Rhuggenaath et al. [142].

6.1 Introduction

A growing fraction of online advertisements are sold via ad exchanges. In an ad exchange, after a visitor arrives on a webpage, advertisers compete in an auction to win the impression (the right to deliver an ad to that visitor). Typically, these auctions are second-price auctions, where the winner pays the second highest bid or a reserve price (whichever is larger), and no sale occurs if all of the bids are lower than the reserve price. However, as indicated by e.g. [10, 11, 122], a non-trivial fraction of auctions only involve a single bidder and this reduces to a posted-price auction [106] when reserve prices known: the seller sets a reserve price and the buyer decides whether to accept or reject it. A single publisher can track a large number of visitors with similar properties over time and sell these to buyers. As buyers typically are involved in a large number of auctions, there is an incentive for them to act strategically [10, 11, 78, 122]. These observations have led to the study of repeated posted-price auctions between a single seller and strategic buyer.

In this Chapter we consider a repeated posted-price auction between a single seller and a single buyer similar to that considered in [10, 122]. In every round, the seller posts a price and the buyer decides to buy or not at that price. The buyer does not know the distribution of his valuation, the seller's pricing algorithm or the seller's price set. Furthermore, the seller does not know the valuation distribution and needs to learn how to set the price over time. There are a number of differences between the work in this Chapter and previous work on repeated posted-price auction such as [10, 11, 122]. First, unlike in previous work, we study the problem from the perspective of a buyer that aims to maximize his expected utility or surplus, instead of the perspective of the seller that aims to maximize his revenue. Second, previous papers assume that the buyer knows his valuation in each round. In this Chapter, we relax this assumption and assume the buyer does not know the distribution of his valuation and the valuation is only revealed after he buys the item. This is motivated by applications in online advertising where the buyer (advertiser) does not know the exact value of showing the ads to a set of users: some users may click on the ad and in some cases the ad may lead to a sale, but the buyer only observes a response after he displays the advertisement to the user.

As the valuation distribution is unknown, buyers face an exploration-exploitation trade-off and their decisions lead to regret: (i) accepting a price that is at most the mean valuation leads to positive expected utility and accepting a price above it leads to negative utility; (ii) buying the item leads to additional information about the mean valuation (at the risk of negative utility), but by not buying there is a risk of missing out on positive utility. We study two types of buyers: strategic buyers and non-strategic buyers. Non-strategic buyers are only interested achieving sub-linear regret given the prices that are observed and do not attempt to manipulate or influence the observed prices. Strategic buyers are also interested in sub-linear regret given the observed prices, but they also actively attempt to influence future prices

that will be offered. If non-strategic buyers knew the mean valuation they would use the following rule: always accept a price that is at most the mean valuation and always reject a price above it. Strategic buyers on the other hand, would sometimes deviate from this rule in an attempt to influence future prices that will be offered. If non-strategic buyers knew the mean valuation, then their decisions would have low regret but the seller could learn to ask a price very close to the mean valuation, resulting in low utility for the buyer [10, 122]. Strategic buyers attempt to influence the learning process of the seller in order to lower the price and to increase the utility. However, as these attempts are not guaranteed to succeed (as buyers don't know the seller's pricing algorithm or price set), strategic buyers still want to ensure sub-linear regret for all possible prices sequences.

In our setting, the seller needs to learn to set his prices because he does not know the valuation distribution. To the best of our knowledge, there are no existing 'optimal' algorithms with performance guarantees (specifically) for repeated posted-price auctions with a single seller and a single strategic buyer that doesn't know his valuation: existing algorithms (e.g., [10, 11, 75, 76, 122, 159]) assume that buyers know their valuation and thus lose their performance guarantees. In our experiments (see Section 6.5) we therefore assume that the seller uses an off-the-shelf low-regret learning algorithm for adaptive adversarial bandit feedback as these have known performance guarantees [37, 106, 148].

Our main contributions are as follows. First, to the best of our knowledge, we are the first to study repeated posted-price auctions in strategic settings from the perspective of the buyer. We do not assume that the buyer knows his valuation distribution. Second, we construct algorithms with sub-linear (in the problem horizon) regret for both non-strategic and strategic buyers by using ideas from popular multi-armed bandit algorithms UCB1 [17] and Thompson Sampling [8]. Our algorithms do not require knowledge about the seller's pricing algorithm or price set. Third, we use experiments to support our theoretical findings. Using experiments we show that, if the seller is using a low-regret learning algorithm based on weights updating (such as EXP3.P [15, 37]), then strategic buyers can obtain much higher utilities compared to non-strategic buyers.

The remainder of this Chapter is organized as follows. In Section 6.2 we discuss the related literature. Section 6.3 provides a formal description of the problem. In Section 6.4 we present the our proposed algorithms and provide a theoretical analysis. In Section 6.5 we perform experiments in order to assess the quality of our proposed algorithms. Section 6.6 provides a conclusion and some directions for further research.

6.2 Related Literature

The work in this Chapter is mainly related to the following areas of the literature: posted-price auctions, low-regret learning by sellers and buyers, and decision making

for buyers in auctions. We discuss these areas in more detail below.

Repeated posted-price auctions with the goal of maximizing revenue for the seller and assuming that the feedback from buyers is i.i.d. distributed was studied in [106]. Other works [10, 11, 75, 76, 95, 122, 159] instead study repeated posted-price auctions with strategic buyers. However, these papers all study the seller side of the problem and assume that buyers know their valuations in each round.

On a high level, work in this Chapter is related to works that study repeated auctions where either the seller and/or the buyer is running a low-regret learning algorithm [32, 35, 67] and the interaction between bandit algorithms and incentives of buyers [21, 22, 68, 84]. The goal in such studies is to design (truthful) mechanisms that either maximize revenue of the seller or welfare, when decisions are made based on low-regret algorithms. This is not the focus of this Chapter.

The aforementioned works focus on either the seller side or on mechanism design, but there is also work that considers the perspective of buyers or bidders. In [72, 155, 156] the focus is on maximizing clicks when click-through-rates are unknown and typically with budget constraints. In this Chapter, rewards for buyers are not determined by the number of clicks, instead the buyer aims to maximize cumulative utilities or his net surplus as in e.g., [10, 11, 122]. In [81] the focus is on designing bidding strategies for buyers that compete against each other and where the buyer valuation is unknown. However, these studies do not focus on repeated posted-price auctions and strategic behaviour of buyers is not considered.

6.3 Problem Formulation

We consider a single buyer and a single seller that interact for T rounds. An item, such as an advertisement space, is repeatedly offered for sale by the seller to the buyer over these T rounds. In each round $t \in \mathcal{T} = \{1, \dots, T\}$, a price $p_t \in \mathcal{P}$ is offered by the seller and a decision $a_t \in \{0, 1\}$ is made by the buyer: $a_t = 1$ when the buyer accepts to buy at that price, $a_t = 0$ otherwise. The buyer holds a private valuation $v_t \in [0, 1]$ for the item in round t. The value of v_t is an i.i.d. draw from a distribution $\mathcal{D}$ and has expectation $\nu = \mathbb{E}\{v_t\}$. The buyer does not know $\mathcal{D}$ and ν. Also, the buyer does not know $\mathcal{P}$ or the seller's pricing algorithm. The value v_t is only revealed to the buyer if he buys the item in round t, i.e., the buyer only observes the value after he buys the item. The seller also does not know $\mathcal{D}$ or ν and does not observe v_t.

The utility of the buyer in round t is given by $u_t = a_t \cdot (v_t - p_t)$. In other words, if the buyer purchases the item the utility is the difference between the valuation and the price. Otherwise, the utility is zero. For a fixed sequence $\overrightarrow{p} = p_1, \dots, p_T$ of observed prices and a fixed sequence of decisions $a_1, \dots, a_T$ by the buyer, the pseudo-regret of the buyer over T rounds is defined as $R_T(\overrightarrow{p}) = \sum_{t=1}^{T} \max\{\nu - p_t, 0\} - \sum_{t=1}^{T} a_t \cdot (\nu - p_t)$. The term $\max\{\nu - p_t, 0\}$ represents the expected utility of the optimal decision in round t and the term $a_t \cdot (\nu - p_t)$ represents the expected utility of the actual decision

that is made by the buyer in round t. The expected pseudo-regret over T rounds is defined as $\mathcal{R}_T(\overrightarrow{p}) = \mathbb{E}\{R_T(\overrightarrow{p})\}$, where the expectation is taken with respect to possible randomization in the selection of the actions $a_1, \ldots, a_T$. In the remainder, the expected pseudo-regret will simply be referred to as the regret. The notation using $\overrightarrow{p}$ makes it clear that the regret depends on the sequence of observed prices. We will omit this dependence when the meaning is clear from the context or when a relation is understood to hold for all possible price sequences. For example, we write $\mathcal{R}_T \leq O(\sqrt{T \log T})$ when $\mathcal{R}_T(\overrightarrow{p}) \leq O(\sqrt{T \log T})$ for all choices of $\overrightarrow{p}$.

We consider two types of buyers: non-strategic buyers and strategic buyers. Non-strategic buyers are interested in achieving sub-linear regret for all possible price sequences, but they treat the price sequence as exogenous. That is, if non-strategic buyers knew ν, then they would follow this rule: buy if and only if $p_t \leq \nu$. Strategic buyers also want sub-linear regret for all possible prices sequences, but they would sometimes deviate from this rule in an attempt to influence (i.e., lower) future prices that will be offered. If non-strategic buyers knew ν, then their decisions would have low regret but the seller could learn to ask a price just below ν, resulting in low utility for the buyer [10, 122]. Strategic buyers actively attempt to influence the learning process of the seller in order to lower the price and to increase the utility. However, as these attempts are not guaranteed to succeed (recall that buyers do not know the seller's pricing algorithm or $\mathcal{P}$), strategic buyers still want to ensure sub-linear regret for all possible prices sequences. The seller does not know $\mathcal{D}$ or ν and does not observe v_t, and so he has to *learn* how to set his price over time under bandit feedback. This Chapter focuses on the buyer side and the regret bounds that we derive do not depend on the seller's pricing algorithm. However, in order to test our algorithms, some assumption about the seller's algorithm is required. To the best of our knowledge, there are no existing 'optimal' algorithms for sellers with performance guarantees (specifically) for repeated posted-price auctions with a single seller and a single strategic buyer that doesn't know his valuation: existing algorithms (e.g., [10, 11, 75, 76, 95, 122, 159]) assume that buyers know v_t and thus lose their performance guarantees. In our experiments (see Section 6.5) we therefore assume that the seller uses an off-the-shelf low-regret learning algorithm for adaptive adversarial bandit feedback as these have known performance guarantees [37, 106, 148].

Algorithm 6.1 UCB-NS

Require: horizon T.
Set $\mathcal{V} = \varnothing$. Set $t = 1$.
Set $n = 1$.
Buy item at price p_t.
Observe v_t. Set $\mathcal{V} = \mathcal{V} \cup \{v_t\}$.
for $t \in \{2, \dots, T\}$ **do**
  Set $n_t = n$.
  Set $\bar{v}_t = \frac{1}{n_t} \sum_{v \in \mathcal{V}} v$.
  Set $r_t = \sqrt{(2 \log t)/n_t}$.
  Set $I_t = \bar{v}_t + r_t$.
  if $I_t \geq p_t$ **then**
    Buy item at price p_t.
    Observe v_t. Set $\mathcal{V} = \mathcal{V} \cup \{v_t\}$.
    Set $n = n + 1$.
  end if
end for

Algorithm 6.2 TS-NS

Require: $N \in \mathbb{N}$, horizon T.
Set $\mathcal{V} = \varnothing$. Set $t = N$.
Set $n = N$.
Buy item in first N rounds.
Observe $\mathcal{V}^N = \cup_{k=1}^{N} \{v_k\}$.
Set $\mathcal{V} = \mathcal{V} \cup \mathcal{V}^N$.
for $t \in \{N + 1, \dots, T\}$ **do**
  Set $n_t = n$.
  Set $\bar{v}_t = \frac{1}{n_t} \sum_{v \in \mathcal{V}} v$.
  Sample $I_t \sim \mathcal{N}(\bar{v}_t, \frac{1}{n_t})$
  if $I_t \geq p_t$ **then**
    Buy item at price p_t.
    Observe v_t. Set $\mathcal{V} = \mathcal{V} \cup \{v_t\}$.
    Set $n = n + 1$.
  end if
end for

6.4 Algorithms and Analysis

In this section we present our proposed algorithms for strategic and non-strategic buyers and we provide a theoretical analysis of these algorithms.

6.4.1 Non-strategic buyers

We provide two algorithms for non-strategic buyers that have sub-linear regret. The first algorithm, UCB-NS, is based on UCB (upper confidence bound) style bandit algorithms [17] and the second algorithm, TS-NS, is based on the Thompson sampling principle [9]. In every round, UCB-NS maintains an optimistic estimate of the unknown mean ν and decides to buy the item if the estimate is at least as large as the offered price p_t. TS-NS samples from a posterior distribution and decides to buy the item if the sampled value is at least as large as the offered price p_t. Proposition 6.1 and 6.2 bound the regret of UCB-NS and TS-NS, respectively.

Proposition 6.1. *If Algorithm 6.1 is run with inputs: T, then $\mathcal{R}_T \leq O(\sqrt{T \log T})$.*

Proof. If $\mathbb{I}\{\nu > p_t > I_t\} = 1$ then the buyer did not buy the item when instead he should have bought it. Similarly, if $\mathbb{I}\{\nu < p_t \leq I_t\} = 1$, then the buyer did buy the item when instead he should not have bought it.

Note that we can bound the regret as follows $\mathcal{R}_T \leq 1 + \sum_{t=1}^{T} \mathbb{E}\{(\nu - p_t) \cdot \mathbb{I}\{\nu > p_t > I_t\}\} + \sum_{t=1}^{T} \mathbb{E}\{(p_t - \nu) \cdot \mathbb{I}\{\nu < p_t \leq I_t\}\}$.

Define $A = \sum_{t=1}^{T} \mathbb{E}\{(\nu - p_t) \cdot \mathbb{I}\{\nu > p_t > I_t\}\}$ and $B = \sum_{t=1}^{T} \mathbb{E}\{(p_t - \nu) \cdot \mathbb{I}\{\nu < p_t \leq I_t\}\}$. We will bound each term separately.

Define the following events $F_t = \{\nu > p_t > I_t\}$, $E_t = \{I_t > \nu\}$, $H_t = \{|\bar{v}_t - \nu| \leq \sqrt{\frac{2\log T}{n_t}}\}$ and $H_t^C = \{|\bar{v}_t - \nu| > \sqrt{\frac{2\log T}{n_t}}\}$.

For term A we have,

$$\begin{aligned} A &\leq \sum_{t=1}^{T} \mathbb{E}\{(\nu - p_t) \cdot \mathbb{I}\{F_t\}\} \leq \sum_{t=1}^{T} \mathbb{E}\{1 \cdot \mathbb{I}\{F_t\}\} \\ &\leq \sum_{t=1}^{T} \mathbb{P}\{F_t\} \leq \sum_{t=1}^{T} \mathbb{P}\{\nu > I_t\} \end{aligned}$$

Using Hoeffding's inequality (and a union bound) we obtain $\mathbb{P}\{\nu > I_t\} \leq \frac{1}{t^3} \leq \frac{1}{t^2}$. Therefore, we conclude that $\sum_{t=1}^{T} \mathbb{P}\{\nu > I_t\} \leq \frac{\pi^2}{6}$.

Define $\mathcal{B} = \{t \in \mathcal{T} \mid I_t \geq p_t\}$. For term B we have,

$$\begin{aligned} B &\leq \sum_{t \in \mathcal{B}} \mathbb{E}\{(p_t - \nu) \cdot \mathbb{I}\{\nu < p_t \leq I_t\}\} \leq \sum_{t \in \mathcal{B}} \mathbb{E}\{(I_t - \nu) \cdot \mathbb{I}\{I_t > \nu\}\} \\ &\leq \sum_{t \in \mathcal{B}} \mathbb{E}\{(I_t - \nu) \cdot \mathbb{I}\{E_t \cap H_t\}\} + \sum_{t \in \mathcal{B}} \mathbb{E}\left\{(I_t - \nu) \cdot \mathbb{I}\left\{E_t \cap H_t^C\right\}\right\}. \end{aligned}$$

Define $B_1 = \sum_{t \in \mathcal{B}} \mathbb{E}\{(I_t - \nu) \cdot \mathbb{I}\{E_t \cap H_t\}\}$. We bound B_1 as follows:

$$\begin{aligned} B_1 &\leq \sum_{t \in \mathcal{B}} \mathbb{E}\{|I_t - \nu| \cdot \mathbb{I}\{H_t\}\} \leq \sum_{t \in \mathcal{B}} \mathbb{E}\Big\{ |\nu - I_t| \;\Big|\; H_t \Big\} \cdot \mathbb{P}\{H_t\} \\ &\leq \sum_{t \in \mathcal{B}} \mathbb{E}\Big\{ |\nu - I_t| \;\Big|\; H_t \Big\} \leq \sum_{t \in \mathcal{B}} 2\sqrt{\frac{2\log T}{n_t}} \\ &\leq \sum_{t \in \mathcal{T}} 2\sqrt{\frac{2\log T}{t}} \leq 2\int_0^T \sqrt{\frac{2\log T}{t}} \mathrm{d}t \leq 4\sqrt{2\log T}\sqrt{T}. \end{aligned}$$

Define $B_2 = \sum_{t\in\mathcal{B}} \mathbb{E}\left\{(I_t - \nu)\cdot \mathbb{I}\left\{E_t \cap H_t^C\right\}\right\}$. We bound B_2 as follows:

$$\begin{aligned} B_2 &\leq \sum_{t\in\mathcal{B}} \mathbb{P}\left\{H_t^C\right\} \\ &\leq \sum_{t\in\mathcal{B}} \mathbb{P}\left\{|\bar{v}_t - \nu| > \sqrt{\frac{2\log T}{n_t}}\right\} \overset{(a)}{\leq} T\cdot\frac{2}{T^4}. \end{aligned}$$

Inequality (a) follows from applying Hoeffding's inequality and from the fact that $|\mathcal{B}| \leq T$.

Putting everything together we obtain $\mathcal{R}_T \leq 1 + \frac{\pi^2}{6} + 4\sqrt{2\log T}\sqrt{T} + T\cdot\frac{2}{T^4}$. Therefore, we conclude that $\mathcal{R}_T \leq O(\sqrt{T\log T})$. □

Proposition 6.2. *If Algorithm 6.2 is run with inputs: T and $N = \lceil c_N \cdot T^{\frac{2}{3}}\rceil$, then $\mathcal{R}_T \leq O(T^{\frac{2}{3}}\sqrt{\log T})$.*

Proof. We can bound the regret as follows $\mathcal{R}_T \leq N\cdot 1 + \sum_{t=N+1}^{T}\mathbb{E}\left\{(\nu - p_t)\cdot\mathbb{I}\left\{\nu > p_t > I_t\right\}\right\} + \sum_{t=N+1}^{T}\mathbb{E}\left\{(p_t - \nu)\cdot\mathbb{I}\left\{\nu < p_t \leq I_t\right\}\right\}$.

Define $A = \sum_{t=N+1}^{T}\mathbb{E}\left\{(\nu - p_t)\cdot\mathbb{I}\left\{\nu > p_t > I_t\right\}\right\}$ and $B = \sum_{t=N+1}^{T}\mathbb{E}\left\{(p_t - \nu)\cdot\mathbb{I}\left\{\nu < p_t \leq I_t\right\}\right\}$. We will bound each term separately. Define the event $F_t = \{\nu > p_t > I_t\}$.

$$\begin{aligned} A &\leq \sum_{t=N+1}^{T}\mathbb{E}\left\{(\nu - p_t)\cdot\mathbb{I}\left\{F_t\right\}\right\} \leq \sum_{t=N+1}^{T}\mathbb{E}\left\{(\nu - I_t)\cdot\mathbb{I}\left\{F_t\right\}\right\} \\ &\leq \sum_{t=N+1}^{T}\mathbb{E}\left\{|\nu - I_t|\cdot\mathbb{I}\left\{F_t\right\}\right\} \leq \sum_{t=N+1}^{T}\mathbb{E}\Big\{\,|(\nu - \bar{v}_t) - (I_t - \bar{v}_t)|\;\Big|\;F_t\Big\}\cdot\mathbb{P}\left\{F_t\right\} \\ &\leq \sum_{t=N+1}^{T}\mathbb{E}\Big\{\,|\nu - \bar{v}_t|\;\Big|\;F_t\Big\}\cdot\mathbb{P}\left\{F_t\right\} + \sum_{t=N+1}^{T}\mathbb{E}\Big\{\,|I_t - \bar{v}_t|\;\Big|\;F_t\Big\}\cdot\mathbb{P}\left\{F_t\right\} \\ &\leq \sum_{t=N+1}^{T}\mathbb{E}\left\{|\nu - \bar{v}_t|\right\} + \sum_{t=N+1}^{T}\mathbb{E}\left\{|I_t - \bar{v}_t|\right\} \end{aligned}$$

Using Hoeffding's inequality we obtain, for $t > N$, that $\mathbb{E}\left\{|\nu - \bar{v}_t|\right\} \leq \frac{2}{T^4} + 2\sqrt{\frac{2\log T}{N}}$. Using the fact that $N = \lceil c_N\cdot T^{\frac{2}{3}}\rceil$ and that $T - (N+1) \leq T$, this yields $\sum_{t=N+1}^{T}\mathbb{E}\left\{|\nu - \bar{v}_t|\right\} \leq \frac{2}{T^3} + T^{\frac{2}{3}}2\sqrt{\frac{2\log T}{c_N}}$. Using the fact that, for $t > N$, $I_t - \bar{v}_t \sim \mathcal{N}(0,\sigma^2)$ with $\sigma^2 = \frac{1}{n_t} \leq \frac{1}{N}$, we obtain that $\mathbb{E}\left\{|I_t - \bar{v}_t|\right\} \leq \sqrt{\frac{2}{\pi\cdot c_N}}T^{-\frac{1}{3}}$ and this yields $\sum_{t=N+1}^{T}\mathbb{E}\left\{|I_t - \bar{v}_t|\right\} \leq \sqrt{\frac{2}{\pi\cdot c_N}}T^{\frac{2}{3}}$.

Define $\mathcal{B} = \{t \in \mathcal{T} \mid t > N, I_t \geq p_t\}$. Let $E_t = \{I_t > \nu\}$, let $H_t = \{|\bar{v}_t - \nu| \leq \sqrt{\frac{2\log T}{n_t}}\}$ and let $H_t^C = \{|\bar{v}_t - \nu| > \sqrt{\frac{2\log T}{n_t}}\}$. Let $\hat{v}_s$ denote the sample mean of s i.i.d. draws from distribution $\mathcal{D}$ and let $\hat{I}_s \sim \mathcal{N}(\hat{v}_s, \frac{1}{s})$. For term B we have,

$$\begin{aligned} B &\leq \sum_{t\in\mathcal{B}} \mathbb{E}\left\{(p_t - \nu)\cdot \mathbb{I}\left\{\nu < p_t \leq I_t\right\}\right\} \leq \sum_{t\in\mathcal{B}} \mathbb{E}\left\{(I_t - \nu)\cdot \mathbb{I}\left\{I_t > \nu\right\}\right\} \\ &\leq \sum_{t\in\mathcal{B}} \mathbb{E}\left\{(I_t - \nu)\cdot \mathbb{I}\left\{E_t \cap H_t\right\}\right\} + \sum_{t\in\mathcal{B}} \mathbb{E}\left\{(I_t - \nu)\cdot \mathbb{I}\left\{E_t \cap H_t^C\right\}\right\}. \end{aligned}$$

Define $B_1 = \sum_{t\in\mathcal{B}} \mathbb{E}\left\{(I_t - \nu)\cdot \mathbb{I}\left\{E_t \cap H_t\right\}\right\}$. We bound B_1 as follows:

$$\begin{aligned} B_1 &\leq \sum_{t\in\mathcal{B}} \mathbb{E}\left\{|I_t - \nu|\cdot \mathbb{I}\left\{H_t\right\}\right\} \leq \sum_{t\in\mathcal{B}} \mathbb{E}\Big\{\, |\nu - I_t| \;\Big|\; H_t \,\Big\}\cdot \mathbb{P}\left\{H_t\right\} \\ &\leq \sum_{t\in\mathcal{B}} \mathbb{E}\Big\{\, |I_t - \bar{v}_t| \;\Big|\; H_t \,\Big\}\cdot \mathbb{P}\left\{H_t\right\} + \sum_{t\in\mathcal{B}} \mathbb{E}\Big\{\, |\bar{v}_t - \nu| \;\Big|\; H_t \,\Big\}\cdot \mathbb{P}\left\{H_t\right\}. \end{aligned}$$

Define $B_{11} = \sum_{t\in\mathcal{B}} \mathbb{E}\Big\{\, |I_t - \bar{v}_t| \;\Big|\; H_t \,\Big\}\cdot \mathbb{P}\left\{H_t\right\}$ and $B_{12} = \sum_{t\in\mathcal{B}} \mathbb{E}\Big\{\, |\bar{v}_t - \nu| \;\Big|\; H_t \,\Big\}\cdot \mathbb{P}\left\{H_t\right\}$.

We bound B_{11} as follows:

$$\begin{aligned} B_{11} &\leq \sum_{t\in\mathcal{B}} \mathbb{E}\Big\{\, |I_t - \bar{v}_t| \;\Big|\; H_t \,\Big\}\cdot \mathbb{P}\left\{H_t\right\} + \sum_{t\in\mathcal{B}} \mathbb{E}\Big\{\, |I_t - \bar{v}_t| \;\Big|\; H_t^C \,\Big\}\cdot \mathbb{P}\left\{H_t^C\right\} \\ &= \sum_{t\in\mathcal{B}} \mathbb{E}\{|I_t - \bar{v}_t|\} = \sum_{t\in\mathcal{B}} \mathbb{E}\Big\{|\hat{I}_{n_t} - \hat{v}_{n_t}|\Big\} \leq \sum_{t\in\mathcal{T}} \mathbb{E}\Big\{|\hat{I}_t - \hat{v}_t|\Big\} \\ &\leq \sum_{t\in\mathcal{T}} \sqrt{\frac{2}{\pi t}} \leq \int_0^T \sqrt{\frac{2}{\pi t}}\mathrm{d}t = 2\sqrt{\frac{2}{\pi}T}. \end{aligned}$$

We bound B_{12} as follows:

$$\begin{aligned} B_{12} &\leq \sum_{t\in\mathcal{B}} \mathbb{E}\Big\{\, |\bar{v}_t - \nu| \;\Big|\; H_t \,\Big\}\cdot \mathbb{P}\left\{H_t\right\} \leq \sum_{t\in\mathcal{B}} \mathbb{E}\Big\{\, |\bar{v}_t - \nu| \;\Big|\; H_t \,\Big\} \\ &\leq \sum_{t\in\mathcal{T}} \mathbb{E}\left\{\, |\hat{v}_t - \nu| \;\middle|\; |\hat{v}_t - \nu| \leq \sqrt{\frac{2\log T}{t}} \,\right\} \\ &\leq \sum_{t\in\mathcal{T}} \sqrt{\frac{2\log T}{t}} \leq \int_0^T \sqrt{\frac{2\log T}{t}}\mathrm{d}t \leq 2\sqrt{2\log T}\sqrt{T}. \end{aligned}$$

Define $B_2 = \sum_{t\in\mathcal{B}} \mathbb{E}\left\{(I_t - \nu)\cdot \mathbb{I}\left\{E_t \cap H_t^C\right\}\right\}$. We bound B_2 as follows:

$$B_2 \leq \sum_{t\in\mathcal{B}} \mathbb{P}\left\{H_t^C\right\} \leq \sum_{t\in\mathcal{B}} \mathbb{P}\left\{|\hat{v}_{n_t} - \nu| > \sqrt{\frac{2\log T}{n_t}}\right\}$$
$$\leq \sum_{t\in\mathcal{T}} \mathbb{P}\left\{|\hat{v}_t - \nu| > \sqrt{\frac{2\log T}{t}}\right\} \leq T\cdot\frac{2}{T^4}.$$

Putting everything together we obtain $\mathcal{R}_T \leq N\cdot 1 + \frac{2}{T^3} + 2T^{\frac{2}{3}}\sqrt{\frac{2\log T}{c_N}} + \sqrt{\frac{2}{\pi\cdot c_N}}T^{\frac{2}{3}} + 2\sqrt{\frac{2T}{\pi}} + 2\sqrt{2T\log T} + T\cdot\frac{2}{T^4}$. So, we conclude that $\mathcal{R}_T \leq O(T^{\frac{2}{3}}\sqrt{\log T})$. □

6.4.2 Strategic buyers

In this section we show how the algorithms for non-strategic buyers can be converted into algorithms for strategic buyers with the same growth rate (up to constant factors) for the regret. Our proposed approach BUYER-STRAT is presented in Algorithm 6.3. The main idea behind BUYER-STRAT is to take a base algorithm $\mathcal{A}_{base}$ for non-strategic buyers (e.g. UCB-NS or TS-NS) and modify it using what we refer to as *strategic cycles*.

We now give a description of how Algorithm 6.3 works. In BUYER-STRAT the buyers make decisions according to $\mathcal{A}_{base}$ for the first N_1 rounds. Afterwards, in the next N_2 rounds, we enter a so-called strategic cycle. In this strategic cycle, the buyer only buys the item if the price is below some threshold, that is, if $p_t \leq v^* - c_1$. Here v^* is an estimate of the unknown mean ν and $0 < c_1 < 1$ is a parameter chosen by the buyer (e.g. $c_1 = 0.1$). The purpose of this strategic cycle is to entice the seller into asking prices that are lower than ν. After this strategic cycle comes to an end, we start another strategic cycle of length L with some small probability p_{cycle}. If another strategic cycle has been triggered, we set a new parameter $0 < c_{target} < 1$ and only prices $p_t \leq v^* - c_{target}$ are accepted. If no strategic cycle is triggered, the buyer makes decisions according to $\mathcal{A}_{base}$. In the next round, we start a strategic cycle of length L with probability p_{cycle} and the aforementioned process is repeated.

Algorithm 6.3 makes use of the functions F_1, F_2, F_3, F_4, F_5, F_6. The intuition behind these functions is as follows. In every strategic cycle, only prices that satisfy $p_t \leq v^* - c$ are accepted, where $c \in \mathcal{C}$ for some set $\mathcal{C}$. The value of v^* is selected using the function $F_5(\cdot)$ which takes as input a base algorithm $\mathcal{A}_{base}$. The value $c \in \mathcal{C}$ is selected by using the function $F_1(\cdot)$ which depends on a counter of the number strategic cycles that have passed C_{phase}. Initially, the number of strategic cycles in which values $c \in \mathcal{C}$ are used, is equal to N_{phase}. When $F_2(x) = 1$, this indicates that the last strategic cycle in which a value $c \in \mathcal{C}$ is used has just been completed, and the function $F_3(\cdot)$ is used to collect information about the price trajectory. When $F_6(x) = 1$, a final value for p_{target} is chosen (using $F_4(\cdot)$) and only prices with $p_t \leq p_{target}$ are accepted in all subsequent strategic cycles. In Section 6.5 we discuss these

functions in more detail and give specific examples that are used in our experiments.

The key parameters to control the regret of Algorithm 6.3 are the cycle probability p_{cycle} and the cycle length L. Proposition 6.3 shows that BUYER-STRAT with $\mathcal{A}_{base}$ chosen as UCB-NS has regret of order $O(\sqrt{T \log T})$ if the probability p_{cycle} and the cycle length L is carefully chosen. Proposition 6.4 shows an analogous result for BUYER-STRAT with TS-NS.

Proposition 6.3. *Let A_p, A_L and A_N be positive real constants. Assume that Algorithm 6.3 is run with $\mathcal{A}_{base}$ chosen as UCB-NS and with inputs: T, $N_1 = \lceil T^{\frac{2}{3}} (\log T)^{\frac{1}{2}} \rceil$, $N_2 = \lceil A_N \sqrt{T \log T} \rceil$, $p_{cycle} = A_p T^{-\frac{1}{2}}$ and $L = A_L \sqrt{\log T}$, then $\mathcal{R}_T \leq O(\sqrt{T \log T})$.*

Proof. We will decompose the regret in two parts: the regret incurred in rounds that are part of strategic cycles and rounds that are not. For a arbitrary subset $\mathcal{T}^* \subseteq \mathcal{T}$, let $\mathcal{R}_{T,\mathcal{T}^*} = \sum_{t \in \mathcal{T}^*} \mathbb{E}\left\{(\nu - p_t) \cdot \mathbb{I}\left\{\nu > p_t > I_t\right\}\right\} + \sum_{t \in \mathcal{T}^*} \mathbb{E}\left\{(p_t - \nu) \cdot \mathbb{I}\left\{\nu < p_t \leq I_t\right\}\right\}$. Let $\mathcal{T}_S \subseteq \mathcal{T}$ denote the indices of the rounds that are part of strategic cycles and let $\mathcal{T}_{NS} = \mathcal{T} \setminus \mathcal{T}_S$ denote the indices of the rounds that are not. Then we can write, $\mathcal{R}_T = \mathcal{R}_{T,\mathcal{T}_{NS}} + \mathcal{R}_{T,\mathcal{T}_S}$.

For $\mathcal{R}_{T,\mathcal{T}_S}$ we have that $\mathcal{R}_{T,\mathcal{T}_S} \leq N_2 + T \cdot p_{cycle} \cdot L$. This follows from the fact that the expected number of triggered strategic cycles (after round $N_1 + N_2$) is at most $T \cdot p_{cycle}$ and the regret in every such cycle is at most L. Furthermore, the first strategic cycle has length N_2. For $\mathcal{R}_{T,\mathcal{T}_{NS}}$ we have that $\mathcal{R}_{T,\mathcal{T}_{NS}} \leq 5 + 4\sqrt{2 \log T}\sqrt{T}$. This follows from the fact that $\mathcal{R}_{T,\mathcal{T}_{NS}}$ represents the regret after $|\mathcal{T}_{NS}| \leq T$ rounds in a problem with horizon T, and by Proposition 6.1, this quantity is bounded by $5 + 4\sqrt{2 \log T}\sqrt{T}$. By plugging in the values we get $\mathcal{R}_T = \mathcal{R}_{T,\mathcal{T}_{NS}} + \mathcal{R}_{T,\mathcal{T}_S} \leq O(\sqrt{T \log T})$. □

Proposition 6.4. *Let A_p, A_L and A_N be positive real constants. Assume that Algorithm 6.3 is run with $\mathcal{A}_{base}$ chosen as TS-NS and with inputs: T, $N_1 = \lceil T^{\frac{2}{3}} (\log T)^{\frac{1}{2}} \rceil$, $N_2 = \lceil A_N \sqrt{T \log T} \rceil$, $p_{cycle} = A_p T^{-\frac{1}{2}}$ and $L = A_L \sqrt{\log T}$. Assume that TS-NS is run with inputs: T and $N = \lceil c_N \cdot T^{\frac{2}{3}} \rceil$. Then $\mathcal{R}_T \leq O(T^{\frac{2}{3}} \sqrt{\log T})$.*

Proof. The proof uses similar arguments as the proof of Proposition 6.3.
We will decompose the regret in two parts: the regret incurred in rounds that are part of strategic cycles and rounds that are not. For a arbitrary subset $\mathcal{T}^* \subseteq \mathcal{T}$, let $\mathcal{R}_{T,\mathcal{T}^*} = \sum_{t \in \mathcal{T}^*} \mathbb{E}\left\{(\nu - p_t) \cdot \mathbb{I}\left\{\nu > p_t > I_t\right\}\right\} + \sum_{t \in \mathcal{T}^*} \mathbb{E}\left\{(p_t - \nu) \cdot \mathbb{I}\left\{\nu < p_t \leq I_t\right\}\right\}$. Let $\mathcal{T}_S \subseteq \mathcal{T}$ denote the indices of the rounds that are part of strategic cycles and let $\mathcal{T}_{NS} = \mathcal{T} \setminus \mathcal{T}_S$ denote the indices of the rounds that are not. Then we can write, $\mathcal{R}_T = \mathcal{R}_{T,\mathcal{T}_{NS}} + \mathcal{R}_{T,\mathcal{T}_S}$.

For $\mathcal{R}_{T,\mathcal{T}_S}$ we have that $\mathcal{R}_{T,\mathcal{T}_S} \leq N_2 + T \cdot p_{cycle} \cdot L$. This follows from the fact that the expected number of triggered strategic cycles (after round $N_1 + N_2$) is at most $T \cdot p_{cycle}$ and the regret in every such cycle is at most L. Furthermore, the first strategic cycle has length N_2. For $\mathcal{R}_{T,\mathcal{T}_{NS}}$ we have that $\mathcal{R}_{T,\mathcal{T}_{NS}} \leq O(T^{\frac{2}{3}} \sqrt{\log T})$. This follows from the fact that $\mathcal{R}_{T,\mathcal{T}_{NS}}$ represents the regret after $|\mathcal{T}_{NS}| \leq T$ rounds in a problem

with horizon T, and by Proposition 6.2, this quantity is bounded by $O(T^{\frac{2}{3}}\sqrt{\log T})$. By plugging in the values we get $\mathcal{R}_T = \mathcal{R}_{T,\mathcal{T}_{NS}} + \mathcal{R}_{T,\mathcal{T}_S} \leq O(T^{\frac{2}{3}}\sqrt{\log T})$. □

Algorithm 6.3 BUYER-STRAT

Require: F_1, F_2, F_3, F_4, F_5, F_6, L, p_{cycle}, N_{phase}, N_1, N_2, c_1, T, $\mathcal{A}_{base}$.
Set $L_p = \varnothing$, $L_{target} = \varnothing$, $C_{phase} = 0$, $t = 1$.
for $t = 1, \ldots, N_1$ **do**
  Observe price p_t. Choose to buy or not based on $\mathcal{A}_{base}$.
end for
$v^* = F_5(\mathcal{A}_{base})$.
for $t = N_1 + 1, \ldots, N_1 + N_2$ **do**
  Observe price p_t. Buy if $p_t \leq v^* - c_1$.
end for
while $t \in \{N_1 + N_2 + 1, \ldots, T\}$ **do**
  Draw D from Bernoulli distribution with success parameter p_{cycle}.
  if $D = 1$ **then**
    $v^* = F_5(\mathcal{A}_{base})$.
    if $C_{phase} \leq N_{phase}$ **then**
      Set $c_{target} = F_1(C_{phase})$. Set $p_{target} = v^* - c_{target}$.
    end if
    for $l \in \{1, \ldots, L\}$ **do**
      Observe price p_t.
      $L_p = L_p \cup \{p_t\}$.
      Buy if $p_t \leq p_{target}$.
      Set $t = t + 1$.
    end for
    if $F_2(C_{phase}) = 1$ **then**
      Set $c_e = F_3(L_p)$. Set $L_{target} = L_{target} \cup \{c_e\}$.
    end if
    Set $C_{phase} = C_{phase} + 1$.
    if $F_6(C_{phase}) = 1$ **then**
      $p_{target} = F_4(L_{target})$
    end if
  end if
  if $D = 0$ **then**
    Observe price p_t.
    Choose to buy or not based on $\mathcal{A}_{base}$.
    Set $t = t + 1$.
  end if
end while

Remark 6.1. *In order to derive the results of Proposition 6.3 and 6.4, we only used the fact that the regret for $\mathcal{A}_{base}$ is bounded by $O(\sqrt{T\log T})$ or $O(T^{\frac{2}{3}}\sqrt{\log T})$. The same proof is also valid for any other base algorithm that satisfies these bounds. Also,*

the exact choices for functions F_1, F_2, F_3, F_4, F_5, F_6 do not effect the regret guarantee (in Section 6.5 we discuss these functions in more detail).

In which setting is BUYER-STRAT useful? As the seller does not know $\mathcal{D}$, it is reasonable to assume (as argued in Section 6.3) that the seller uses a low-regret algorithm to *learn* how to set prices. Note that many online learning algorithms (e.g. EXP3 and its variants) are *weight-based* algorithms: at round t, there are weights $w_{k,t}, \ldots, w_{K,t}$ and an action $k \in \{1, \ldots, K\}$ is chosen with probability $w_{k,t} / \sum_{k=1}^{K} w_{k,t}$. We call an algorithm a *pure weight-based* algorithm if in round t, only the weight of the selected action gets updated and if weights can only increase due to positive rewards (note that EXP3 is an example, see the Appendix for a general definition). Proposition 6.5 shows that, if the seller uses a pure weight-based algorithm, then BUYER-STRAT tends to encourage lower prices by using strategic cycles.

Proposition 6.5. *Assume that the buyer uses Algorithm 6.3, that the seller is using a pure weight-based algorithm and that the price set $\mathcal{P}$ is finite. Suppose that a strategic cycle runs from round $t+1$ to round $t+L$ with p_{target}, then $\mathbb{P}\{p_{t+L+1} \leq p_{target}\} \geq \mathbb{P}\{p_{t+1} \leq p_{target}\}$.*

Proof. The proof can be found in the Appendix. □

6.5 Experiments

In this section, we perform experiments in order to verify the theoretical results that were derived, and to investigate the effects of strategic behaviour on the regret in different scenarios.

6.5.1 Setup of experiments

In the experiments v_t is drawn from an uniform distribution on $[a-0.3, a+0.3]$, where a is drawn from an uniform distribution on $[0.4, 0.7]$ independently for each run. We consider two settings for the set of prices used by the seller and these are given by $\mathcal{P}_1$ and $\mathcal{P}_2$: $\mathcal{P}_1 = \{a + x \mid x \in \{-0.35, -0.3, -0.25, -0.2, -0.1, -0.05, -0.02, 0.0, 0.1, 0.3\}\}$, $\mathcal{P}_2 = \{a + x \mid x \in \{-0.05, -0.02, 0.0, 0.1, 0.3\}\}$. We will use the following abbreviations: P1 and P2. The abbreviation P1 means that $\mathcal{P}_1$ is used. The other abbreviations have a similar interpretation.

We consider three options for the seller pricing algorithm: (i) the seller chooses a price at random from the price set (RAND seller); (ii) the seller uses the low-regret learning algorithm EXP3.P (EXP3.P seller); (iii) the seller uses the full-information algorithm HEDGE (HEDGE seller). RAND seller is included because it models a situation where the buyer has no influence over the prices. EXP3.P seller is included because it is a bandit algorithm designed for adaptive adversaries and it enjoys high-probability regret bounds [15, 37]. It models a seller that is learning which prices

to use based on bandit feedback that is non-stochastic. HEDGE seller is included in order to investigate whether the restriction to bandit feedback has a major impact on the performance of BUYER-STRAT. HEDGE seller is tuned according to Remark 5.17 in [148] and EXP3.P according to Theorem 3.2 in [37].

In the experiments, BUYER-STRAT is tuned with $N_1 = \lceil T^{\frac{2}{3}} \log T \rceil$, $N_2 = \lceil 2\sqrt{T \log T} \rceil$, $L = \lfloor 25\sqrt{\log T} \rfloor$, $p_{cycle} = \frac{5}{\sqrt{T}}$, $c_1 = 0.1$. We set $N_{phase} = 4 \cdot N_3$, where $N_3 = \lceil 0.1 \cdot \sqrt{T} \rceil$. TS-NS is tuned with $N = \lceil 0.005 \cdot T^{\frac{2}{3}} \rceil$. We will refer to BUYER-STRAT with $\mathcal{A}_{base}$ chosen as UCB-NS, as UCB-S (Upper Confidence Bound Strategic). Similarly, We will refer to BUYER-STRAT with $\mathcal{A}_{base}$ chosen as TS-NS, as TS-S (Thompson sampling Strategic). The functions F_1, F_2, F_3, F_4, F_5, F_6 are chosen as follows.

$$F_1(x) = \begin{cases} 0.2 & \text{if } x \leq N_3 \\ 0.3 & \text{if } 1 \cdot N_3 < x \leq 2 \cdot N_3 \\ 0.4 & \text{if } 2 \cdot N_3 < x \leq 3 \cdot N_3 \\ 0.5 & \text{if } 3 \cdot N_3 < x \leq 4 \cdot N_3 \end{cases} \tag{6.1}$$

For $F_2(x)$ we take $F_2(x) = \mathbb{I}\{x \in \{N_3, 2 \cdot N_3, 3 \cdot N_3, 4 \cdot N_3\}\}$. The function $F_3(L_p)$ takes the last 100 elements added to the input list L_p and then calculates the 25-th percentile of these 100 values. The function $F_4(\cdot)$ is defined as $F_4(L_{target}) = \min\{L_{target}\} + \varepsilon$. The function $F_4(L_{target})$ takes the smallest number in the set L_{target} and adds a small value to it. In our experiments we use $\varepsilon = 0.005$. The function $F_5(\cdot)$ takes as input a base algorithm and returns the value of $\bar{v}_t$ in the base algorithm. For $F_6(x)$ we take $F_6(x) = \mathbb{I}\{x = 4 \cdot N_3\}$.

The intuition behind these choices is as follows. In every strategic cycle, only prices that satisfy $p_t \leq v^* - c$ are accepted, where $c \in \mathcal{C} = \{0.1, 0.2, 0.3, 0.4, 0.5\}$ and where c is chosen in increasing order (to try to reduce the price in stages) as the number of strategic cycles increases (this is specified by the function $F_1(\cdot)$). Initially, the number of strategic cycles in which every $c \in \mathcal{C}$ is used, is proportional to N_3. When $F_2(x) = 1$, this indicates that the last strategic cycle in which $c = x$ has just been completed, and the function $F_3(\cdot)$ is used to collect information about the price trajectory. When $F_6(x) = 1$, a final value for p_{target} is chosen (using $F_4(\cdot)$) and only prices with $p_t \leq p_{target}$ are accepted in all subsequent strategic cycles.

We perform 100 independent simulation runs in order to calculate our performance metrics. We use three performance metrics in order to evaluate our algorithm. In each run, we calculate the cumulative regret $R_T = \sum_{t=1}^{T} \max\{\nu - p_t, 0\} - \sum_{t=1}^{T} a_t \cdot (\nu - p_t)$, the cumulative utility $U_T = \sum_{t=1}^{T} a_t \cdot (\nu - p_t)$ and the scaled cumulative regret $R_T^S = R_T / \sum_{t=1}^{T} \max\{\nu - p_t, 0\}$. In the experiments we set $T \in \{25000, 50000, 75000, 100000, 200000, \ldots, 1000000\}$.

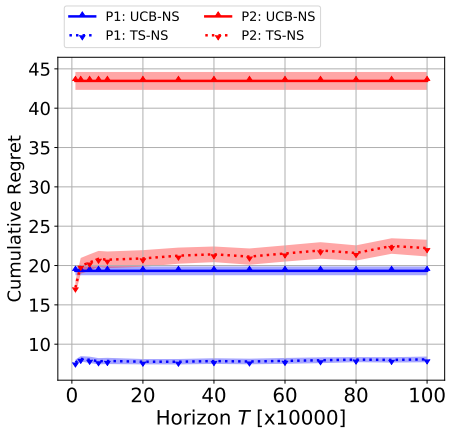
6.5.2 Results: non-strategic buyers vs. strategic buyers

Non-strategic buyers

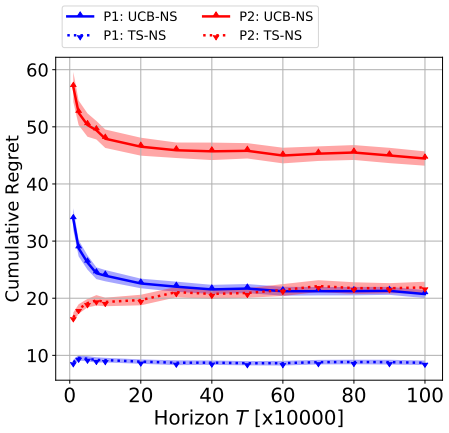
In Figures 6.1a and 6.1b the cumulative regret is shown for different experimental settings and different values for the problem horizon. Each point in the graph shows the cumulative regret over T rounds for a problem of horizon T averaged over 100 simulations. In all figures, the lines indicate the mean and the shaded region indicates a 95% confidence interval. The results indicate that the expected regret indeed grows as a sub-linear function of T and that this pattern holds for both RAND seller and EXP3.P seller. An interesting finding is that the regret for TS-NS is lower than UCB-NS: based on the theoretical analysis one would expect the opposite pattern. Figures 6.1e and 6.1f show the scaled cumulative regret and provides further evidence that the expected regret is a sub-linear function of the horizon T, as the curve shows a monotonically decreasing pattern. Figures 6.1c and 6.1d show the cumulative utility against different sellers. Here we observe that the utility tends to be higher if the seller uses $\mathcal{P}_1$, which makes intuitive sense as this price set contains lower prices.

Strategic buyers

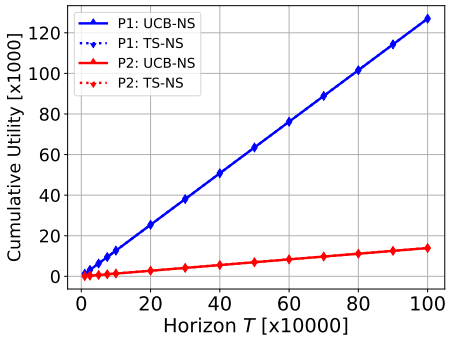
Figures 6.2a to 6.2f show the same performance metrics as for the non-strategic bidders. Figures 6.1a and 6.1b show that the level of the expected regret for strategic bidders is higher compared to the non-strategic bidders. Figures 6.2e and 6.2f again indicate that the expected regret is sub-linear in T, as the curves show a monotonically decreasing pattern (from Figure 6.2a it is hard to tell) . Thus, we observe sub-linear regret for both UCB-S and TS-S regardless of the seller algorithm and this is in line with the theoretical analysis. If we compare the cumulative utility in Figures 6.2c and 6.2d with those in Figures 6.1c and 6.1d, then we observe some interesting results. First, when strategic buyers are facing RAND seller (Figure 6.2c), then we see that the cumulative utility is about 70%-80% of the cumulative utility if non-strategic buyers are facing RAND seller (Figure 6.1c). Second, we see that if the seller is using EXP3.P (i.e, a low-regret learning algorithm), then the cumulative utility for strategic buyers is much higher compared to the cumulative utility for non-strategic buyers. In scenario P1 utilities are about 2.5-3 times higher and in scenario P2 utilities are about 2 times higher. The results for scenario P2 imply that, even when the lowest price is very close to the unknown mean valuation (absolute distance at most 0.05), it is still beneficial to act strategically. Additional experimental results when the seller uses EXP3.S [15] can be found in the Appendix.



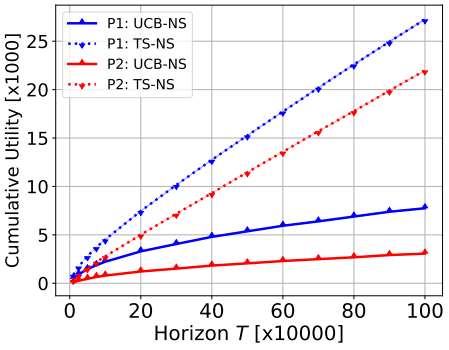
(a) R_T with RAND seller.



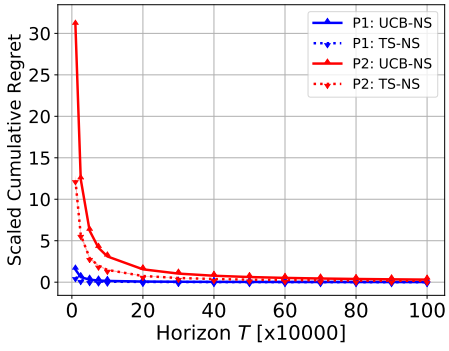
(b) R_T with EXP3.P seller.



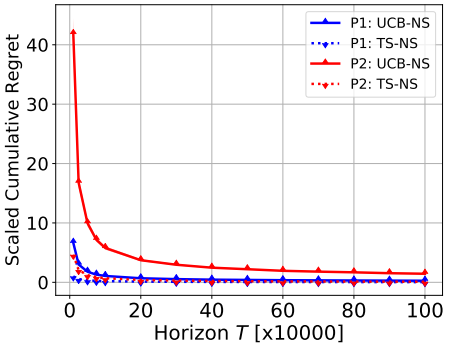
(c) U_T with RAND seller.



(d) U_T with EXP3.P seller.

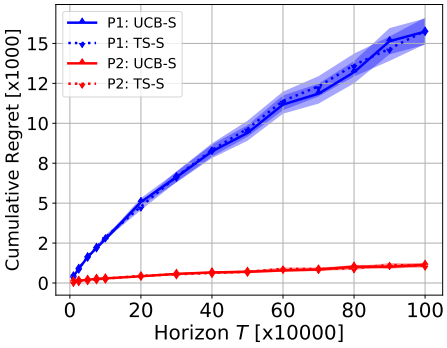


(e) R_T^S with RAND seller.

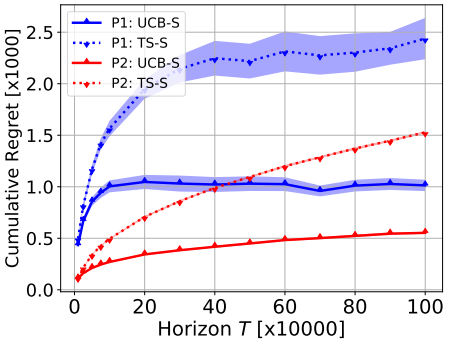


(f) R_T^S with EXP3.P seller.

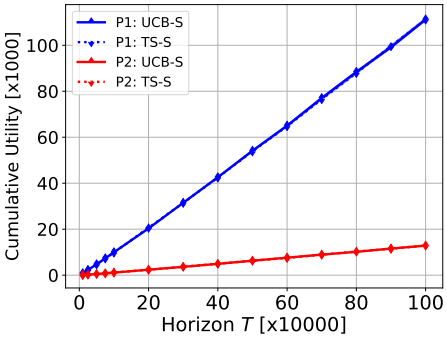
Figure 6.1: Results for non-strategic buyers with RAND seller and EXP3.P seller.



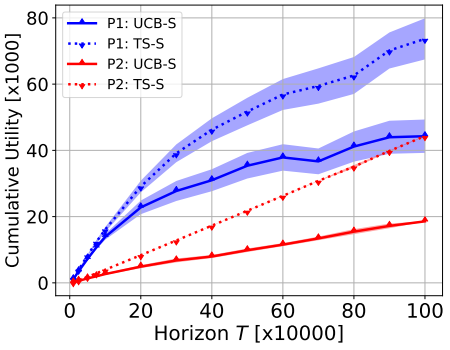
(a) R_T with RAND seller.



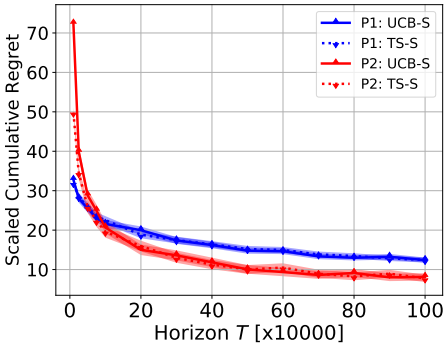
(b) R_T with EXP3.P seller.



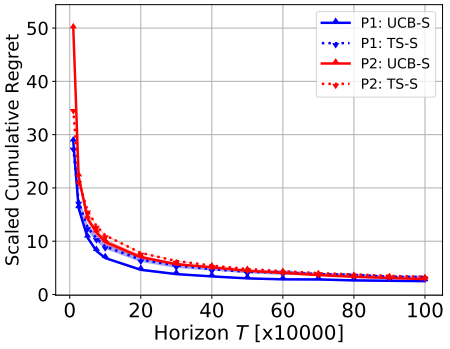
(c) U_T with RAND seller.



(d) U_T with EXP3.P seller.



(e) R_T^S with RAND seller.



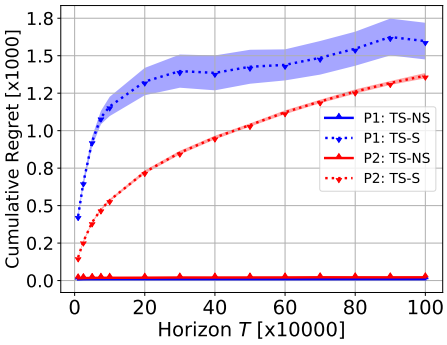
(f) R_T^S with EXP3.P seller.

Figure 6.2: Results for strategic buyers with RAND seller and EXP3.P seller.

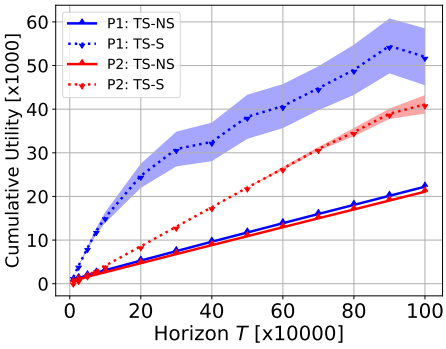
6.5.3 Explanation of differences

In order to study the impact of the quality of feedback that the seller observes, we give the seller full-information feedback instead of bandit feedback. More specifically, we assume the seller uses the algorithm HEDGE. Figures 6.3a-6.3c show results for TS-S and TS-NS against HEDGE seller. Even with full-information the results are qualitatively similar as before: the regret for the strategic buyers is sub-linear and cumulative utility is much higher for strategic buyers. Thus, the results indicate that the feedback type is not the main driver for the observed patterns.

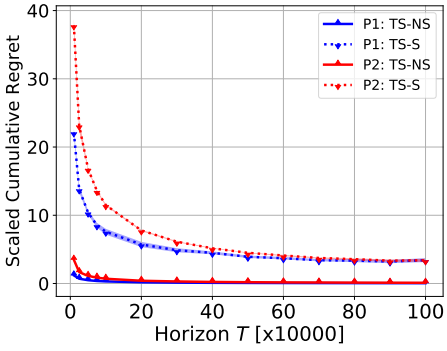
Figures 6.4a and 6.4b display the gap $\nu - p_t$ for a problem with horizon $T = 200000$ averaged over the 100 simulation runs. If the seller is using a low-regret algorithm in order to set prices and buyers are non-strategic, then we observe that prices tend to increase towards the mean valuation ν. This effect is stronger for HEDGE seller compared to EXP3.P seller and this is in line with expectations as HEDGE uses full-information feedback. Furthermore, we see a qualitatively similar pattern for the price sets $\mathcal{P}_1$ and $\mathcal{P}_2$, although the increase in price with $\mathcal{P}_2$ is slightly larger. For HEDGE seller, we hardly see any difference for different price sets. If buyers are strategic then we see the opposite pattern. The algorithms for strategic buyers tend to lower the price over time and the magnitude of this reduction depends on the price set of the seller (reduction for $\mathcal{P}_1$ is larger than for $\mathcal{P}_2$). However, even with price set $\mathcal{P}_2$ where the lowest prices are very close to ν, strategic behaviour is beneficial and strategic buyers can induce prices that are almost twice as far from ν.



(a) R_T with HEDGE seller.

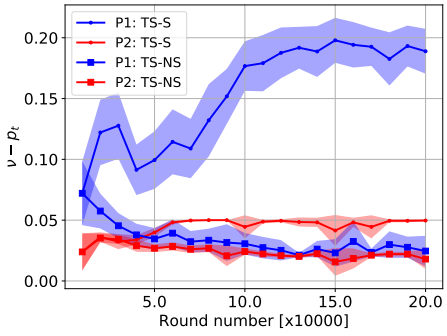


(b) U_T with HEDGE seller.

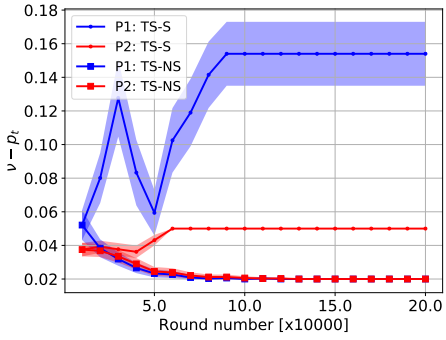


(c) R_T^S with HEDGE seller.

Figure 6.3: Results for strategic buyers with HEDGE seller.



(a) $\nu - p_t$ with EXP3.P seller.



(b) $\nu - p_t$ with HEDGE seller.

Figure 6.4: $\nu - p_t$ with EXP3.P seller and HEDGE seller.

6.6 Conclusion

This is Chapter we study repeated posted-price auctions with a single seller from the perspective of a utility maximizing buyer that does not know the distribution of his valuation. Previous work has only focused on the seller side and does not study how buyers should make decisions. Furthermore, in previous work it is assumed that the buyer knows his valuation in each round. Hence, in this Chapter, we address these gaps in the literature. We study two types of buyers (strategic and non-strategic) and derive sub-linear regret bounds that hold for all possible sequences of observed prices. Our algorithms are based on ideas from UCB-type bandit algorithms and Thompson Sampling. Our experiments show that, if the seller is using a low-regret learning algorithm based on weights updating, then strategic buyers can obtain much higher utilities compared to non-strategic buyers. Only when the prices of the seller are not related to the choices of the buyer, it is not beneficial to be strategic, but strategic buyers can still attain utilities of about 75% of the utility of non-strategic buyers.

In practice, buyers have limited budgets for purchasing items. However, the models and algorithms presented in this Chapter do not take budget constraints into account. Future work can be directed towards analyzing the repeated posted-price problem studied in this Chapter with budget constraints. In particular, it would be interesting to analyze how a budget constraint would affect the regret guarantees derived in this Chapter and whether budget constraints make it easier or harder to engage in strategic behavior.

Appendix

Section 6.A contains proofs that are omitted from the main text. Section 6.B presents some additional experimental results.

6.A Proofs for Section 6.4

This part contains the proofs of the Propositions are omitted from the main text. For convenience, we restate the Propositions here.

6.A.1 Proof of Proposition 6.5

In this section we give a proof of Proposition 6.5. We first give a definition of a pure weight-based algorithm.

Definition 6.1. *Let there be K actions in total and let $\mathcal{K} = \{1, \ldots, K\}$. Let $w_{k,t} \in \mathbb{R}$ denote the weight of action k at the beginning of round t. Suppose that action j is selected in round t and that the observed reward for action j in round t equals $r_{j,t}$.*

Let $\hat{p}_{k,t}$ denote the probability that action k is selected in round t. An algorithm $\mathcal{A}$ is called a pure weight-based algorithm if the following conditions are satisfied:

1. *if $r_{j,t} > 0$, then $w_{j,t+1} > w_{j,t}$.*
2. *if $r_{j,t} = 0$, then $w_{j,t+1} = w_{j,t}$.*
3. *if $k \neq j$, then $w_{k,t+1} = w_{k,t}$.*
4. *$\sum_{k\in\mathcal{K}^*} \hat{p}_{k,t} = F(\sum_{k\in\mathcal{K}^*} w_{k,t}/\sum_{k=1}^{K} w_{k,t})$ for all subsets $\mathcal{K}^* \subseteq \mathcal{K}$, where $F(\cdot)$ is an increasing function. That is, for all subsets $\mathcal{K}^* \subseteq \mathcal{K}$, if $a = \sum_{k\in\mathcal{K}^*} w_{k,t}/\sum_{k=1}^{K} w_{k,t}$, $b = \sum_{k\in\mathcal{K}^*} w'_{k,t}/\sum_{k=1}^{K} w'_{k,t}$ and $a > b$, then $F(a) > F(b)$.*

Note that if $\hat{p}_{k,t} = w_{k,t}/\sum_{k=1}^{K} w_{k,t}$ then condition 4 in Definition 6.1 is satisfied. Also note that EXP3 of [15] uses $\hat{p}_{k,t} = (1-\gamma)w_{k,t}/\sum_{k=1}^{K} w_{k,t} + \gamma/K$ for some $0 < \gamma < 1$ and this choice also satisfies condition 4 in Definition 6.1.

Proposition 6.5. *Assume that the buyer uses Algorithm 6.3, that the seller is using a pure weight-based algorithm and that the price set $\mathcal{P}$ is finite. Suppose that a strategic cycle runs from round $t+1$ to round $t+L$ with p_{target}, then $\mathbb{P}\{p_{t+L+1} \leq p_{target}\} \geq \mathbb{P}\{p_{t+1} \leq p_{target}\}$.*

Proof. Let $|\mathcal{P}| = K$, $\mathcal{K} = \{1, \ldots, K\}$, $p_{max} = \max\{\mathcal{P}\}$ and $p_{min} = \min\{\mathcal{P}\}$. Assume, without loss of generality, that $\mathcal{P} = \{p^1, \ldots, p^K\}$ and that $0 < p_{min} = p^1 \leq p^2 \leq \cdots \leq p^{K-1} \leq p^K = p_{max}$. Let $\hat{\mathcal{P}} = \{p \in \mathcal{P} |\ p \leq p_{target}\}$. Let $\bar{\mathcal{P}} = \{p \in \mathcal{P} |\ p > p_{target}\}$. Let $\hat{\mathcal{K}} = \{k \in \mathcal{K} |\ p^k \in \hat{\mathcal{P}}\}$. Let $\bar{\mathcal{K}} = \{k \in \mathcal{K} |\ p^k \in \bar{\mathcal{P}}\}$. Let $w_{k,t}$ denote the weight of action k at the beginning of round t.

We now proceed to prove the statement in the Proposition. We prove the Proposition for $L = 1$. The case for general L follows by repeatedly applying the result for $L = 1$.

We distinguish the following cases. Case 1: $p_{target} \geq p_{max}$. Case 2: $p_{target} < p_{min}$. Case 3: $p_{min} \leq p_{target} < p_{max}$.

- Case 1: $p_{target} \geq p_{max}$. In this case, $p \leq p_{target}$ for all $p \in \mathcal{P}$. Therefore, $\mathbb{P}\{p_{t+1} \leq p_{target}\} = 1$ and $\mathbb{P}\{p_{t+2} \leq p_{target}\} = 1$ and the statement in the Proposition holds.
- Case 2: $p_{target} < p_{min}$. In this case, $p > p_{target}$ for all $p \in \mathcal{P}$. Therefore, $\mathbb{P}\{p_{t+1} \leq p_{target}\} = 0$ and $\mathbb{P}\{p_{t+2} \leq p_{target}\} = 0$ and the statement in the Proposition holds.
- Case 3: $p_{min} \leq p_{target} < p_{max}$. There are 2 subcases to consider. Case A: $p_{t+1} > p_{target}$ and Case B: $p_{t+1} \leq p_{target}$.

– In Case A, none of the weights get updated. This is true because none of the prices in $\hat{\mathcal{P}}$ are selected since $p_{t+1} > p_{target}$. By condition 3 in Definition 6.1, it follows that none of the weights corresponding to the prices in $\hat{\mathcal{P}}$ will get updated.

  Also, none of the prices in $\bar{\mathcal{P}}$ will get a positive reward because they will all be rejected by the buyer. By condition 2 in Definition 6.1, it follows that none of the weights corresponding to the prices in $\bar{\mathcal{P}}$ will get updated. As none of the weights will get updated after round $t+1$ is completed, we have that $\mathbb{P}\{p_{t+2} \leq p_{target}\} = \mathbb{P}\{p_{t+1} \leq p_{target}\}$. So we conclude that the statement in the Proposition holds.

– In Case B, there exists a $j \in \{1, \ldots, K\}$ such that $p_{t+1} = p^j$ and the reward for action j satisfies $r_{j,t+1} > 0$. This is true because the price $p_{t+1} = p^j$ will be accepted by the buyer and the reward $r_{j,t+1}$ equals the price p^j which (by assumption) satisfies $p^j \geq p_{min} > 0$.

  By condition 1 in Definition 6.1, it follows that $w_{j,t+2} > w_{j,t+1}$. By condition 3 in Definition 6.1, it follows that $w_{k,t+2} = w_{k,t+1}$ for all $k \neq j$, since these prices/actions are not selected in round $t+1$.

  This yields the following:

$$\sum_{k \in \hat{\mathcal{K}}} w_{k,t+2} > \sum_{k \in \hat{\mathcal{K}}} w_{k,t+1} \tag{6.2}$$

$$\sum_{k \in \bar{\mathcal{K}}} w_{k,t+2} = \sum_{k \in \bar{\mathcal{K}}} w_{k,t+1} \tag{6.3}$$

$$\sum_{k=1}^{K} w_{k,t+2} > \sum_{k=1}^{K} w_{k,t+1} \tag{6.4}$$

  Note that we also have:

$$\mathbb{P}\{p_{t+2} \leq p_{target}\} = 1 - \mathbb{P}\{p_{t+2} > p_{target}\}, \tag{6.5}$$

$$\mathbb{P}\{p_{t+1} \leq p_{target}\} = 1 - \mathbb{P}\{p_{t+1} > p_{target}\}. \tag{6.6}$$

  By combining (6.3) and (6.4), and by condition 4 in Definition 6.1, we obtain that $\mathbb{P}\{p_{t+2} > p_{target}\} < \mathbb{P}\{p_{t+1} > p_{target}\}$. As a consequence, by using (6.5) and (6.6), it follows that $\mathbb{P}\{p_{t+2} \leq p_{target}\} > \mathbb{P}\{p_{t+1} \leq p_{target}\}$. So we conclude that the statement in the Proposition holds.

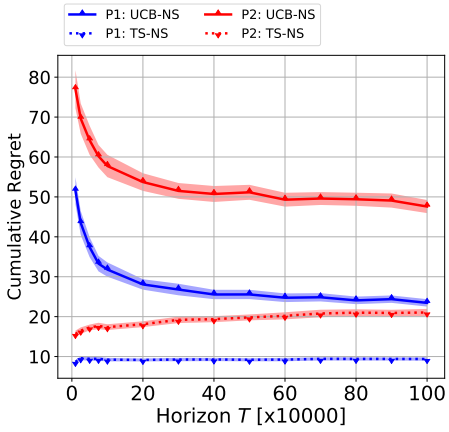
The case for general L follows from repeatedly applying the above argument. Note that the argument above works every initial weight vector. By repeatedly applying the above argument, one can show that $\mathbb{P}\{p_{t+1} \leq p_{target}\} \leq \mathbb{P}\{p_{t+2} \leq p_{target}\} \leq \cdots \leq \mathbb{P}\{p_{t+L} \leq p_{target}\} \leq \mathbb{P}\{p_{t+L+1} \leq p_{target}\}$. This concludes the proof. □

6.B Additional experiments

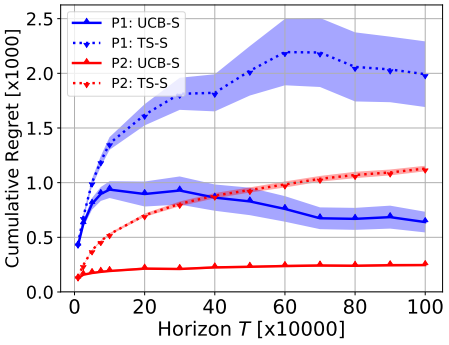
This part contains additional results related to the experiments in the main text. We show results for non-strategic and strategic buyers against another (more powerful) seller algorithm. We assume the seller uses the EXP3.S algorithm from [15]. We will refer to this as EXP3.S Seller. This algorithm has sub-linear regret with respect to action sequences with at most S switches. EXP3.S Seller is tuned according to Corollary 8.2 in [15].

Figures 6.5a, 6.5c and 6.5e display the results for non-strategic buyers and Figures 6.5b, 6.5d and 6.5f display the results for strategic buyers. In all figures, the lines indicate the mean and the shaded region indicates a 95% confidence interval. The results are qualitatively similar to those reported in the main text. The results indicate that the proposed algorithms for strategic and non-strategic buyers have sub-linear regret in all cases considered.

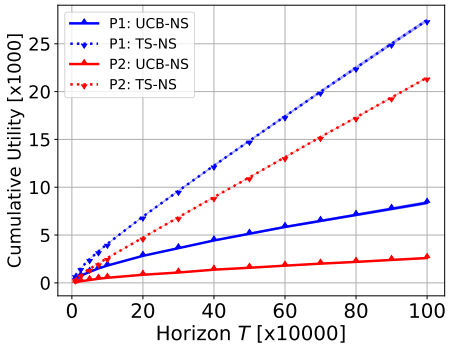
In scenario P1 utilities are about 2.0-2.5 times higher. In scenario P2 the differences are smaller, which is in line with expectations since the lowest price of the seller is very close to the unknown mean valuation. In general, the strategic buyers tend have higher utilities.



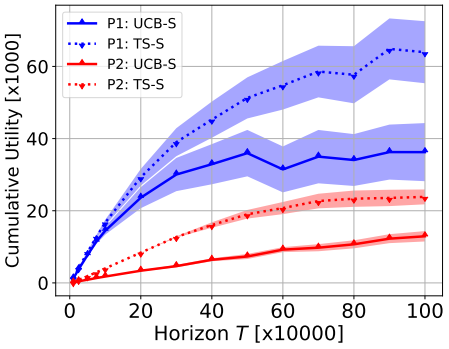
(a) R_T with EXP3.S seller.



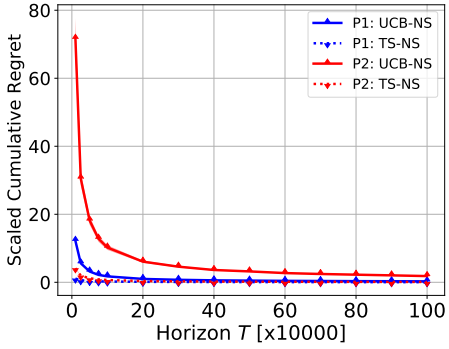
(b) R_T with EXP3.S seller.



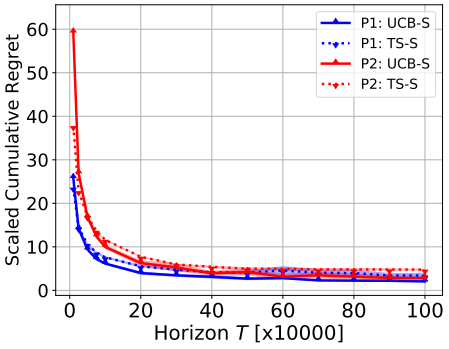
(c) U_T with EXP3.S seller.



(d) U_T with EXP3.S seller.



(e) R_T^S with EXP3.S seller.



(f) R_T^S with EXP3.S seller.

Figure 6.5: Results for non-strategic (left column) and strategic (right column) buyers with EXP3.s seller.

Chapter 7

Dynamic pricing with limited price changes and censored demand*

Chapter 7 moves away from decisions in auction settings and instead studies a more general dynamic pricing problem. More specifically, we consider a seller that has to adjust his selling prices and inventory levels over time in order to maximize his expected revenue. In addition, the seller faces a business constraint on the number of price changes allowed during the selling horizon and the seller only has a limited amount of inventories on hand in each selling period.

---

*This chapter is based on Rhuggenaath et al. [140].

7.1 Introduction

In this Chapter, we consider a dynamic pricing and learning problem with demand censoring and limited price changes. The typical approach when tackling dynamic pricing problems with uncertain demand in revenue management, is to conduct price experimentation. By using price experimentation the seller of a product can learn the optimal price to charge for his product. Many e-commerce companies have the ability to change prices at little costs, but frequent price changes are not always desirable. For example in [55], the authors note that Groupon (a large e-commerce marketplace) does not approve of charging different prices to each arriving customer because it may confuse customers and lead to negative customer feedback.

In practice, companies also have limited inventories on hand during a selling period. Due to limited inventories, the observed sales are not equal to demand anymore. As a consequence the true underlying demand is only partially observed, that is, demand is *censored*.

In this Chapter, we consider a seller that faces demand uncertainty and has to adjust his selling price over the sales horizon in order to learn the optimal price and maximize his cumulative revenue over the sales horizon. The seller faces a business constraint on the number of price changes allowed during the selling horizon and the seller only has a limited (finite) amount of inventories on hand in each selling period. The goal of the seller is to design a pricing and inventory policy that has low *regret*, defined as the gap between the revenue of a clairvoyant who has full information on the demand function and the revenue achieved by a seller facing unknown demand. Demand censoring has three important implications for a seller that sets prices based on observed sales data. First, it makes demand learning harder since he does not observe true demand at a particular price. Second, the seller needs to optimize the inventory level as well, since it affects (via demand censoring) the observed sales. Third, demand censoring can lead to *lost sales* since some potential demand is not satisfied and this leads to lost revenues for the seller.

In this Chapter, we propose a heuristic policy called HPI-LPC-CD (heuristic pricing and inventory policy with limited price changes and censored demand) for the aforementioned problem of the seller. We summarize the main contributions of this Chapter as follows:

- We study a dynamic pricing problem with limited price changes and censored demand. In contrast with previous work, we do not assume that we can observe true demand and lost sales. Furthermore, in our setting, the lost sales is part of the objective function that the seller aims to optimize.
- We propose a pricing policy that adjust prices and inventory levels for this problem.
- We conduct numerical experiments in order to test the performance of our policy. Experimental results are promising and show that the growth rate of regret is

sub-linear with respect to the sales horizon.

The remainder of this Chapter is organized as follows. In Section 7.2 we discuss the related literature. Section 7.3 provides a formal formulation of the problem. In Section 7.4 we present our proposed policy. In Section 7.5 we perform experiments in order to assess the performance of our policy. Section 7.6 concludes this Chapter and provides some interesting directions for further research.

7.2 Related Literature

The work in this Chapter is related to studies about dynamic pricing and learning with demand uncertainty, newsvendor problems, pricing with limited inventories, and pricing with limited price changes. Dynamic pricing and learning with demand uncertainty has received increasing attention in the recent years, see [63] for an extensive review. In the general dynamic pricing problem with demand learning, there is a seller that needs to decide on the optimal selling price to charge for a product. The seller, however, does not know the precise relationship between price and demand. Most of the literature (e.g. [28, 36, 65, 66, 98, 105]) assumes that there is some functional form that relates prices and demand, but that the parameters of this model are unknown, and hence, needs to be estimated from sales data. This unknown relationship between price and demand, and the objective of revenue maximization gives rise to the so-called *exploration-exploitation* trade-off. In order to learn the demand at various prices the seller needs to use price experimentation (exploration), but due to this price experimentation the seller is not setting the optimal price in each period and this experimentation comes at the cost of revenue maximization (exploitation).

The studies that are most relevant for the work presented in this Chapter are [45, 48, 49, 55]. In [55] the authors study a dynamic pricing problem with limited price changes but there is no demand censoring and no inventory decisions. The authors present a policy for this problem and characterize the regret as a function of the number of price changes. In [48], a multi-period stochastic inventory system with backlogs and demand uncertainty is considered. Although [48] do consider a setting with limited price changes, they make the assumption that potential demand (and thus the lost sales) is observed and can be backlogged. In this Chapter we do not make such an assumption. In [45, 49] a problem similar to the one in this Chapter is considered, but in [45, 49] there are no restrictions on the number of price changes. We note that demand censoring has also been studied in the setting of (repeated) newsvendor and stochastic inventory problems (e.g. [27, 92, 93]) but in these studies the demand is assumed to be stationary and there is no pricing component.

7.3 Problem Formulation

In this section we give a formal description of the problem statement. The model considered here is based on the demand assumptions made in [55], but is extended to include a stochastic inventory control component similar to [45, 49, 92, 93].

Remark 7.1. *We will use* $(x)^+$ *to denote* $\max\{x, 0\}$. *We use* $\lceil x \rceil$ *to denote the ceiling function applied to* x. *We will occasionally use the symbol* $x \wedge y$ *to denote* $\min\{x, y\}$ *and the symbol* $\vee$ *to denote* $\max\{x, y\}$.

We consider a monopolist seller that sells a single nonperishable product over time horizon of $T \in \mathbb{N}$ periods. At the beginning of each period the seller can order products to adjust the on hand inventory. Furthermore, at the beginning of each period the seller has to decide on a price $p_t \in \mathcal{P} = [p^L, p^H]$. The prices $p^L < p^H$ are the minimum and maximum prices that are acceptable to the seller. Demand is stochastic and the seller does not know the true underlying demand model. Demand is also *unobserved* due to censoring, because we assume that only the sales are observed. That is, if demand exceeds the available inventory in period t, then the sales equals the inventory in that period. The objective of the seller is to set prices and inventories in order to maximize revenues over the sales horizon.

7.3.1 Demand assumptions

In each period t the demand for the product is given by $D(p) = \lambda(p) + \epsilon_t$. Here $\lambda(p) = \mathbb{E}\{D(p)\}$ is the expected demand at price p and $\epsilon_t, t = 1, \dots T$ are identically and independently distributed (i.i.d) random variables with $\mathbb{E}\{\epsilon_t\} = 0$ and with cumulative distribution function (CDF) given by $F(\cdot)$. We assume that ϵ_t is bounded with known bounds $[l, u]$. The function $\lambda(p)$ gives the expected demand at price p and we assume that $\lambda(\cdot)$ is a non-increasing function. The function $\lambda(\cdot)$ and the distribution of ϵ_t is unknown to the seller and the seller has to learn these during the sales horizon.

Although the demand model is unknown to the seller, we assume that the seller does have some prior knowledge about demand. In particular, we assume that the seller has a finite hypothesis set $\Lambda = \{\lambda_1(p), \dots, \lambda_K(p)\}$ consisting of $|\Lambda| = K$ mean demand functions. The demand according to hypothesis $\lambda_k(p) \in \Lambda$ is given by $D(p) = \lambda_k(p) + \epsilon_t$. We assume that the true mean demand function is an element of the set Λ. In order to distinguish between the different demand models in the hypothesis set, we make the assumption that a set of *discriminative* prices $\mathcal{P}^D$ is available. A price $p_D \in \mathcal{P}^D$ is called discriminative if $\lambda_k(p_D) \neq \lambda_j(p_D)$ for all $\lambda_k(\cdot), \lambda_j(\cdot) \in \Lambda$. That is, a price is called discriminative if the mean demand at that price is different for all demand functions in Λ. The assumptions on demand that are made in this section are similar to the assumptions made in [55]. In the next subsection we discuss the assumptions made on the inventory.

7.3.2 Inventory assumptions and dynamics of the system

Let x_t and y_t denote the inventory levels at the beginning of period t before and after an inventory replenishment decision, respectively. We assume that the system is initially empty, i.e., $x_1 = 0$. We assume that inventory lies in a bounded interval, that is, $y_t \in \mathcal{Y} = [y^L, y^H]$ with $y^H \geq \lambda(0) + u$. Under these assumptions there is no demand censoring if $y_t = y^H$.

An admissible or feasible policy is represented by a sequence of prices and order-up-to levels, $\{(p_t, y_t), t \geq 1\}$ with $y_t \geq x_t$, where (p_t, y_t) depends only on the demand and decisions made prior to time t, that is, (p_t, y_t) is adapted to the filtration generated by $\{(p_t, y_t), t \geq 1\}$ under censored demand.

Given an admissible policy π, the following sequence of events occurs in each period t:

- At the beginning of period t the seller observes the current inventory level x_t.
- The seller decides to increase the inventory level to $y_t \in \mathcal{Y}$ and decides on the price p_t that will be charged in period t. Similar to previous studies (e.g. [48, 92, 93]), we assume that replenishment occurs instantly without any delay.
- The demand during period t, denoted by $d_t(p_t)$ is realized and the seller tries to satisfy as much of this demand as possible using the inventory available during the period.
- Demand that is not satisfied is *lost* and *unobservable*. More specifically, the seller only observes the sales $\min\{d_t, y_t\}$ during period t.
- At the end of period t, the seller incurs a profit given by:

$$\begin{aligned} R_t(p_t, y_t) &= p_t \cdot \min\{d_t, y_t\} - b \cdot (d_t - y_t)^+ \\ &\quad - h \cdot (y_t - d_t)^+ \\ &= p_t \cdot d_t - (b + p_t) \cdot (d_t - y_t)^+ \\ &\quad - h \cdot (y_t - d_t)^+ \end{aligned} \tag{7.1}$$

  Here b is a parameter that represents the costs due to lost sales and h represents the holding costs due to inventory that is left over at the end of period t. Similar to previous studies (e.g. [48, 92, 93]), ordering costs are assumed to be zero. Note that the profit given by Equation (7.1) is unobserved since it depends on the unobserved realized demand $d_t(p_t)$.

7.3.3 Objective function under full information

The objective of the publisher is to maximize the cumulative revenue over the sales horizon of length T:

$$\max_{(p_t,y_t)\in\mathcal{P}\times\mathcal{Y},y_t\geq x_t} \mathbb{E}\left\{\sum_{t=1}^{T} R_t(p_t,y_t)\right\}. \tag{7.2}$$

Note that if $\lambda(p)$ and the distribution of ϵ_t was known and the seller could observe lost sales, then the optimal policy can be found by solving the following optimization problem:

$$\begin{aligned}\max_{(p_t,y_t)\in\mathcal{P}\times\mathcal{Y},y_t\geq x_t} & \sum_{t=1}^{T} p_t\cdot\mathbb{E}\left\{D_t(p_t)\right\}\\ & -\sum_{t=1}^{T}(b+p_t)\cdot\mathbb{E}\left\{(D_t(p_t)-y_t)^+\right\}\\ & -\sum_{t=1}^{T} h\cdot\mathbb{E}\left\{(y_t-D_t(p_t))^+\right\}\end{aligned} \tag{7.3}$$

However, in our setting the seller knows neither the function $\lambda(p)$ nor the distribution of ϵ_t and cannot observe lost-sales.

Suppose that the seller knows the function $\lambda(p)$ and the distribution of ϵ_t. In this case it has been shown [149] that a myopic policy is optimal. We can define the single-period profit function as follows:

$$\begin{aligned}Q(p,y) = p\cdot\mathbb{E}\left\{D_t(p)\right\} & -(b+p)\cdot\mathbb{E}\left\{(D_t(p)-y)^+\right\}\\ & -h\cdot\mathbb{E}\left\{(y-D_t(p))^+\right\}\end{aligned} \tag{7.4}$$

To find the optimal pricing and inventory decision we thus need to optimize $Q(p,y)$. The full information optimization problem (FI-OPT) that assumes that the seller knows the function $\lambda(p)$ and the distribution of ϵ_t, can be written more compactly using (7.5) and (7.6):

$$\begin{aligned}Z(p,\lambda(p)) = -\min_{y\in\mathcal{Y}}\Big\{(b+p)\cdot\mathbb{E}\left\{(\lambda(p)+\epsilon-y)^+\right\}\\ -h\cdot\mathbb{E}\left\{(y-\lambda(p)-\epsilon)^+\right\}\Big\}+(p\cdot\lambda(p))\end{aligned} \tag{7.5}$$

$$\max_{p\in\mathcal{P},y\in\mathcal{Y}} Q(p,y) = \max_{p\in\mathcal{P}}\left\{Z(p,\lambda(p))\right\} \tag{7.6}$$

7.3.4 Regret

Let the optimal solution to FI-OPT be denoted by (p^*, y^*) and the optimal single-period profit by $Q(p^*, y^*)$. The *regret* of an admissible policy π that generates $\{(p_t, y_t), t \geq 1\}$ can now be defined as follows

$$\mathcal{R}(\pi, T) = T \cdot Q(p^*, y^*) - \mathbb{E}\left\{\sum_{t=1}^{T} R_t(p_t, y_t)\right\}. \tag{7.7}$$

The regret measures the expected difference in revenue that arises from the fact that the seller is using policy π instead of the optimal policy that uses (p^*, y^*) in each period. Note that minimizing (7.7) is equivalent to minimizing the per period regret given by:

$$\mathcal{R}^P(\pi, T) = Q(p^*, y^*) - \frac{1}{T}\mathbb{E}\left\{\sum_{t=1}^{T} R_t(p_t, y_t)\right\}. \tag{7.8}$$

7.4 Proposed policy

In this section we discuss our proposed pricing and inventory policy. First we define some preliminary notation and concepts that are needed for our policy.

7.4.1 Preliminaries

Define for each $\lambda_k(p) \in \Lambda$ the following counterpart to FI-OPT, which we denote by k-OPT:

$$\max_{p \in \mathcal{P}, y \in \mathcal{Y}} Q_k(p, y) = \max_{p \in \mathcal{P}} \{Z(p, \lambda_k(p))\} \tag{7.9}$$

Here $Q_k(p, y)$ represents the expected single-period profit (with respect to the distribution of ϵ_t) when using the mean demand function $\lambda_k(p) \in \Lambda$ instead of the true mean demand function $\lambda(p)$.

Next, we define a counterpart to k-OPT that uses samples drawn from the distribution of ϵ. Suppose that we have access to M samples $\{\hat{\epsilon}_t, t = 1, \ldots, M\}$ from the distribution of ϵ. In that case we can define the sampled version of $Q_k(p, y)$, which we denote by $\hat{Q}_k^{SAM}(p, y)$. Using $\hat{Q}_k^{SAM}(p, y)$ we define a sampled version of the optimization problem k-OPT, denoted by k-SAM:

$$\max_{p \in \mathcal{P}, y \in \mathcal{Y}} \hat{Q}_k^{SAM}(p, y) = \max_{p \in \mathcal{P}} \hat{Z}(p, \lambda_k(p)) \tag{7.10}$$

$$\hat{Z}(p, \lambda_k(p)) = -\min_{y \in \mathcal{Y}} \left\{ \frac{1}{M} \sum_{t=1}^{M} (b+p) \cdot (\lambda_k(p) + \hat{\epsilon}_t - y)^+ - h \cdot (y - \lambda_k(p) - \hat{\epsilon}_t)^+ \right\} + (p \cdot \lambda_k(p)) \tag{7.11}$$

Note that, in general, the true demand is not observed due to possible censoring. Since the true demand is not observed due to censoring, it is useful to define a counterpart to k-SAM that is not based on samples from the distribution of ϵ. Suppose that we have access to M samples $\{\hat{\eta}_t, t = 1, \ldots, M\}$ that *are not necessarily* from the distribution ϵ. We define another sampled version of the optimization problem k-OPT, denoted by k-APPROX:

$$\max_{p \in \mathcal{P}, y \in \mathcal{Y}} \hat{Q}_k^{APPROX}(p, y) = \max_{p \in \mathcal{P}} \bar{Z}(p, \lambda_k(p)) \tag{7.12}$$

$$\bar{Z}(p, \lambda_k(p)) = -\min_{y \in \mathcal{Y}} \left\{ \frac{1}{M} \sum_{t=1}^{M} (b+p) \cdot (\lambda_k(p) + \hat{\eta}_t - y)^+ - h \cdot (y - \lambda_k(p) - \hat{\eta}_t)^+ \right\} + (p \cdot \lambda_k(p)) \tag{7.13}$$

7.4.2 Heuristic policy

After having defined some preliminary concepts in the previous subsection, we now proceed to present our heuristic policy. The full procedure for the heuristic policy is described in Algorithm 7.1. The policy takes the following parameters as input:

1. The sales horizon T and an upperbound $m \in \mathbb{N}$ on the number of price changes that are allowed.
2. The price set $\mathcal{P} = [p^L, p^H]$ and the order-to-levels $\mathcal{Y} = [y^L, y^H]$.
3. An initialization parameter $y_{start} \in \mathcal{Y}$.
4. An initialization parameter $p_{start} \in \mathcal{P}$.
5. A fixed parameter $0 < v < 1$.
6. A set of demand functions $\Lambda = \{\lambda_1(p), \ldots, \lambda_K(p)\}$.
7. A set of discriminative prices $\mathcal{P}^D$.
8. A constant C_T that only depends on T.

The main idea of the algorithm behind our policy is to split the sales horizon into a number of phases. More specifically, the algorithm splits the sales horizon in $m+1$ phases. For each $0 \leq \ell \leq m$, a single price P_ℓ^* is offered through phase ℓ, which starts

at period $\tau_\ell + 1$ and ends at period $\tau_{\ell+1}$. We call phases 0 to $m-1$ learning phases and phase m is called the earning phase.

Except for a constant factor, the lengths of the phases are iterated-exponentially increasing. Suppose we are in phase ℓ and that it runs from period t_1 to period t_2. In our policy we have that $t_1 = \tau_\ell + 1$ and $t_2 = \tau_{\ell+1} = \tau_\ell + C_T \lceil \log^{(m-\ell)} T \rceil$ for some constant C_T that only depends on T. Here $\log^{(m)} T$ denotes m iterations of the (natural) logarithm. We define $\log^{(m)} T = 0$ if $m > m^*$ where m^* is the smallest integer such that $0 < \log^{(m^*)} T \leq 1$. This setup of partitioning the sales horizon is similar to the approach used in [55].

The inventory decisions are made according to the following rule:

$$y_t = \begin{cases} y_{start}, & \text{if } t = 1 \\ y_{t-1}, & \text{if } y_{t-1} > d_{t-1}, t > 1 \\ \min\{y_{t-1} \cdot (1+v), y^H\}, & \text{otherwise} \end{cases} \tag{7.14}$$

The main idea behind (7.14) is to adjust the order-up-to level upwards by a fixed factor v, if a stock-out occurs. If no stock-out occurs, then we keep the order-up-to level the same as in the previous period. We note that a similar rule was also used by [93] in a stochastic inventory problem with no pricing decisions and by [45, 49] in a pricing problem with inventory decisions and unlimited price changes.

At the end of learning phase ℓ the policy computes the sample mean of the observed sales under price P_ℓ^*. Since P_ℓ^* is a discriminative price, the seller gains information about the identity of the true demand function in this learning phase. In particular, the seller compares the observed mean sales with the mean demand at the price P_ℓ^* for each demand model in the hypothesis set Λ (Line 15). The demand model k that has a mean demand (at price P_ℓ^*) that is the closest to the observed mean sales, is then selected to generate samples of the error-term of the demand model (Line 15 and 16). The intuition behind this procedure is that the mean demand (at price P_ℓ^*) under the true demand model should be close to the mean of the observed sales if the effect of demand censoring is not too severe. The update rule for the inventory decisions given by (7.14) ensures that the observed sales does not suffer too much demand censoring. The seller subsequently (Line 17) determines the optimal solution (p^*, y^*) to the optimization problem given by (7.12) and (7.13). If we are not yet in the last learning phase, we select the next price to be equal to the discriminative price that is the closest to p^* (Line 19). The next value for the order-up-to level is then optimized conditional on the discriminative price that is closest to p^* (Line 20). If we are at the end of the last learning phase, then the price p^* will be used in all subsequent periods. The order-up-to level for all subsequent periods is then equal to y^*.

Algorithm 7.1 Heuristic Policy: HPI-LPC-CD

Require: v, m, p_{start}, T, y_{start}, C_T, $\mathcal{P}^D$, $\mathcal{P}$, $\mathcal{Y}$.
Set $\ell = 0$.
Set $\tau_\ell = 0$.
Set $P_\ell^* = p_{start}$.
Set $y_1 = y_{start}$.
for $\ell = 0$ **to** $m-1$ **do**
  Set $\tau_{\ell+1} = \tau_\ell + C_T \lceil \log^{(m-\ell)} T \rceil$.
  Set $t_s = \tau_\ell + 1$.
  Set $t_e = \tau_{\ell+1}$.
  if $t_e > t_s$ **then**
    for $t = t_s$ **to** t_e **do**
      Set inventory equal to y_t using (7.14).
      Set $p_t = P_\ell^*$.
    end for
    At the end of period t_e compute $\bar{X}_\ell = \sum_{t=t_s}^{t_e} \frac{\min\{d_t(P_\ell^*), y_t\}}{t_e - t_s}$.
    Compute the index $i_\ell = \text{argmin}_{i \in \{1,\ldots,K\}} |\bar{X}_\ell - \lambda_i(P_\ell^*)|$.
    Set $k = i_\ell$, $\hat{\eta}_t = \min\{d_t(P_\ell^*), y_t\} - \lambda_k(P_\ell^*)$.
    Set $(p^*, y^*) = \text{argmax}_{p \in \mathcal{P}, y \in \mathcal{Y}} \hat{Q}_k^{APPROX}(p, y)$ using (7.12) and (7.13).
    if $\ell < m-1$ **then**
      Set $P_{\ell+1}^*$ as the price in $\mathcal{P}^D$ that is closest to p^*.
      Set $y_{t+1} = \bar{Z}(P_{\ell+1}^*, \lambda_k(P_{\ell+1}^*))$.
    end if
    if $\ell = m-1$ **then**
      Set $P_{\ell+1}^* = p^*$.
      $\ell = \ell + 1$.
    end if
  else
    Set $\tau_{\ell+1} = \tau_\ell$.
  end if
end for
for $t = t_e + 1$ **to** T **do**
  Set $y_t = y^*$.
  Set $p_t = P_\ell^*$.
end for

7.5 Numerical experiments

We conducted a number of numerical experiments in order to assess the performance of the policy.

7.5.1 Setup of experiments

We used the following parameter settings: $\mathcal{P} = [0.0, 50.0]$, $\mathcal{Y} = [0.0, 200.0]$, $K = 8$, $v = 0.1$, discriminative prices $\mathcal{P}^D = \{5, 15, 12.5, 20, 10, 35\}$. The value of y_{start} is randomly chosen from the set $\{10, 20, 30\}$ and the value of p_{start} is randomly chosen from the set $\mathcal{P}^D$.

We used the following demand functions for the hypothesis set Λ:

$$\exp(7.0 - 0.2p) + \epsilon \tag{7.15}$$
$$\exp(6.0 - 0.2p) + \epsilon \tag{7.16}$$
$$\exp(6.0 - 0.15p) + \epsilon \tag{7.17}$$
$$\exp(5.0 - 0.1p) + \epsilon \tag{7.18}$$
$$\exp(4.0 - 0.2p) + \epsilon \tag{7.19}$$
$$\exp(5.5 - 0.2p) + \epsilon \tag{7.20}$$
$$\exp(5.5 - 0.15p) + \epsilon \tag{7.21}$$
$$\exp(4.5 - 0.2p) + \epsilon \tag{7.22}$$

In our experiments the true demand model is given by (7.17). This choice for the hypothesis set Λ models a scenario where there are multiple mean demand functions that are similar but distinct, which in turn makes it harder to the policy to identify the correct demand model.

In the experiments the errors are taken from an uniform distribution with $\epsilon \sim \mathcal{U}(-5, 5)$. We let $b \in \{4, 6\}$, $h \in \{2, 4\}$, $T \in \{250, 500, 750, 1000, 5000, 10000\}$ and $m \in \{3, 4\}$. These choices for the parameter values of b and h model a scenario where lost sales are more costly to the seller than holding costs. Furthermore, in our experiments test two rules in order set the length of the learning and earning phases: the first rule uses $C_T = \lceil 10 \cdot \log T \rceil$ and the second rule uses $C_T = \lceil \sqrt{T} \rceil$.

In order to interpret the results, we report the estimated scaled per-period regret which is given by:

$$\hat{\mathcal{R}}^S(\pi, T) = \frac{|\hat{Q}(p^*, y^*) - \frac{1}{T}\sum_{t=1}^{T} \hat{R}_t(p_t, y_t)|}{\hat{Q}(p^*, y^*)} \cdot 100. \tag{7.23}$$

Here $\hat{Q}(p^*, y^*)$ is an estimate of $Q(p^*, y^*)$ based on simulation and $\hat{R}_t(p_t, y_t)$ is the observed revenue in period t.

$\hat{\mathcal{R}}^S(\pi, T)$ is an estimate of the scaled per-period regret $\mathcal{R}^{\mathcal{S}}(\pi, T)$ which is given by:

$$\mathcal{R}^{\mathcal{S}}(\pi, T) = \frac{Q(p^*, y^*) - \frac{1}{T}\mathbb{E}\left\{\sum_{t=1}^{T} R_t(p_t, y_t)\right\}}{Q(p^*, y^*)} \cdot 100 \tag{7.24}$$

In the remainder the estimated scaled per-period regret will simply referred to as the scaled per-period regret. For a particular choice of parameter settings, we average the scaled per-period regret over 250 simulations. The results are displayed in Table 7.1 and 7.2.

7.5.2 Results

The main insights from the experiments are as follows. The scaled per-period regret is decreasing with respect to the sales horizon which indicates that the per-period regret is decreasing with respect to the sales horizon. The scaled per-period regret tends to be larger for smaller sales horizons, and this makes sense, since the policy has less sales periods to determine the true underlying demand model. For smaller sales horizons the scaled per-period regret is relatively high, but as the sales horizon increases, it decreases relatively quickly for most of the parameter values considered. This indicates that, as the sales horizon increases, the policy is better able to make decisions that are close to optimal.

The magnitude of the scaled per-period regret depends on rule that is used to determine C_T. The results indicate that the rule $C_T = \lceil\sqrt{T}\rceil$ performs much better than $C_T = \lceil 10 \cdot \log T\rceil$. The differences in performance between the two rules is especially noticeable for smaller sales horizons. For smaller sales horizons the scaled per-period regret with $C_T = \lceil 10 \cdot \log T\rceil$ can be as high as 15% - 18% and this is about twice as high compared to $C_T = \lceil\sqrt{T}\rceil$. For larger horizons, the performance of the two rules are very similar.

The rate at which the scaled per-period regret decreases with the horizon appears to be related to (the difficulty of) the problem instance. In particular, when $h = b = 4$ the scaled regret decreases at a slower rate compared to the case when $h = 2$ and $b = 4$. One possible explanation is that, when $h = 2$ and $b = 4$, the policy recognizes that lost sales are more costly and it more obvious for the policy that it needs to learn the right values for the inventory decisions quickly. However, when $h = b = 4$, their is less incentive for the policy to learn the right values for inventory decisions quickly.

Overall, the results suggest that the growth rate of regret is sub-linear in the sales horizon T. The results are promising and indicate that the policy is able to learn the true demand model and set the right values for the price and inventory.

Table 7.1: Performance of heuristic policy with $C_T = \lceil 10 \cdot \log T \rceil$.

h	b	m	T	mean	std	h	b	m	T	mean	std
2	4	3	250	15.87	6.38	4	4	3	250	17.47	7.76
2	4	3	500	10.25	3.68	4	4	3	500	11.71	4.48
2	4	3	750	7.73	2.63	4	4	3	750	9.11	3.23
2	4	3	1000	5.79	2.08	4	4	3	1000	7.02	7.02
2	4	3	5000	0.98	0.60	4	4	3	5000	1.81	0.80
2	4	3	10000	0.26	0.26	4	4	3	10000	0.90	0.44
2	4	4	250	15.24	6.18	4	4	4	250	16.95	7.52
2	4	4	500	9.86	3.54	4	4	4	500	11.35	4.33
2	4	4	750	7.47	2.52	4	4	4	750	8.83	3.11
2	4	4	1000	5.60	2.00	4	4	4	1000	6.78	2.47
2	4	4	5000	0.94	0.60	4	4	4	5000	1.76	0.77
2	4	4	10000	0.25	0.25	4	4	4	10000	0.87	0.42
2	6	3	250	16.05	6.57	4	6	3	250	18.33	7.85
2	6	3	500	10.18	3.73	4	6	3	500	12.08	4.47
2	6	3	750	7.59	2.66	4	6	3	750	9.27	3.25
2	6	3	1000	5.65	2.09	4	6	3	1000	7.08	2.58
2	6	3	5000	0.75	0.62	4	6	3	5000	1.64	0.81
2	6	3	10000	0.34	0.10	4	6	3	10000	0.68	0.45
2	6	4	250	16.34	6.20	4	6	4	250	18.14	7.36
2	6	4	500	10.34	3.53	4	6	4	500	11.93	4.23
2	6	4	750	7.70	2.54	4	6	4	750	9.16	3.03
2	6	4	1000	5.70	2.01	4	6	4	1000	6.97	2.41
2	6	4	5000	0.76	0.59	4	6	4	5000	1.56	0.73
2	6	4	10000	0.32	0.09	4	6	4	10000	0.64	0.40

Table 7.2: Performance of heuristic policy with $C_T = \lceil \sqrt{T} \rceil$.

h	b	m	T	mean	std	h	b	m	T	mean	std
2	4	3	250	7.95	3.43	4	4	3	250	8.89	3.24
2	4	3	500	4.86	1.70	4	4	3	500	5.89	1.86
2	4	3	750	3.48	1.20	4	4	3	750	4.32	1.32
2	4	3	1000	2.74	1.06	4	4	3	1000	3.61	1.22
2	4	3	5000	0.69	0.48	4	4	3	5000	1.49	0.66
2	4	3	10000	0.29	0.32	4	4	3	10000	0.98	0.48
2	4	4	250	7.89	3.38	4	4	4	250	9.18	3.44
2	4	4	500	4.82	1.71	4	4	4	500	5.95	1.86
2	4	4	750	3.45	1.21	4	4	4	750	4.36	1.32
2	4	4	1000	2.69	1.04	4	4	4	1000	3.64	1.24
2	4	4	5000	0.66	0.47	4	4	4	5000	1.43	0.62
2	4	4	10000	0.27	0.31	4	4	4	10000	0.96	0.45
2	6	3	250	8.15	4.23	4	6	3	250	8.87	3.72
2	6	3	500	4.76	2.01	4	6	3	500	5.70	1.99
2	6	3	750	3.32	1.38	4	6	3	750	4.21	1.43
2	6	3	1000	2.55	1.17	4	6	3	1000	3.45	1.26
2	6	3	5000	0.48	0.46	4	6	3	5000	1.29	0.65
2	6	3	10000	0.35	0.14	4	6	3	10000	0.76	0.49
2	6	4	250	8.36	4.36	4	6	4	250	9.58	4.57
2	6	4	500	4.93	2.04	4	6	4	500	5.97	2.22
2	6	4	750	3.43	1.38	4	6	4	750	4.34	1.50
2	6	4	1000	2.66	1.20	4	6	4	1000	3.57	1.40
2	6	4	5000	0.49	0.45	4	6	4	5000	1.24	0.60
2	6	4	10000	0.33	0.13	4	6	4	10000	0.72	0.43

7.6 Conclusion

In this Chapter we studied a dynamic pricing problem with limited price changes and censored demand. In contrast with previous work, we did not assume that we can observe true demand and lost sales. Furthermore, in our setting, the lost sales is part of the objective function that the seller aims to optimize. We proposed a heuristic pricing policy that adjust prices and inventory levels for this problem. Using numerical experiments we tested the performance of our policy. Experimental results are promising and suggest that the growth rate of regret is sub-linear with respect to the sales horizon.

Future work could be directed towards deriving analytical results related to the dynamic pricing problem studied in this Chapter. In particular, it would be interesting to study upper and lower bounds on the growth rate of regret and how these bounds depend on the number of price changes and the sales horizon.

Chapter 8

Summary and conclusions

In this Chapter we provide an overview of the main contributions of this thesis. Next, we discuss some limitations of the approaches used and we indicate some directions for future research. Finally, we put the contributions of this thesis in a broader context and relate them to other topics and research areas that relate to revenue management in online markets.

8.1 General overview

In this dissertation we have focused on revenue management in online markets. The revenue management problems that are studied apply to decision making problems that arise in online markets such as online advertisement markets and in e-commerce settings such as online retail markets.

The common theme across the Chapters in this thesis is that the decisions in the revenue management problems are made under uncertainty. The revenue management problems that are considered in this thesis can be broadly organized in three categories: allocation decisions, pricing decisions, and buying decisions. Chapters 2, 3, 4, 5, 7 focus on the perspective of a seller and tackles allocation decisions and pricing decisions. In Chapter 6, the emphasis is on the buyer perspective and studies buying decisions.

In Chapters 2, 3, 4, 5, 6 the focus is on decision making in the context of online advertising. The revenue management problems studied in these chapters are related to the decisions of buyers and sellers that participate in online auctions for advertisements and to the interaction of various selling mechanisms that are used in order to sell online advertisements. Chapter 2 starts with an allocation problem – the display-ad allocation problem – that publishers face in online advertising. We consider a publisher that sells impressions via guaranteed contracts and via the RTB market by using a waterfall mechanism. In this problem, the publisher needs to take the uncertainty of the RTB market into account and decide which impressions to

allocate to guaranteed contracts and which impressions to allocate to the RTB market. In Chapter 3, we shift our attention to pricing decisions that publishers need to make. Publishers typically specify a minimum price – called the *floor price* – for their impressions when selling on the RTB market via an ad exchange. Chapter 3 considers a situation where impressions are sold via second-price auctions (one of the dominant auction formats used in online advertising) and where the publisher needs to learn the best floor price over time. In second-price auctions, the floor price is often referred to as the *reserve price* and so we refer to this problem as the reserve price optimization problem. Chapter 4 builds on Chapter 3 and studies reserve price optimization in the setting of Header Bidding. Note that, in the reserve price optimization problem of Chapter 3, the publisher received a single offer at a time for each impression. However, in Chapter 4, we consider a publisher that receives multiple offers for a single impression and where each offer is the result of a second-price auction with a reserve price. The goal of the publisher is to learn the best reserve price for each second-price auction (i.e., a vector of reserve prices) in order to maximize his expected revenue. Chapter 5 builds on the ideas and problem settings of Chapter 3 and 4 and studies how publishers should make decisions in order to maximize revenues when they have access to both header bidding and an ad exchange. Chapter 6 switches perspectives and focuses on buying decisions in online advertising auctions. More specifically, we consider buying decisions in repeated posted-price auction where a seller repeatedly interacts with a buyer for a number of time periods and where the buyer wants to maximize his expected utility over time. In this problem, the buyer needs to learn at what prices it is worthwhile to purchase an item when her valuation for the item is unknown. Chapter 7 moves away from decisions in auction settings and instead studies a more general dynamic pricing problem. More specifically, we consider a seller that has to adjust his selling prices and inventory levels over time in order to maximize his expected revenue. In addition, the seller faces a business constraint on the number of price changes allowed during the selling horizon and the seller only has a limited amount of inventories on hand in each selling period.

8.2 Detailed overview of main results

Below we present a detailed overview of the main results and conclusions for each Chapter in this dissertation.

In Chapter 2 we consider a display-ad allocation problem where an online publisher needs to decide which subset of impressions for advertisement slots should be used in order to fulfill guaranteed contracts and which subset should be sold (in a waterfall mechanism) via Supply Side Platforms (SSPs) in order to maximize the expected revenue. The way that information is revealed over time allows us to model the display-ad allocation problem as a two-stage stochastic program. Moreover, our modeling approach also takes the uncertainty associated with the sale of an impres-

sion by an SSP into account. The experiments indicate that by carefully modeling the sequential nature of decisions in the waterfall mechanism and incorporating this in a stochastic programming framework, it is possible to outperform greedy heuristics that ignore uncertainty and give priority to guaranteed contracts. Our results suggest that the benefit of using our proposed method is highest in periods where the website traffic is high compared with the targets for the guaranteed contracts.

In Chapter 3, we study reserve price optimization in second-price auctions in a setting where the publisher has limited information. In particular, we study a limited information setting where the probability distribution of the bids from advertisers is unknown and the values of the bids are not revealed to the publisher. Furthermore, we do not assume that the publisher has access to a historical data set with bids. The experiments in Chapter 3 show that by incorporating the rules of second-price auctions into a multi-armed bandit framework, one can often improve upon the performance that can be obtained by traditional bandit algorithms. More specifically, algorithms that exploit the rules of the second-price auction tend to learn better actions much quicker than state-of-the-art bandit algorithms that ignore these rules. In non-stationary environments, algorithms that exploit the rules of the second-price auction tend to adapt much quicker to new environments compared to bandit algorithms that ignore these rules.

In Chapter 4, we study the reserve price optimization problem in the context of Header Bidding. We show that the reserve price optimization problem can be modeled as slate bandit problem with a non-separable reward function (i.e., the optimal value of the function cannot be determined by learning the optimal action for each slot). We are mainly interested in cases where the number of slates is large relative to the time horizon, so that trying each slate as a separate arm in a traditional multi-armed bandit, would not be feasible. In Chapter 4, we first show that existing algorithms are not suitable for the slate bandit problem with non-separable reward functions. Next, we propose algorithms that have sub-linear regret with respect to the time horizon and that avoid trying each slate. In addition, we show that our algorithms can even be applied when the number of slates is larger than the horizon and that they will still have regret of the same order. This is in contrast with benchmark algorithms such as UCB1, which cannot be applied in that case. Our solution therefore provides a substantial improvement relative to what is possible based on the current state-of-the-art. Experiments using simulated data and real-world data illustrate the effectiveness of our algorithms. Our experiments show that ignoring non-separability can have a large effect on the regret.

In Chapter 5, we study how publishers should make decisions in order to maximize revenues when they have access to both header bidding and an ad exchange in order to sell impressions. More specifically, we consider a publisher that first observes an offer from header bidding and can accept or reject this offer. If the offer is rejected, they can try to sell the impression on an ad exchange. In this problem, the publisher needs to make two decisions: (i) when to accept the offer from header bidding and

when to use the ad exchange; and (ii) if the publisher uses the ad exchange, which floor price it should use on the ad exchange. In Chapter 5, we study how publishers should set their floor prices in order to maximize expected revenues when they have access to both selling mechanisms. We propose two algorithms for this problem based on techniques from the multi-armed bandit literature, and show that their regret – the performance loss compared to the optimal algorithm – is sub-linear in the time horizon. Experiments using simulated data and real-world data illustrate the effectiveness of our algorithms. Our experiments show that optimization of revenues that takes both selling mechanisms into account leads to a substantial improvement over the expected revenue that can be obtained by using just one selling mechanism.

In Chapter 6, we study buying decisions in repeated posted-price auction where a seller repeatedly interacts with a buyer for a number of rounds. We study a setting where the buyer does not know the distribution of his valuation and must learn this over time. We study two types of buyers: non-strategic buyers and strategic buyers. We first consider non-strategic buyers and derive algorithms with sub-linear regret bounds that hold irrespective of the observed prices offered by the seller. These algorithms are then adapted into algorithms with similar guarantees for non-strategic buyers. We provide a theoretical analysis of our proposed algorithms and support our findings with numerical experiments. Our experiments, confirm our theoretical findings and show that buyers can indeed learn to make optimal buying decisions that minimize their regret. Our experiments also show that, if the seller uses a particular class of low-regret learning algorithms for selecting the price, then strategic buyers can obtain much higher utilities compared to non-strategic buyers.

In Chapter 7, we study a dynamic pricing problem with demand censoring and limited price changes. More specifically, we consider a seller that faces demand uncertainty and has to adjust his selling price over the selling horizon in order to learn the optimal price and maximize his cumulative revenue over the selling horizon. The seller faces a business constraint on the number of price changes allowed during the selling horizon and the seller only has a limited (finite) amount of inventories on hand in each selling period. This is a challenging problem and it is not immediately clear whether the seller can learn the right actions over time. We propose a policy that adjust prices and inventory levels and study its performance using numerical experiments. The experiments show that, despite the difficulty of the problem, the seller can still learn to take good actions in a variety of problem instances. In particular, the experiments indicate that the proposed policy has sub-linear regret with respect to the sales horizon.

8.3 Limitations and future research

The models and algorithms developed in this thesis can be extended in a number of ways. Here we give an overview of some interesting directions for future research.

In Chapter 2, we study a revenue management problem where publishers needs to balance between allocating impressions to guaranteed contracts and allocating impressions in order to sell them on the RTB market via Supply Side Platforms (SSPs) in a waterfall mechanism. The model presented in Chapter 2 assumes that SSPs can be ordered according to expected revenue and the main source of uncertainty is associated with the probability of a sale by an SSP. However, in practice, publishers can also specify a reserve price when they sell impressions on the RTB market. Future work can be directed towards incorporating reserve prices in the allocation decisions between guaranteed contracts and sales on the RTB market using a waterfall mechanism.

In Chapter 3, we study reserve price optimization in second-price auctions in a setting where the publisher has limited information. The experiments in Chapter 3 show that by incorporating the rules of second-price auctions, one can often improve upon the performance that can be obtained by traditional bandit algorithms. However, the algorithms presented in Chapter 3 have not been analyzed theoretically. Future work can be directed towards deriving theoretical guarantees for algorithms that incorporate the rules of second-price auctions with multi-armed bandit algorithms. Such an analysis would provide extra insight about the reserve price problem.

In Chapter 4, we study reserve price optimization in the context of header bidding. The optimization problem is formulated as a slate bandit problem and a theoretical analysis of the proposed algorithms is provided. Future work can be directed towards deriving similar or improved theoretical guarantees for algorithms that make less restrictive assumptions. For example, it would be interesting to derive efficient algorithms that do not require the independence assumption between the slots. Our algorithms have an explore-then-commit type of structure. Another interesting question is whether techniques such as Thompson Sampling can be used to guide the exploration-exploitation trade-off. Moreover, the analysis in Chapter 4, does not take any additional features related to online advertisement auctions (such as characteristics of the ad slot and users) into account. Future research can be directed towards extending the algorithms in Chapter 4 in order that such contextual information can also be taken into account.

In Chapter 5, we study how publishers should make decisions in order to maximize revenues when they have access to both header bidding and an ad exchange in order to sell impressions. Two algorithms are proposed and analyzed both theoretically and using numerical experiments. Also, a lower bound on the regret is provided. Our analysis showed that our algorithms near-optimal since the upper bound on the regret matches the lower bound up to logarithmic factors. The analysis in Chapter 5, does not take any additional features related to online advertisement auctions (such as characteristics of the ad slot and users) into account. Future work can be directed towards algorithms that take additional information into account in the form of features, similar to for example contextual bandits. It would be interesting to study whether optimal algorithms exist and which assumptions are needed in order to derive such performance guarantees. Another direction would be to investigate whether it is

possible to derive performance guarantees when the rewards on the ad exchange are adversarial instead of stochastic.

In Chapter 6, we study repeated posted-price auctions between a single seller and a single buyer. Several algorithms are proposed and analyzed both theoretically and using numerical experiments. In practice, buyers also face budget constraints when making their purchasing decisions. However, the models and algorithms presented in Chapter 6 do not take budget constraints into account. Future work can be directed towards deriving algorithms that can take both strategic behaviour and budget constraints into account.

In Chapter 7, we study a dynamic pricing problem with limited price changes and censored demand. We proposed a heuristic pricing policy that adjust prices and inventory levels for this problem. Using numerical experiments we tested the performance of our policy. Experimental results suggest that the regret is sub-linear with respect to the sales horizon. Future work could be directed towards deriving analytical results related to this pricing problem. In particular, it would be interesting to study upper and lower bounds on the regret and how these bounds depend on the number of price changes and the sales horizon.

8.4 General discussion

In this dissertation we have focused on revenue management in online markets. The revenue management problems that we study are related to situations that arise in online markets such as online advertisement markets and online retail markets. In this section we put the contributions of this thesis in a broader context and relate them to other topics and research areas that relate to revenue management in online markets.

The research in this thesis focuses on online display advertising, but there is another area in online advertising called sponsored search advertising. Display advertising refers to advertisements that are displayed when users visit websites in browsers and are typically displayed in banners located in specific ad slots on a website. Sponsored search advertising refers to advertisements that are displayed when users search for keywords on search engines. In sponsored search advertising, the advertisers bid for their positions in the ranking of displayed search results, and the slots are also sold in real-time via auctions (similar to display advertising). However, there are also some fundamental differences between the two types of advertising. For example, in sponsored search advertising the auction format (generalized second-price auction) and the payment model (the seller only gets revenue if the user clicks on the advertisement) are different. As a consequence, sponsored search advertising requires different models and algorithms than those presented in this thesis.

In Chapter 7 we considered a dynamic pricing problem which is applicable to the setting of online retail markets. While pricing is an important aspect of revenue

management for online retail markets, another important aspect is assortment optimization [5, 131]. Assortment optimization is a critical decision that is regularly made by retailers. The decision involves a trade-off between offering a larger assortment of products but smaller inventories of each product and offering a smaller number of varieties with more inventory of each product. Assortment optimization decisions are important, because sub-optimal assortment decisions lead to missed sales or increased inventory costs and these have a negative impact on revenues.

Looking beyond online advertising and online retail markets, we note that there are other online markets where revenue management is a challenging task. One example is related to the sharing-economy and is given by companies that provide ride-sharing services. Ride-sharing companies such as Uber and Lyft in the USA and Didi Chuxing in China face a number of challenges with respect to revenue management. Two key aspects that determine the revenue for ride-sharing companies are pricing and matching [128]. Pricing determines both customer demand and driver supply because lower prices attract customers and higher prices attract drivers. Meanwhile, matching connects a customer requesting a ride with a driver, and that determines how long the customer must wait for driver pick-up. Together, the pricing and matching decisions affect the geospatial distribution of the drivers at any given point in time. In turn, that geospatial distribution determines the set of drivers who are available to be matched with an arriving customer. However, a customer offered a far-away driver may not accept the ride due to the long pick-up time. Revenue management in this context is a challenging problem, since firms need to decide on the right pricing mechanism and contracts to offer the suppliers [38, 153], and also take other considerations into account such as flexibility [150] and participation behaviour of passengers [74].

The examples mentioned above show that there are many more revenue management problems other than those discussed in this thesis. However, a common property of the revenue management problems outlined above and those studied in this thesis, is that they involve decision making under uncertainty. In particular, some of the problems mentioned above involve decision making with partial feedback. We believe that the techniques and results developed in this thesis can therefore be of interest when solving aforementioned revenue management problems.

8.5 Final remarks

In this thesis we have focused on revenue management in online markets. The revenue management problems that are studied apply to decision making problems that arise in online markets such as online advertisement markets and in e-commerce settings such as online retail markets. The revenue management problems that are considered in this thesis can be broadly organized in three categories: allocation decisions, pricing decisions, and buying decisions. This thesis proposes models and algorithms that can be used for each of these decisions. Our techniques and results

are of interest to anyone that wants to make decisions that involve uncertainty in revenue management problems.

You have just finished reading this dissertation. We hope it was an enjoyable experience.

References

[1] S. Adikari and K. Dutta. 'A New Approach to Real-Time Bidding in Online Advertisements: Auto Pricing Strategy'. *INFORMS Journal on Computing* 31.1 (2019), pp. 66–82.

[2] M. Adler, P.B. Gibbons and Y. Matias. 'Scheduling space-sharing for internet advertising'. *Journal of Scheduling* 5.2 (2002), pp. 103–119.

[3] R.R. Afshar, Y. Zhang, M. Firat and U. Kaymak. 'A Decision Support Method to Increase the Revenue of Ad Publishers in Waterfall Strategy'. *2019 IEEE Conference on Computational Intelligence for Financial Engineering Economics (CIFEr)*. May 2019, pp. 1–8.

[4] R.R. Afshar., Y. Zhang., M. Firat. and U. Kaymak. 'A Reinforcement Learning Method to Select Ad Networks in Waterfall Strategy'. *Proceedings of the 11th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART,* INSTICC. SciTePress, 2019, pp. 256–265.

[5] S. Agrawal, V. Avadhanula, V. Goyal and A. Zeevi. 'MNL-Bandit: A Dynamic Learning Approach to Assortment Selection'. *Operations Research* 67.5 (2019), pp. 1453–1485.

[6] S. Agrawal and N. Goyal. 'Analysis of Thompson Sampling for the Multi-armed Bandit Problem'. *Proceedings of the 25th Annual Conference on Learning Theory*. Ed. by S. Mannor, N. Srebro and R.C. Williamson. Vol. 23. Proceedings of Machine Learning Research. Edinburgh, Scotland: PMLR, 25–27 Jun 2012, pp. 39.1–39.26.

[7] S. Agrawal and N. Goyal. 'Analysis of Thompson Sampling for the Multi-armed Bandit Problem'. *Proceedings of the 25th Annual Conference on Learning Theory*. Vol. 23. Proceedings of Machine Learning Research. Edinburgh, Scotland: PMLR, 25–27 Jun 2012, pp. 39.1–39.26.

[8] S. Agrawal and N. Goyal. 'Further Optimal Regret Bounds for Thompson Sampling'. *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*. Vol. 31. 2013, pp. 99–107.

[9] S. Agrawal and N. Goyal. 'Further Optimal Regret Bounds for Thompson Sampling'. *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*. Vol. 31. PMLR, 2013, pp. 99–107.

[10] K. Amin, A. Rostamizadeh and U. Syed. 'Learning Prices for Repeated Auctions with Strategic Buyers'. *Proceedings of the 26th International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2013, pp. 1169–1177.

[11] K. Amin, A. Rostamizadeh and U. Syed. 'Repeated Contextual Auctions with Strategic Buyers'. *Advances in Neural Information Processing Systems 27*. 2014, pp. 622–630.

[12] W. Aqeel, D. Bhattacherjee, B. Chandrasekaran, P.B. Godfrey, G. Laughlin, B. Maggs and A. Singla. 'Untangling Header Bidding Lore'. *Passive and Active Measurement*. Ed. by A. Sperotto, A. Dainotti and B. Stiller. Cham: Springer International Publishing, 2020, pp. 280–297.

[13] J.-Y. Audibert and S. Bubeck. 'Regret Bounds and Minimax Policies Under Partial Monitoring'. *J. Mach. Learn. Res.* 11 (Dec. 2010), pp. 2785–2836.

[14] J.-Y. Audibert, R. Munos and C. Szepesvári. 'Exploration–exploitation tradeoff using variance estimates in multi-armed bandits'. *Theoretical Computer Science* 410.19 (2009). Algorithmic Learning Theory, pp. 1876–1902.

[15] P. Auer, N. Cesa-Bianchi, Y. Freund and R. Schapire. 'The Nonstochastic Multiarmed Bandit Problem'. *SIAM Journal on Computing* 32.1 (2002), pp. 48–77.

[16] P. Auer. 'Using Confidence Bounds for Exploitation-exploration Trade-offs'. *J. Mach. Learn. Res.* 3 (Mar. 2003), pp. 397–422.

[17] P. Auer, N. Cesa-Bianchi and P. Fischer. 'Finite-time Analysis of the Multiarmed Bandit Problem'. *Machine Learning* 47.2 (May 2002), pp. 235–256.

[18] D. Austin, S. Seljan, J. Monello and S. Tzeng. 'Reserve Price Optimization at Scale'. *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. Oct. 2016, pp. 528–536.

[19] A. Ayanso and A. Karimi. 'The moderating effects of keyword competition on the determinants of ad position in sponsored search advertising'. *Decision Support Systems* 70 (2015), pp. 42–59.

[20] M. Babaioff, S. Dughmi, R. Kleinberg and A. Slivkins. 'Dynamic Pricing with Limited Supply'. *ACM Trans. Econ. Comput.* 3.1 (Mar. 2015).

[21] M. Babaioff, R.D. Kleinberg and A. Slivkins. 'Truthful Mechanisms with Implicit Payment Computation'. *Proceedings of the 11th ACM Conference on Electronic Commerce*. Association for Computing Machinery, 2010, pp. 43–52.

[22] M. Babaioff, Y. Sharma and A. Slivkins. 'Characterizing truthful multi-armed bandit mechanisms'. *SIAM Journal on Computing* 43.1 (2014), pp. 194–230.

[23] S.R. Balseiro, O. Besbes and G.Y. Weintraub. 'Repeated Auctions with Budgets in Ad Exchanges: Approximations and Design'. *Management Science* 61.4 (2015), pp. 864–884.

[24] S.R. Balseiro, J. Feldman, V. Mirrokni and S. Muthukrishnan. 'Yield Optimization of Display Advertising with Ad Exchange'. *Management Science* 60.12 (2014), pp. 2886–2907.

[25] G. Beliakov, A. Pradera and T. Calvo. *Aggregation Functions: A Guide for Practitioners*. Springer Publishing Company, Incorporated, 2008.

[26] O. Besbes, Y. Gur and A. Zeevi. 'Optimal Exploration–Exploitation in a Multi-armed Bandit Problem with Non-stationary Rewards'. *Stochastic Systems* 9.4 (2019), pp. 319–337.

[27] O. Besbes and A. Muharremoglu. 'On Implications of Demand Censoring in the Newsvendor Problem'. *Management Science* 59.6 (2013), pp. 1407–1424.

[28] O. Besbes and A. Zeevi. 'On the (Surprising) Sufficiency of Linear Models for Dynamic Pricing with Demand Learning'. *Management Science* 61.4 (2015), pp. 723–739.

[29] V. Bharadwaj, P. Chen, W. Ma, C. Nagarajan, J. Tomlin, S. Vassilvitskii, E. Vee and J. Yang. 'SHALE: An Efficient Algorithm for Allocation of Guaranteed Display Advertising'. *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '12. Beijing, China: ACM, 2012, pp. 1195–1203.

[30] J.R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. 2nd. Springer Publishing Company, Incorporated, 2011.

[31] C.M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[32] A. Blum, V. Kumar, A. Rudra and F. Wu. 'Online Learning in Online Auctions'. *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2003, pp. 202–204.

[33] V. Boskamp, A. Knoops, F. Frasincar and A. Gabor. 'Maximizing Revenue with Allocation of Multiple Advertisements on a Web Banner'. *Comput. Oper. Res.* 38.10 (Oct. 2011), pp. 1412–1424.

[34] D. Bouneffouf and R. Féraud. 'Multi-armed bandit problem with known trend'. *Neurocomputing* 205 (2016), pp. 16–21.

[35] M. Braverman, J. Mao, J. Schneider and M. Weinberg. 'Selling to a No-Regret Buyer'. *Proceedings of the 2018 ACM Conference on Economics and Computation*. ACM, 2018, pp. 523–538.

[36] J. Broder and P. Rusmevichientong. 'Dynamic Pricing Under a General Parametric Choice Model'. *Operations Research* 60.4 (2012), pp. 965–980.

[37] S. Bubeck and N. Cesa-Bianchi. 'Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems'. *Foundations and Trends® in Machine Learning* 5.1 (2012), pp. 1–122.

[38] G.P. Cachon, K.M. Daniels and R. Lobel. 'The Role of Surge Pricing on a Service Platform with Self-Scheduling Capacity'. *Manufacturing & Service Operations Management* 19.3 (2017), pp. 368–384.

[39] Y. Cao, Z. Wen, B. Kveton and Y. Xie. 'Nearly Optimal Adaptive Procedure with Change Detection for Piecewise-Stationary Bandit'. *Proceedings of Machine Learning Research*. Ed. by K. Chaudhuri and M. Sugiyama. Vol. 89. Proceedings of Machine Learning Research. PMLR, 16–18 Apr 2019, pp. 418–427.

[40] S. Caron, B. Kveton, M. Lelarge and S. Bhagat. 'Leveraging Side Observations in Stochastic Bandits'. *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*. UAI'12. Catalina Island, CA: AUAI Press, 2012, pp. 142–151.

[41] N. Cesa-Bianchi, C. Gentile and Y. Mansour. 'Regret Minimization for Reserve Prices in Second-Price Auctions'. *IEEE Transactions on Information Theory* 61.1 (Jan. 2015), pp. 549–564.

[42] N. Cesa-Bianchi, C. Gentile and Y. Mansour. 'Regret Minimization for Reserve Prices in Second-Price Auctions'. *IEEE Transactions on Information Theory* 61.1 (Jan. 2015), pp. 549–564.

[43] N. Cesa-Bianchi and G. Lugosi. 'Combinatorial bandits'. *Journal of Computer and System Sciences* 78.5 (2012). JCSS Special Issue: Cloud Computing 2011, pp. 1404–1422.

[44] P. Chahuara, N. Grislain, G. Jauvion and J.-M. Renders. 'Real-Time Optimization of Web Publisher RTB Revenues'. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '17. Halifax, NS, Canada: ACM, 2017, pp. 1743–1751.

[45] B. Chen, X. Chao and H. Ahn. 'Dynamic pricing and inventory control with nonparametric demand learning'. *Working paper* (2018).

[46] B. Chen, J. Huang, Y. Huang, S. Kollias and S. Yue. 'Combining guaranteed and spot markets in display advertising: Selling guaranteed page views with stochastic demand'. *European Journal of Operational Research* 280.3 (2020), pp. 1144–1159.

[47] B. Chen, S. Yuan and J. Wang. 'A Dynamic Pricing Model for Unifying Programmatic Guarantee and Real-Time Bidding in Display Advertising'. *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*. ADKDD'14. New York, NY, USA: ACM, 2014, 1:1–1:9.

[48] B. Chen and X. Chao. 'Parametric demand learning with limited price explorations in a backlog stochastic inventory system'. *IISE Transactions* 0.0 (2019), pp. 1–9.

[49] B. Chen, X. Chao and H.-S. Ahn. 'Coordinating Pricing and Inventory Replenishment with Nonparametric Demand Learning'. *Operations Research* 0.0 (0), null.

[50] P. Chen, W. Ma, S. Mandalapu, C. Nagarjan, J. Shanmugasundaram, S. Vassilvitskii, E. Vee, M. Yu and J. Zien. 'Ad Serving Using a Compact Allocation Plan'. *Proceedings of the 13th ACM Conference on Electronic Commerce*. EC '12. Valencia, Spain: ACM, 2012, pp. 319–336.

[51] W. Chen, W. Hu, F. Li, J. Li, Y. Liu and P. Lu. 'Combinatorial Multi-Armed Bandit with General Reward Functions'. *Advances in Neural Information Processing Systems 29*. 2016, pp. 1659–1667.

[52] W. Chen, Y. Wang and Y. Yuan. 'Combinatorial Multi-Armed Bandit: General Framework and Applications'. *Proceedings of the 30th International Conference on Machine Learning*. Vol. 28. Proceedings of Machine Learning Research 1. PMLR, 17–19 Jun 2013, pp. 151–159.

[53] W. Chen, Y. Wang, Y. Yuan and Q. Wang. 'Combinatorial Multi-Armed Bandit and Its Extension to Probabilistically Triggered Arms'. *J. Mach. Learn. Res.* 17.1 (Jan. 2016), pp. 1746–1778.

[54] Y.-J. Chen. 'Optimal Dynamic Auctions for Display Advertising'. *Operations Research* 65.4 (2017), pp. 897–913.

[55] W.C. Cheung, D. Simchi-Levi and H. Wang. 'Technical Note—Dynamic Pricing and Demand Learning with Limited Price Experimentation'. *Operations Research* 65.6 (2017), pp. 1722–1731.

[56] W.C. Cheung, D. Simchi-Levi and H. Wang. 'Technical Note—Dynamic Pricing and Demand Learning with Limited Price Experimentation'. *Operations Research* 65.6 (2017), pp. 1722–1731.

[57] H. Choi, C.F. Mela, S. Balseiro and A. Leary. 'Online display advertising markets: A literature review and future directions'. *Columbia Business School Research Paper* 18-1 (2019).

[58] R. Combes and A. Proutiere. 'Unimodal Bandits: Regret Lower Bounds and Optimal Algorithms'. *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*. ICML'14. Beijing, China: JMLR.org, 2014, pp. I-521–I-529.

[59] R. Combes, M.S. Talebi Mazraeh Shahi, A. Proutiere and m. lelarge marc. 'Combinatorial Bandits Revisited'. *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 2015, pp. 2116–2124.

[60] J. Deane and A. Agarwal. 'Scheduling online advertisements to maximize revenue under variable display frequency'. *Omega* 40.5 (2012), pp. 562–570.

[61] R. Degenne, E. Garcelon and V. Perchet. 'Bandits with Side Observations: Bounded vs. Logarithmic Regret'. *CoRR* abs/1807.03558 (2018).

[62] R. Degenne and V. Perchet. 'Anytime optimal algorithms in stochastic multi-armed bandits'. *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by M.F. Balcan and K.Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1587–1595.

[63] A.V. den Boer. 'Dynamic pricing and learning: Historical origins, current research, and new directions'. *Surveys in Operations Research and Management Science* 20.1 (2015), pp. 1–18.

[64] A.V. den Boer. 'Tracking the market: Dynamic pricing and learning in a changing environment'. *European Journal of Operational Research* 247.3 (2015), pp. 914–927.

[65] A.V. den Boer and B. Zwart. 'Simultaneously Learning and Optimizing Using Controlled Variance Pricing'. *Management Science* 60.3 (2014), pp. 770–783.

[66] A.V. den Boer and B. Zwart. 'Dynamic Pricing and Learning with Finite Inventories'. *Operations Research* 63.4 (2015), pp. 965–978.

[67] Y. Deng, J. Schneider and B. Sivan. 'Prior-Free Dynamic Auctions with Low Regret Buyers'. *Advances in Neural Information Processing Systems 32*. 2019, pp. 4803–4813.

[68] N.R. Devanur and S.M. Kakade. 'The Price of Truthfulness for Pay-per-Click Auctions'. *Proceedings of the 10th ACM Conference on Electronic Commerce*. 2009, pp. 99–106.

[69] A. Deza, K. Huang and M.R. Metel. 'Chance constrained optimization for targeted Internet advertising'. *Omega* 53 (2015), pp. 90–96.

[70] M. Dimakopoulou, N. Vlassis and T. Jebara. 'Marginal Posterior Sampling for Slate Bandits'. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. July 2019, pp. 2223–2229.

[71] L. Ding, M. Liu, W. Kang and X. Zhao. 'Prior-free auction mechanism for online supplier with risk taking'. *Computers & Industrial Engineering* 133 (2019), pp. 1–8.

[72] W. Ding, T. Qiny, X.-D. Zhang and T.-Y. Liu. 'Multi-Armed Bandit with Budget Constraint and Variable Costs'. *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*. AAAI Press, 2013, pp. 232–238.

[73] P. Dinodia. *Header bidding vs waterfall: how the two revenue optimisation hacks differ*. Feb. 2017.

[74] Z. Dong and M. Leng. 'Managing on-demand ridesharing operations: Optimal pricing decisions for a ridesharing platform'. *International Journal of Production Economics* (2020), p. 107958.

[75] A. Drutsa. 'Horizon-Independent Optimal Pricing in Repeated Auctions with Truthful and Strategic Buyers'. *Proceedings of the 26th International Conference on World Wide Web*. 2017, pp. 33–42.

[76] A. Drutsa. 'Weakly Consistent Optimal Pricing Algorithms in Repeated Posted-Price Auctions with Strategic Buyer'. *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. PMLR, Oct. 2018, pp. 1319–1328.

[77] eBay. *How reserve prices work*. 2020.

[78] B. Edelman and M. Ostrovsky. 'Strategic bidder behavior in sponsored search auctions'. *Decision support systems* 43.1 (2007), pp. 192–198.

[79] J. Feldman, M. Henzinger, N. Korula, V.S. Mirrokni and C. Stein. 'Online Stochastic Packing Applied to Display Ad Allocation'. *Proceedings of the 18th Annual European Conference on Algorithms: Part I*. ESA'10. Liverpool, UK: Springer-Verlag, 2010, pp. 182–194.

[80] J. Feldman, A. Mehta, V. Mirrokni and S. Muthukrishnan. 'Online Stochastic Matching: Beating 1-1/e'. *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science*. FOCS '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 117–126.

[81] Z. Feng, C. Podimata and V. Syrgkanis. 'Learning to Bid Without Knowing Your Value'. *Proceedings of the 2018 ACM Conference on Economics and Computation*. 2018, pp. 505–522.

[82] A. Freund and J. Naor. 'Approximating the Advertisement Placement Problem'. *Journal of Scheduling* 7.5 (Sept. 2004), pp. 365–374.

[83] A. Garivier and O. Cappé. 'The KL-UCB Algorithm for Bounded Stochastic Bandits and Beyond'. *Proceedings of the 24th Annual Conference on Learning Theory*. Ed. by S.M. Kakade and U. von Luxburg. Vol. 19. Proceedings of Machine Learning Research. Budapest, Hungary: PMLR, Sept. 2011, pp. 359–376.

[84] N. Gatti, A. Lazaric and F. Trovò. 'A Truthful Learning Mechanism for Contextual Multi-Slot Sponsored Search Auctions with Externalities'. *Proceedings of the 13th ACM Conference on Electronic Commerce*. ACM, 2012, pp. 605–622.

[85] R. Gopal, X. Li and R. Sankaranarayanan. 'Online keyword based advertising: Impact of ad impressions on own-channel and cross-channel click-through rates'. *Decision Support Systems* 52.1 (2011), pp. 1–8.

[86] D. He, W. Chen, L. Wang and T.-Y. Liu. 'Online learning for auction mechanism in bandit setting'. *Decision Support Systems* 56 (2013), pp. 379–386.

[87] B. Heymann. 'How to bid in unified second-price auctions when requests are duplicated'. *Operations Research Letters* 48.4 (2020), pp. 446–451.

[88] W. Hoeffding. 'Probability Inequalities for Sums of Bounded Random Variables'. *Journal of the American Statistical Association* 58.301 (1963), pp. 13–30.

[89] W. Hoeffding. 'Probability Inequalities for Sums of Bounded Random Variables'. *Journal of the American Statistical Association* 58.301 (1963), pp. 13–30.

[90] A. Hojjat, J. Turner, S. Cetintas and J. Yang. 'Delivering Guaranteed Display Ads Under Reach and Frequency Requirements'. *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI'14. Québec City, Québec, Canada: AAAI Press, 2014, pp. 2278–2284.

[91] Z. Huang, J. Liu and X. Wang. 'Learning Optimal Reserve Price Against Non-myopic Bidders'. *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*. NIPS'18. Montreal, Canada: Curran Associates Inc., 2018, pp. 2042–2052.

[92] W.T. Huh, R. Levi, P. Rusmevichientong and J.B. Orlin. 'Adaptive Data-Driven Inventory Control with Censored Demand Based on Kaplan-Meier Estimator'. *Operations Research* 59.4 (2011), pp. 929–941.

[93] W.T. Huh and P. Rusmevichientong. 'A Nonparametric Asymptotic Analysis of Inventory Planning with Censored Demand'. *Mathematics of Operations Research* 34.1 (2009), pp. 103–123.

[94] IAB Technology Lab. *OpenRTB API Specification Version 2.5*. 2016.

[95] N. Immorlica, B. Lucier, E. Pountourakis and S. Taggart. 'Repeated Sales with Multiple Strategic Buyers'. *Proceedings of the 2017 ACM Conference on Economics and Computation*. 2017, pp. 167–168.

[96] G. Jauvion and N. Grislain. 'Optimal Allocation of Real-Time-Bidding and Direct Campaigns'. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '18. London, United Kingdom: ACM, 2018, pp. 416–424.

[97] G. Jauvion, N. Grislain, P. Dkengne Sielenou, A. Garivier and S. Gerchinovitz. 'Optimization of a SSP's Header Bidding Strategy Using Thompson Sampling'. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '18. London, United Kingdom: ACM, 2018, pp. 425–432.

[98] A. Javanmard. 'Perishability of Data: Dynamic Pricing under Varying-Coefficient Models'. *Journal of Machine Learning Research* 18.53 (2017), pp. 1–31.

[99] S. Kale, L. Reyzin and R.E. Schapire. 'Non-stochastic Bandit Slate Problems'. *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*. NIPS'10. Vancouver, British Columbia, Canada: Curran Associates Inc., 2010, pp. 1054–1062.

[100] A. Kalra, C. Wang, C. Borcea and Y. Chen. 'Reserve Price Failure Rate Prediction with Header Bidding in Display Advertising'. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19. Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 2819–2827.

[101] Y. Kamijo. 'Bidding behaviors for a keyword auction in a sealed-bid environment'. *Decision Support Systems* 56 (2013), pp. 371–378.

[102] R.M. Karp, U.V. Vazirani and V.V. Vazirani. 'An Optimal Algorithm for Online Bipartite Matching'. *Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing*. STOC '90. Baltimore, Maryland, USA: ACM, 1990, pp. 352–358.

[103] G.G. Karuga, A.M. Khraban, S.K. Nair and D.O. Rice. 'AdPalette: an algorithm for customizing online advertisements on the fly'. *Decision Support Systems* 32.2 (2001). Decision Support Issues in Customer Relationship Management and Interactive Marketing for E-Commerce, pp. 85–106.

[104] E. Kaufmann, N. Korda and R. Munos. 'Thompson Sampling: An Asymptotically Optimal Finite-Time Analysis'. *Algorithmic Learning Theory*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 199–213.

[105] N.B. Keskin and A. Zeevi. 'Chasing Demand: Learning and Earning in a Changing Environment'. *Mathematics of Operations Research* 42.2 (2017), pp. 277–307.

[106] R. Kleinberg and T. Leighton. 'The Value of Knowing a Demand Curve: Bounds on Regret for Online Posted-Price Auctions'. *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*. 2003, pp. 594–.

[107] R. Kleinberg and T. Leighton. 'The Value of Knowing a Demand Curve: Bounds on Regret for Online Posted-Price Auctions'. *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*. FOCS '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 594–.

[108] V. Krishna. *Auction theory*. Academic press, 2009.

[109] S. Kumar, V.S. Jacob and C. Sriskandarajah. 'Scheduling advertisements on a web page to maximize revenue'. *European Journal of Operational Research* 173.3 (2006), pp. 1067–1089.

[110] B. Kveton, Z. Wen, A. Ashkan and C. Szepesvari. 'Tight Regret Bounds for Stochastic Combinatorial Semi-Bandits'. *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*. Vol. 38. Proceedings of Machine Learning Research. PMLR, Sept. 2015, pp. 535–543.

[111] T. Lattimore. 'Optimally Confident UCB : Improved Regret for Finite-Armed Bandits'. *CoRR* abs/1507.07880 (2015).

[112] S. Li, B. Wang, S. Zhang and W. Chen. 'Contextual Combinatorial Cascading Bandits'. *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML'16. New York, NY, USA: JMLR.org, 2016, pp. 1245–1253.

[113] Y.-L. Lin and Y.-W. Chen. 'Effects of ad types, positions, animation lengths, and exposure times on the click-through rate of animated online advertisings'. *Computers & Industrial Engineering* 57.2 (2009). Challenges for Advanced Technology, pp. 580–591.

[114] W. Ma and B. Sivan. 'Separation between second price auctions with personalized reserves and the revenue optimal auction'. *Operations Research Letters* 48.2 (2020), pp. 176–179.

[115] S. Mannor and O. Shamir. 'From Bandits to Experts: On the Value of Side-Observations'. *Advances in Neural Information Processing Systems 24*. Ed. by J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira and K.Q. Weinberger. Curran Associates, Inc., 2011, pp. 684–692.

[116] W. Mao, Z. Zheng, F. Wu and G. Chen. 'Online Pricing for Revenue Maximization with Unknown Time Discounting Valuations'. *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. IJCAI'18. Stockholm, Sweden: AAAI Press, 2018, pp. 440–446.

[117] A. Mehta. 'Online Matching and Ad Allocation'. *Foundations and Trends® in Theoretical Computer Science* 8.4 (2013), pp. 265–368.

[118] A. Mehta, B. Waggoner and M. Zadimoghaddam. 'Online Stochastic Matching with Unequal Probabilities'. *Proceedings of the Twenty-sixth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '15. San Diego, California: Society for Industrial and Applied Mathematics, 2015, pp. 1388–1404.

[119] S. Menon and A. Amiri. 'Scheduling Banner Advertisements on the Web'. *INFORMS Journal on Computing* 16.1 (2004), pp. 95–105.

[120] P. Milgrom. *Putting Auction Theory to Work*. Churchill Lectures in Economics. Cambridge University Press, 2004.

[121] K. Misra, E.M. Schwartz and J. Abernethy. 'Dynamic Online Pricing with Incomplete Information Using Multiarmed Bandit Experiments'. *Marketing Science* 38.2 (2019), pp. 226–252.

[122] M. Mohri and A.M. Medina. 'Optimal Regret Minimization in Posted-price Auctions with Strategic Buyers'. *Proceedings of the 27th International Conference on Neural Information Processing Systems*. 2014, pp. 1871–1879.

[123] M. Mohri and A.M. Medina. 'Learning Algorithms for Second-price Auctions with Reserve'. *J. Mach. Learn. Res.* 17.1 (Jan. 2016), pp. 2632–2656.

[124] M. Mohri and A.M. Medina. 'Learning Algorithms for Second-price Auctions with Reserve'. *J. Mach. Learn. Res.* 17.1 (Jan. 2016), pp. 2632–2656.

[125] M. Mohri and A.M. Medina. 'Learning Algorithms for Second-price Auctions with Reserve'. *J. Mach. Learn. Res.* 17.1 (Jan. 2016), pp. 2632–2656.

[126] R.B. Myerson. 'Optimal auction design'. *Mathematics of operations research* 6.1 (1981), pp. 58–73.

[127] M. Ostrovsky and M. Schwarz. 'Reserve Prices in Internet Advertising Auctions: A Field Experiment'. *Proceedings of the 12th ACM Conference on Electronic Commerce*. EC '11. San Jose, California, USA: ACM, 2011, pp. 59–60.

[128] E. Özkan. 'Joint pricing and matching in ride-sharing systems'. *European Journal of Operational Research* (2020).

[129] M. Pachilakis, P. Papadopoulos, E.P. Markatos and N. Kourtellis. 'No More Chasing Waterfalls: A Measurement Study of the Header Bidding Ad-Ecosystem'. *Proceedings of the Internet Measurement Conference*. IMC '19. Amsterdam, Netherlands: Association for Computing Machinery, 2019, pp. 280–293.

[130] S. Paladino, F. Trovò, M. Restelli and N. Gatti. 'Unimodal Thompson Sampling for Graph-structured Arms'. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI'17. San Francisco, California, USA: AAAI Press, 2017, pp. 2457–2463.

[131] D.W. Pentico. 'The assortment problem: A survey'. *European Journal of Operational Research* 190.2 (2008), pp. 295–309.

[132] L. Qin, S. Chen and X. Zhu. 'Contextual combinatorial bandit and its application on diversified online recommendation'. *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM. 2014, pp. 461–469.

[133] R. Qin, Y. Yuan and F. Wang. 'Optimizing the revenue for ad exchanges in header bidding advertising markets'. *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2017, pp. 432–437.

[134] J. Rhuggenaath, R.R. Afshar, A. Akcay, Y. Zhang, U. Kaymak, F. Çolak and M. Tanyerli. 'Maximizing revenue for publishers using header bidding and ad exchange auctions'. Submitted to Operations Research Letters. 2020.

[135] J. Rhuggenaath, A. Akcay, Y. Zhang and U. Kaymak. 'A PSO-based Algorithm for Reserve Price Optimization in Online Ad Auctions'. *2019 IEEE Congress on Evolutionary Computation (CEC)*. June 2019, pp. 2611–2619.

[136] J. Rhuggenaath, A. Akcay, Y. Zhang and U. Kaymak. 'Fuzzy Logic based Pricing combined with Adaptive Search for Reserve Price Optimization in Online Ad Auctions'. *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. June 2019, pp. 1–8.

[137] J. Rhuggenaath, A. Akcay, Y. Zhang and U. Kaymak. 'Optimizing reserve prices for publishers in online ad auctions'. *2019 IEEE Conference on Computational Intelligence for Financial Engineering Economics (CIFEr)*. May 2019, pp. 1–8.

[138] J. Rhuggenaath, A. Akcay, Y. Zhang and U. Kaymak. 'Setting reserve prices in second-price auctions with unobserved bids'. Submitted to INFORMS Journal on Computing. 2020.

[139] J. Rhuggenaath, A. Akcay, Y. Zhang and U. Kaymak. 'Slate bandits with non-separable reward functions'. Submitted to Neurocomputing. 2020.

[140] J. Rhuggenaath, P.R. de Oliveira da Costa, A. Akcay, Y. Zhang and U. Kaymak. 'A heuristic policy for dynamic pricing and demand learning with limited price changes and censored demand'. *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. Oct. 2019, pp. 3693–3698.

[141] J. Rhuggenaath, A. Akcay, Y. Zhang and U. Kaymak. 'Optimal display-ad allocation with guaranteed contracts and supply side platforms'. *Computers & Industrial Engineering* 137 (2019), p. 106071.

[142] J. Rhuggenaath, P.R. de Oliveira da Costa, Y. Zhang, A. Akcay and U. Kaymak. 'Low-regret algorithms for strategic buyers with unknown valuations in repeated posted-price auctions'. *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*. 2020.

[143] J.G. Riley and W.F. Samuelson. 'Optimal auctions'. *The American Economic Review* 71.3 (1981), pp. 381–392.

[144] M.R. Rudolph, J.G. Ellis and D.M. Blei. 'Objective Variables for Probabilistic Revenue Maximization in Second-Price Auctions with Reserve'. *Proceedings of the 25th International Conference on World Wide Web*. WWW '16. Montreal, Quebec, Canada: International World Wide Web Conferences Steering Committee, 2016, pp. 1113–1122.

[145] K. Salomatin, T.-Y. Liu and Y. Yang. 'A Unified Optimization Framework for Auction and Guaranteed Delivery in Online Advertising'. *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. CIKM '12. Maui, Hawaii, USA: ACM, 2012, pp. 2005–2009.

[146] W. Shen, S. Lahaie and R.P. Leme. 'Learning to Clear the Market'. *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research. Long Beach, California, USA: PMLR, Sept. 2019, pp. 5710–5718.

[147] W. Shen, S. Lahaie and R.P. Leme. 'Learning to Clear the Market'. *Proceedings of the 36th International Conference on Machine Learning*. Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. Long Beach, California, USA: PMLR, Sept. 2019, pp. 5710–5718.

[148] A. Slivkins. 'Introduction to Multi-Armed Bandits'. *Foundations and Trends® in Machine Learning* 12.1-2 (2019), pp. 1–286.

[149] M.J. Sobel. 'Myopic Solutions of Markov Decision Processes and Stochastic Games'. *Oper. Res.* 29.5 (Oct. 1981), pp. 995–1009.

[150] M. Stiglic, N. Agatz, M. Savelsbergh and M. Gradisar. 'Making dynamic ride-sharing work: The impact of driver and rider flexibility'. *Transportation Research Part E: Logistics and Transportation Review* 91 (2016), pp. 190–207.

[151] Y. Sun, Y. Zhou, M. Yin and X. Deng. 'On the Convergence and Robustness of Reserve Pricing in Keyword Auctions'. *Proceedings of the 14th Annual International Conference on Electronic Commerce*. ICEC '12. Singapore, Singapore: ACM, 2012, pp. 113–120.

[152] A. Swaminathan, A. Krishnamurthy, A. Agarwal, M. Dudik, J. Langford, D. Jose and I. Zitouni. 'Off-policy evaluation for slate recommendation'. *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 3632–3642.

[153] T.A. Taylor. 'On-Demand Service Platforms'. *Manufacturing & Service Operations Management* 20.4 (2018), pp. 704–720.

[154] W.R. Thompson. 'On the likelihood that one unknown probability exceeds another in view of the evidence of two samples'. *Biometrika* 25.3-4 (1933), pp. 285–294.

[155] L. Tran-Thanh, A. Chapman, A. Rogers and N.R. Jennings. 'Knapsack Based Optimal Policies for Budget-Limited Multi-Armed Bandits'. *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*. AAAI Press, 2012, pp. 1134–1140.

[156] F. Trovò, S. Paladino, M. Restelli and N. Gatti. 'Budgeted Multi–Armed Bandit in Continuous Action Space'. *Proceedings of the Twenty-Second European Conference on Artificial Intelligence*. IOS Press, 2016, pp. 560–568.

[157] F. Trovò, S. Paladino, M. Restelli and N. Gatti. 'Improving multi-armed bandit algorithms in online pricing settings'. *International Journal of Approximate Reasoning* 98 (2018), pp. 196–235.

[158] J. Turner. 'The Planning of Guaranteed Targeted Display Advertising'. *Operations Research* 60.1 (2012), pp. 18–33.

[159] A. Vanunts and A. Drutsa. 'Optimal Pricing in Repeated Posted-Price Auctions with Different Patience of the Seller and the Buyer'. *Advances in Neural Information Processing Systems 32*. 2019.

[160] E. Vee, S. Vassilvitskii and J. Shanmugasundaram. 'Optimal Online Assignment with Forecasts'. *Proceedings of the 11th ACM Conference on Electronic Commerce*. EC '10. Cambridge, Massachusetts, USA: ACM, 2010, pp. 109–118.

[161] J. Wang, W. Zhang and S. Yuan. 'Display Advertising with Real-Time Bidding (RTB) and Behavioural Targeting'. *Foundations and Trends® in Information Retrieval* 11.4-5 (2017), pp. 297–435.

[162] J. Wang, W. Zhang and S. Yuan. 'Display Advertising with Real-Time Bidding (RTB) and Behavioural Targeting'. *Foundations and Trends® in Information Retrieval* 11.4-5 (2017), pp. 297–435.

[163] S. Wang and W. Chen. 'Thompson Sampling for Combinatorial Semi-Bandits'. *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 5114–5122.

[164] Y. Wang, H. Ouyang, C. Wang, J. Chen, T. Asamov and Y. Chang. 'Efficient Ordered Combinatorial Semi-bandits for Whole-page Recommendation'. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI'17. San Francisco, California, USA: AAAI Press, 2017, pp. 2746–2753.

[165] Z. Wen, B. Kveton and A. Ashkan. 'Efficient Learning in Large-scale Combinatorial Semi-bandits'. *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, 2015, pp. 1113–1122.

[166] Z. Xie, K.-C. Lee and L. Wang. 'Optimal Reserve Price for Online Ads Trading Based on Inventory Identification'. *Proceedings of the ADKDD'17*. ADKDD'17. Halifax, NS, Canada: ACM, 2017, 6:1–6:7.

[167] J. Yang, E. Vee, S. Vassilvitskii, J. Tomlin, J. Shanmugasundaram, T. Anastasakos and O. Kennedy. 'Inventory allocation for online graphical display advertising using multi-objective optimization'. *Proceedings of the 1st International Conference on Operations Research and Enterprise Systems - Volume 1: ICORES,* 2012, pp. 293–304.

[168] W. Yang, B. Xiao and L. Wu. 'Learning and pricing models for repeated generalized second-price auction in search advertising'. *European Journal of Operational Research* 282.2 (2020), pp. 696–711.

[169] S. Yuan, J. Wang, B. Chen, P. Mason and S. Seljan. 'An Empirical Study of Reserve Price Optimisation in Real-time Bidding'. *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '14. New York, New York, USA: ACM, 2014, pp. 1897–1906.

[170] W. Zhang, S. Yuan, J. Wang and X. Shen. 'Real-time bidding benchmarking with iPinYou dataset'. *arXiv preprint arXiv:1407.7073* (2014).

[171] H. Zhao and W. Chen. 'Online Second Price Auction with Semi-bandit Feedback Under the Non-Stationary Setting'. *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*. Feb. 2020.

[172] J. Zhao, G. Qiu, Z. Guan, W. Zhao and X. He. 'Deep Reinforcement Learning for Sponsored Search Real-time Bidding'. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '18. London, United Kingdom: ACM, 2018, pp. 1021–1030.

Summary

Revenue management in online markets: pricing and online advertising

In the current economy goods and services are increasingly sold via the internet. Online retailers offer millions of products for sale including books, mobile phones, computers, clothes and various other electronics. There are different types of goods and services that are sold via the internet and there are multiple ways to sell goods and services via the internet. In some cases, potential consumers can browse through the various pages on the website of the seller, where different products are listed together with their prices. If the consumer finds the product that it is looking for and is willing the pay the displayed price, it can proceed to purchase the item by making a payment via the internet. Another popular way to purchase goods on the internet is via an online auction. An auction is a way of selling items, which can be goods or services, that are put up for bid by an auctioneer. In an auction the potential buyers (or bidders) compete with each other by placing bids for an item. The value of the bid indicates the price they are willing to pay for an item. The higher the bid, the better the chance that a bidder will win. With the rise of the internet, auctions have become increasingly important in the domain of online advertising where publishers (owners of websites) sell page views – these are called impressions – to interested advertisers.

The design of the online marketplaces and technological advances have a number of implications for companies that operate on or sell products on online markets. First, it has become possible to store large amounts of information related to various business operations. Second, a lot of transactions in online markets are high volume transactions with a repeated nature. Third, it has become easier and less costly for companies to change key parameters (e.g. prices, website design etc.) that effect sales and revenues.

These developments present both opportunities and challenges for the practice of revenue management. The fact that outcomes at various parameter settings can be logged and stored results in a lot of information. This information provides companies with opportunities in the sense that this information can be leveraged in order to design models and algorithms that can be used for improved decision making in various revenue management problems. However, leveraging this information can also be a challenging task. Most decisions in revenue management problems are

made in uncertain environments and often only partial feedback of these decisions is received. As a consequence, designing models and algorithms that leverage available information is not a straightforward task.

This dissertation studies revenue management in online markets. The revenue management problems that are studied apply to decision making problems that arise in online markets such as online advertisement markets and in e-commerce settings such as online retail markets. The common theme in this thesis is that the decisions in the revenue management problems are made under uncertainty. The revenue management problems that are considered in this thesis can be broadly organized in three categories: allocation decisions, pricing decisions, and buying decisions. In the context of online advertising, these decisions are often related to the decisions of buyers and sellers that participate in online auctions for advertisements and to the interaction of various selling mechanisms that are used in order to sell online advertisements. Some concrete problems that are studied are: (i) how should sellers divide their inventory of impressions (allocation decision) over different selling mechanisms?; (ii) what is the minimum price that sellers should ask (pricing decision) for their impressions?; (iii) at which prices should a buyer purchase (buying decision) an item?

This thesis proposes models and algorithms that can be used for each of these decisions. This thesis uses a combination of techniques from the operations research and computer science communities to tackle the aforementioned revenue management problems. The techniques and results are of interest to anyone that wants to make decisions that involve uncertainty in revenue management problems.

About the Author

Jason Rhuggenaath was born on June 6, 1991 in Willemstad, Curaçao. After finishing his secondary education in 2009 at Radulphus College in Curaçao, he studied Economics and Business Economics at Erasmus University Rotterdam in Rotterdam, The Netherlands. In 2011 he obtained his BSc degree in Economics and Business Economics with honors (cum laude). He then pursued a MSc degree in Economics at Erasmus University Rotterdam, where he obtained his degree with honors (cum laude) in 2014. Afterwards, he pursued a MSc degree in Econometrics and Management Science at the same university, and obtained his degree in 2015. From 2014 to 2016 he worked at CPB Netherlands Bureau for Economic Policy Analysis at the department of macro-economic analysis.

Jason joined the Information Systems (IS) group of the Department of Industrial Engineering and Innovation Sciences at Eindhoven University of Technology (TU/e) in September 2016 as a PhD student under the supervision of prof.dr.ir. Uzay Kaymak, dr. Alp Akçay and dr. Yingqian Zhang. His research interests include data-driven decision-making, operations research, optimization, algorithm design, artificial intelligence and machine learning. In his research, he developed models and algorithms for revenue management problems in online markets. The results of this research are presented in this dissertation. The research reported in this dissertation has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems. In addition, Jason has also taken part in the PhD programme of the LNMB (in Dutch: Landelijk Netwerk Mathematische Besliskunde) and has received the LNMB diploma for PhD courses in the Mathematics of Operations Research. During his PhD, Jason has been involved in several teaching activities related to topics such as machine learning, optimization, business analytics, and data-driven design of service operations at the Jheronimus Academy of Data Science (JADS) and TU/e.

SIKS Dissertation Series

2011 01 Botond Cseke (RUN), Variational Algorithms for Bayesian Inference in Latent Gaussian Models

02 Nick Tinnemeier (UU), Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language

03 Jan Martijn van der Werf (TUE), Compositional Design and Verification of Component-Based Information Systems

04 Hado van Hasselt (UU), Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference

05 Bas van der Raadt (VU), Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline.

06 Yiwen Wang (TUE), Semantically-Enhanced Recommendations in Cultural Heritage

07 Yujia Cao (UT), Multimodal Information Presentation for High Load Human Computer Interaction

08 Nieske Vergunst (UU), BDI-based Generation of Robust Task-Oriented Dialogues

09 Tim de Jong (OU), Contextualised Mobile Media for Learning

10 Bart Bogaert (UvT), Cloud Content Contention

11 Dhaval Vyas (UT), Designing for Awareness: An Experience-focused HCI Perspective

12 Carmen Bratosin (TUE), Grid Architecture for Distributed Process Mining

13 Xiaoyu Mao (UvT), Airport under Control. Multiagent Scheduling for Airport Ground Handling

14 Milan Lovric (EUR), Behavioral Finance and Agent-Based Artificial Markets

15 Marijn Koolen (UvA), The Meaning of Structure: the Value of Link Evidence for Information Retrieval

16 Maarten Schadd (UM), Selective Search in Games of Different Complexity

17 Jiyin He (UVA), Exploring Topic Structure: Coherence, Diversity and Relatedness

18 Mark Ponsen (UM), Strategic Decision-Making in complex games

19 Ellen Rusman (OU), The Mind's Eye on Personal Profiles

20 Qing Gu (VU), Guiding service-oriented software engineering - A view-based approach

21 Linda Terlouw (TUD), Modularization and Specification of Service-Oriented Systems
22 Junte Zhang (UVA), System Evaluation of Archival Description and Access
23 Wouter Weerkamp (UVA), Finding People and their Utterances in Social Media
24 Herwin van Welbergen (UT), Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior
25 Syed Waqar ul Qounain Jaffry (VU), Analysis and Validation of Models for Trust Dynamics
26 Matthijs Aart Pontier (VU), Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots
27 Aniel Bhulai (VU), Dynamic website optimization through autonomous management of design patterns
28 Rianne Kaptein (UVA), Effective Focused Retrieval by Exploiting Query Context and Document Structure
29 Faisal Kamiran (TUE), Discrimination-aware Classification
30 Egon van den Broek (UT), Affective Signal Processing (ASP): Unraveling the mystery of emotions
31 Ludo Waltman (EUR), Computational and Game-Theoretic Approaches for Modeling Bounded Rationality
32 Nees-Jan van Eck (EUR), Methodological Advances in Bibliometric Mapping of Science
33 Tom van der Weide (UU), Arguing to Motivate Decisions
34 Paolo Turrini (UU), Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations
35 Maaike Harbers (UU), Explaining Agent Behavior in Virtual Training
36 Erik van der Spek (UU), Experiments in serious game design: a cognitive approach
37 Adriana Burlutiu (RUN), Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference
38 Nyree Lemmens (UM), Bee-inspired Distributed Optimization
39 Joost Westra (UU), Organizing Adaptation using Agents in Serious Games
40 Viktor Clerc (VU), Architectural Knowledge Management in Global Software Development
41 Luan Ibraimi (UT), Cryptographically Enforced Distributed Data Access Control
42 Michal Sindlar (UU), Explaining Behavior through Mental State Attribution

43 Henk van der Schuur (UU), Process Improvement through Software Operation Knowledge
44 Boris Reuderink (UT), Robust Brain-Computer Interfaces
45 Herman Stehouwer (UvT), Statistical Language Models for Alternative Sequence Selection
46 Beibei Hu (TUD), Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work
47 Azizi Bin Ab Aziz (VU), Exploring Computational Models for Intelligent Support of Persons with Depression
48 Mark Ter Maat (UT), Response Selection and Turn-taking for a Sensitive Artificial Listening Agent
49 Andreea Niculescu (UT), Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality

2012 01 Terry Kakeeto (UvT), Relationship Marketing for SMEs in Uganda
02 Muhammad Umair (VU), Adaptivity, emotion, and Rationality in Human and Ambient Agent Models
03 Adam Vanya (VU), Supporting Architecture Evolution by Mining Software Repositories
04 Jurriaan Souer (UU), Development of Content Management System-based Web Applications
05 Marijn Plomp (UU), Maturing Interorganisational Information Systems
06 Wolfgang Reinhardt (OU), Awareness Support for Knowledge Workers in Research Networks
07 Rianne van Lambalgen (VU), When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions
08 Gerben de Vries (UVA), Kernel Methods for Vessel Trajectories
09 Ricardo Neisse (UT), Trust and Privacy Management Support for Context-Aware Service Platforms
10 David Smits (TUE), Towards a Generic Distributed Adaptive Hypermedia Environment
11 J.C.B. Rantham Prabhakara (TUE), Process Mining in the Large: Preprocessing, Discovery, and Diagnostics
12 Kees van der Sluijs (TUE), Model Driven Design and Data Integration in Semantic Web Information Systems
13 Suleman Shahid (UvT), Fun and Face: Exploring non-verbal expressions of emotion during playful interactions
14 Evgeny Knutov (TUE), Generic Adaptation Framework for Unifying Adaptive Web-based Systems
15 Natalie van der Wal (VU), Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes.

16 Fiemke Both (VU), Helping people by understanding them - Ambient Agents supporting task execution and depression treatment
17 Amal Elgammal (UvT), Towards a Comprehensive Framework for Business Process Compliance
18 Eltjo Poort (VU), Improving Solution Architecting Practices
19 Helen Schonenberg (TUE), What's Next? Operational Support for Business Process Execution
20 Ali Bahramisharif (RUN), Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing
21 Roberto Cornacchia (TUD), Querying Sparse Matrices for Information Retrieval
22 Thijs Vis (UvT), Intelligence, politie en veiligheidsdienst: verenigbare grootheden?
23 Christian Muehl (UT), Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction
24 Laurens van der Werff (UT), Evaluation of Noisy Transcripts for Spoken Document Retrieval
25 Silja Eckartz (UT), Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application
26 Emile de Maat (UVA), Making Sense of Legal Text
27 Hayrettin Gurkok (UT), Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games
28 Nancy Pascall (UvT), Engendering Technology Empowering Women
29 Almer Tigelaar (UT), Peer-to-Peer Information Retrieval
30 Alina Pommeranz (TUD), Designing Human-Centered Systems for Reflective Decision Making
31 Emily Bagarukayo (RUN), A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure
32 Wietske Visser (TUD), Qualitative multi-criteria preference representation and reasoning
33 Rory Sie (OUN), Coalitions in Cooperation Networks (COCOON)
34 Pavol Jancura (RUN), Evolutionary analysis in PPI networks and applications
35 Evert Haasdijk (VU), Never Too Old To Learn – On-line Evolution of Controllers in Swarm- and Modular Robotics
36 Denis Ssebugwawo (RUN), Analysis and Evaluation of Collaborative Modeling Processes
37 Agnes Nakakawa (RUN), A Collaboration Process for Enterprise Architecture Creation
38 Selmar Smit (VU), Parameter Tuning and Scientific Testing in Evolutionary Algorithms

39 Hassan Fatemi (UT), Risk-aware design of value and coordination networks
40 Agus Gunawan (UvT), Information Access for SMEs in Indonesia
41 Sebastian Kelle (OU), Game Design Patterns for Learning
42 Dominique Verpoorten (OU), Reflection Amplifiers in self-regulated Learning
43 Withdrawn
44 Anna Tordai (VU), On Combining Alignment Techniques
45 Benedikt Kratz (UvT), A Model and Language for Business-aware Transactions
46 Simon Carter (UVA), Exploration and Exploitation of Multilingual Data for Statistical Machine Translation
47 Manos Tsagkias (UVA), Mining Social Media: Tracking Content and Predicting Behavior
48 Jorn Bakker (TUE), Handling Abrupt Changes in Evolving Time-series Data
49 Michael Kaisers (UM), Learning against Learning - Evolutionary dynamics of reinforcement learning algorithms in strategic interactions
50 Steven van Kervel (TUD), Ontologogy driven Enterprise Information Systems Engineering
51 Jeroen de Jong (TUD), Heuristics in Dynamic Sceduling; a practical framework with a case study in elevator dispatching

2013 01 Viorel Milea (EUR), News Analytics for Financial Decision Support
02 Erietta Liarou (CWI), MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing
03 Szymon Klarman (VU), Reasoning with Contexts in Description Logics
04 Chetan Yadati (TUD), Coordinating autonomous planning and scheduling
05 Dulce Pumareja (UT), Groupware Requirements Evolutions Patterns
06 Romulo Goncalves (CWI), The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience
07 Giel van Lankveld (UvT), Quantifying Individual Player Differences
08 Robbert-Jan Merk (VU), Making enemies: cognitive modeling for opponent agents in fighter pilot simulators
09 Fabio Gori (RUN), Metagenomic Data Analysis: Computational Methods and Applications
10 Jeewanie Jayasinghe Arachchige (UvT), A Unified Modeling Framework for Service Design.
11 Evangelos Pournaras (TUD), Multi-level Reconfigurable Self-organization in Overlay Services
12 Marian Razavian (VU), Knowledge-driven Migration to Services

13 Mohammad Safiri (UT), Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly
14 Jafar Tanha (UVA), Ensemble Approaches to Semi-Supervised Learning Learning
15 Daniel Hennes (UM), Multiagent Learning - Dynamic Games and Applications
16 Eric Kok (UU), Exploring the practical benefits of argumentation in multi-agent deliberation
17 Koen Kok (VU), The PowerMatcher: Smart Coordination for the Smart Electricity Grid
18 Jeroen Janssens (UvT), Outlier Selection and One-Class Classification
19 Renze Steenhuizen (TUD), Coordinated Multi-Agent Planning and Scheduling
20 Katja Hofmann (UvA), Fast and Reliable Online Learning to Rank for Information Retrieval
21 Sander Wubben (UvT), Text-to-text generation by monolingual machine translation
22 Tom Claassen (RUN), Causal Discovery and Logic
23 Patricio de Alencar Silva (UvT), Value Activity Monitoring
24 Haitham Bou Ammar (UM), Automated Transfer in Reinforcement Learning
25 Agnieszka Anna Latoszek-Berendsen (UM), Intention-based Decision Support. A new way of representing and implementing clinical guidelines in a Decision Support System
26 Alireza Zarghami (UT), Architectural Support for Dynamic Homecare Service Provisioning
27 Mohammad Huq (UT), Inference-based Framework Managing Data Provenance
28 Frans van der Sluis (UT), When Complexity becomes Interesting: An Inquiry into the Information eXperience
29 Iwan de Kok (UT), Listening Heads
30 Joyce Nakatumba (TUE), Resource-Aware Business Process Management: Analysis and Support
31 Dinh Khoa Nguyen (UvT), Blueprint Model and Language for Engineering Cloud Applications
32 Kamakshi Rajagopal (OUN), Networking For Learning; The role of Networking in a Lifelong Learner's Professional Development
33 Qi Gao (TUD), User Modeling and Personalization in the Microblogging Sphere
34 Kien Tjin-Kam-Jet (UT), Distributed Deep Web Search
35 Abdallah El Ali (UvA), Minimal Mobile Human Computer Interaction
36 Than Lam Hoang (TUe), Pattern Mining in Data Streams

37 Dirk Börner (OUN), Ambient Learning Displays
38 Eelco den Heijer (VU), Autonomous Evolutionary Art
39 Joop de Jong (TUD), A Method for Enterprise Ontology based Design of Enterprise Information Systems
40 Pim Nijssen (UM), Monte-Carlo Tree Search for Multi-Player Games
41 Jochem Liem (UVA), Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning
42 Léon Planken (TUD), Algorithms for Simple Temporal Reasoning
43 Marc Bron (UVA), Exploration and Contextualization through Interaction and Concepts

2014 01 Nicola Barile (UU), Studies in Learning Monotone Models from Data
02 Fiona Tuliyano (RUN), Combining System Dynamics with a Domain Modeling Method
03 Sergio Raul Duarte Torres (UT), Information Retrieval for Children: Search Behavior and Solutions
04 Hanna Jochmann-Mannak (UT), Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation
05 Jurriaan van Reijsen (UU), Knowledge Perspectives on Advancing Dynamic Capability
06 Damian Tamburri (VU), Supporting Networked Software Development
07 Arya Adriansyah (TUE), Aligning Observed and Modeled Behavior
08 Samur Araujo (TUD), Data Integration over Distributed and Heterogeneous Data Endpoints
09 Philip Jackson (UvT), Toward Human-Level Artificial Intelligence: Representation and Computation of Meaning in Natural Language
10 Ivan Salvador Razo Zapata (VU), Service Value Networks
11 Janneke van der Zwaan (TUD), An Empathic Virtual Buddy for Social Support
12 Willem van Willigen (VU), Look Ma, No Hands: Aspects of Autonomous Vehicle Control
13 Arlette van Wissen (VU), Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains
14 Yangyang Shi (TUD), Language Models With Meta-information
15 Natalya Mogles (VU), Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare
16 Krystyna Milian (VU), Supporting trial recruitment and design by automatically interpreting eligibility criteria
17 Kathrin Dentler (VU), Computing healthcare quality indicators automatically: Secondary Use of Patient Data and Semantic Interoperability

18 Mattijs Ghijsen (UVA), Methods and Models for the Design and Study of Dynamic Agent Organizations
19 Vinicius Ramos (TUE), Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support
20 Mena Habib (UT), Named Entity Extraction and Disambiguation for Informal Text: The Missing Link
21 Kassidy Clark (TUD), Negotiation and Monitoring in Open Environments
22 Marieke Peeters (UU), Personalized Educational Games - Developing agent-supported scenario-based training
23 Eleftherios Sidirourgos (UvA/CWI), Space Efficient Indexes for the Big Data Era
24 Davide Ceolin (VU), Trusting Semi-structured Web Data
25 Martijn Lappenschaar (RUN), New network models for the analysis of disease interaction
26 Tim Baarslag (TUD), What to Bid and When to Stop
27 Rui Jorge Almeida (EUR), Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty
28 Anna Chmielowiec (VU), Decentralized k-Clique Matching
29 Jaap Kabbedijk (UU), Variability in Multi-Tenant Enterprise Software
30 Peter de Cock (UvT), Anticipating Criminal Behaviour
31 Leo van Moergestel (UU), Agent Technology in Agile Multiparallel Manufacturing and Product Support
32 Naser Ayat (UvA), On Entity Resolution in Probabilistic Data
33 Tesfa Tegegne (RUN), Service Discovery in eHealth
34 Christina Manteli (VU), The Effect of Governance in Global Software Development: Analyzing Transactive Memory Systems.
35 Joost van Ooijen (UU), Cognitive Agents in Virtual Worlds: A Middleware Design Approach
36 Joos Buijs (TUE), Flexible Evolutionary Algorithms for Mining Structured Process Models
37 Maral Dadvar (UT), Experts and Machines United Against Cyberbullying
38 Danny Plass-Oude Bos (UT), Making brain-computer interfaces better: improving usability through post-processing.
39 Jasmina Maric (UvT), Web Communities, Immigration, and Social Capital
40 Walter Omona (RUN), A Framework for Knowledge Management Using ICT in Higher Education
41 Frederic Hogenboom (EUR), Automated Detection of Financial Events in News Text
42 Carsten Eijckhof (CWI/TUD), Contextual Multidimensional Relevance Models

43 Kevin Vlaanderen (UU), Supporting Process Improvement using Method Increments
44 Paulien Meesters (UvT), Intelligent Blauw. Met als ondertitel: Intelligence-gestuurde politiezorg in gebiedsgebonden eenheden.
45 Birgit Schmitz (OUN), Mobile Games for Learning: A Pattern-Based Approach
46 Ke Tao (TUD), Social Web Data Analytics: Relevance, Redundancy, Diversity
47 Shangsong Liang (UVA), Fusion and Diversification in Information Retrieval

---

2015 01 Niels Netten (UvA), Machine Learning for Relevance of Information in Crisis Response
02 Faiza Bukhsh (UvT), Smart auditing: Innovative Compliance Checking in Customs Controls
03 Twan van Laarhoven (RUN), Machine learning for network data
04 Howard Spoelstra (OUN), Collaborations in Open Learning Environments
05 Christoph Bösch (UT), Cryptographically Enforced Search Pattern Hiding
06 Farideh Heidari (TUD), Business Process Quality Computation - Computing Non-Functional Requirements to Improve Business Processes
07 Maria-Hendrike Peetz (UvA), Time-Aware Online Reputation Analysis
08 Jie Jiang (TUD), Organizational Compliance: An agent-based model for designing and evaluating organizational interactions
09 Randy Klaassen (UT), HCI Perspectives on Behavior Change Support Systems
10 Henry Hermans (OUN), OpenU: design of an integrated system to support lifelong learning
11 Yongming Luo (TUE), Designing algorithms for big graph datasets: A study of computing bisimulation and joins
12 Julie M. Birkholz (VU), Modi Operandi of Social Network Dynamics: The Effect of Context on Scientific Collaboration Networks
13 Giuseppe Procaccianti (VU), Energy-Efficient Software
14 Bart van Straalen (UT), A cognitive approach to modeling bad news conversations
15 Klaas Andries de Graaf (VU), Ontology-based Software Architecture Documentation
16 Changyun Wei (UT), Cognitive Coordination for Cooperative Multi-Robot Teamwork
17 André van Cleeff (UT), Physical and Digital Security Mechanisms: Properties, Combinations and Trade-offs
18 Holger Pirk (CWI), Waste Not, Want Not! - Managing Relational Data in Asymmetric Memories

	19	Bernardo Tabuenca (OUN), Ubiquitous Technology for Lifelong Learners
	20	Lois Vanhée (UU), Using Culture and Values to Support Flexible Coordination
	21	Sibren Fetter (OUN), Using Peer-Support to Expand and Stabilize Online Learning
	22	Zhemin Zhu (UT), Co-occurrence Rate Networks
	23	Luit Gazendam (VU), Cataloguer Support in Cultural Heritage
	24	Richard Berendsen (UVA), Finding People, Papers, and Posts: Vertical Search Algorithms and Evaluation
	25	Steven Woudenberg (UU), Bayesian Tools for Early Disease Detection
	26	Alexander Hogenboom (EUR), Sentiment Analysis of Text Guided by Semantics and Structure
	27	Sándor Héman (CWI), Updating compressed colomn stores
	28	Janet Bagorogoza (TiU), Knowledge Management and High Performance; The Uganda Financial Institutions Model for HPO
	29	Hendrik Baier (UM), Monte-Carlo Tree Search Enhancements for One-Player and Two-Player Domains
	30	Kiavash Bahreini (OU), Real-time Multimodal Emotion Recognition in E-Learning
	31	Yakup Koç (TUD), On the robustness of Power Grids
	32	Jerome Gard (UL), Corporate Venture Management in SMEs
	33	Frederik Schadd (TUD), Ontology Mapping with Auxiliary Resources
	34	Victor de Graaf (UT), Gesocial Recommender Systems
	35	Jungxao Xu (TUD), Affective Body Language of Humanoid Robots: Perception and Effects in Human Robot Interaction
2016	01	Syed Saiden Abbas (RUN), Recognition of Shapes by Humans and Machines
	02	Michiel Christiaan Meulendijk (UU), Optimizing medication reviews through decision support: prescribing a better pill to swallow
	03	Maya Sappelli (RUN), Knowledge Work in Context: User Centered Knowledge Worker Support
	04	Laurens Rietveld (VU), Publishing and Consuming Linked Data
	05	Evgeny Sherkhonov (UVA), Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers
	06	Michel Wilson (TUD), Robust scheduling in an uncertain environment
	07	Jeroen de Man (VU), Measuring and modeling negative emotions for virtual training
	08	Matje van de Camp (TiU), A Link to the Past: Constructing Historical Social Networks from Unstructured Data
	09	Archana Nottamkandath (VU), Trusting Crowdsourced Information on Cultural Artefacts

10 George Karafotias (VUA), Parameter Control for Evolutionary Algorithms
11 Anne Schuth (UVA), Search Engines that Learn from Their Users
12 Max Knobbout (UU), Logics for Modelling and Verifying Normative Multi-Agent Systems
13 Nana Baah Gyan (VU), The Web, Speech Technologies and Rural Development in West Africa - An ICT4D Approach
14 Ravi Khadka (UU), Revisiting Legacy Software System Modernization
15 Steffen Michels (RUN), Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments
16 Guangliang Li (UVA), Socially Intelligent Autonomous Agents that Learn from Human Reward
17 Berend Weel (VU), Towards Embodied Evolution of Robot Organisms
18 Albert Meroño Peñuela (VU), Refining Statistical Data on the Web
19 Julia Efremova (Tu/e), Mining Social Structures from Genealogical Data
20 Daan Odijk (UVA), Context & Semantics in News & Web Search
21 Alejandro Moreno Célleri (UT), From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground
22 Grace Lewis (VU), Software Architecture Strategies for Cyber-Foraging Systems
23 Fei Cai (UVA), Query Auto Completion in Information Retrieval
24 Brend Wanders (UT), Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach
25 Julia Kiseleva (TU/e), Using Contextual Information to Understand Searching and Browsing Behavior
26 Dilhan Thilakarathne (VU), In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains
27 Wen Li (TUD), Understanding Geo-spatial Information on Social Media
28 Mingxin Zhang (TUD), Large-scale Agent-based Social Simulation - A study on epidemic prediction and control
29 Nicolas Höning (TUD), Peak reduction in decentralised electricity systems - Markets and prices for flexible planning
30 Ruud Mattheij (UvT), The Eyes Have It
31 Mohammad Khelghati (UT), Deep web content monitoring
32 Eelco Vriezekolk (UT), Assessing Telecommunication Service Availability Risks for Crisis Organisations
33 Peter Bloem (UVA), Single Sample Statistics, exercises in learning from just one example
34 Dennis Schunselaar (TUE), Configurable Process Trees: Elicitation, Analysis, and Enactment

35 Zhaochun Ren (UVA), Monitoring Social Media: Summarization, Classification and Recommendation
36 Daphne Karreman (UT), Beyond R2D2: The design of nonverbal interaction behavior optimized for robot-specific morphologies
37 Giovanni Sileno (UvA), Aligning Law and Action - a conceptual and computational inquiry
38 Andrea Minuto (UT), Materials that Matter - Smart Materials meet Art & Interaction Design
39 Merijn Bruijnes (UT), Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect
40 Christian Detweiler (TUD), Accounting for Values in Design
41 Thomas King (TUD), Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance
42 Spyros Martzoukos (UVA), Combinatorial and Compositional Aspects of Bilingual Aligned Corpora
43 Saskia Koldijk (RUN), Context-Aware Support for Stress Self-Management: From Theory to Practice
44 Thibault Sellam (UVA), Automatic Assistants for Database Exploration
45 Bram van de Laar (UT), Experiencing Brain-Computer Interface Control
46 Jorge Gallego Perez (UT), Robots to Make you Happy
47 Christina Weber (UL), Real-time foresight - Preparedness for dynamic innovation networks
48 Tanja Buttler (TUD), Collecting Lessons Learned
49 Gleb Polevoy (TUD), Participation and Interaction in Projects. A Game-Theoretic Analysis
50 Yan Wang (UVT), The Bridge of Dreams: Towards a Method for Operational Performance Alignment in IT-enabled Service Supply Chains

2017 01 Jan-Jaap Oerlemans (UL), Investigating Cybercrime
02 Sjoerd Timmer (UU), Designing and Understanding Forensic Bayesian Networks using Argumentation
03 Daniël Harold Telgen (UU), Grid Manufacturing; A Cyber-Physical Approach with Autonomous Products and Reconfigurable Manufacturing Machines
04 Mrunal Gawade (CWI), Multi-core Parallelism in a Column-store
05 Mahdieh Shadi (UVA), Collaboration Behavior
06 Damir Vandic (EUR), Intelligent Information Systems for Web Product Search
07 Roel Bertens (UU), Insight in Information: from Abstract to Anomaly
08 Rob Konijn (VU) , Detecting Interesting Differences:Data Mining in Health Insurance Data using Outlier Detection and Subgroup Discovery

09 Dong Nguyen (UT), Text as Social and Cultural Data: A Computational Perspective on Variation in Text
10 Robby van Delden (UT), (Steering) Interactive Play Behavior
11 Florian Kunneman (RUN), Modelling patterns of time and emotion in Twitter #anticipointment
12 Sander Leemans (TUE), Robust Process Mining with Guarantees
13 Gijs Huisman (UT), Social Touch Technology - Extending the reach of social touch through haptic technology
14 Shoshannah Tekofsky (UvT), You Are Who You Play You Are: Modelling Player Traits from Video Game Behavior
15 Peter Berck (RUN), Memory-Based Text Correction
16 Aleksandr Chuklin (UVA), Understanding and Modeling Users of Modern Search Engines
17 Daniel Dimov (UL), Crowdsourced Online Dispute Resolution
18 Ridho Reinanda (UVA), Entity Associations for Search
19 Jeroen Vuurens (UT), Proximity of Terms, Texts and Semantic Vectors in Information Retrieval
20 Mohammadbashir Sedighi (TUD), Fostering Engagement in Knowledge Sharing: The Role of Perceived Benefits, Costs and Visibility
21 Jeroen Linssen (UT), Meta Matters in Interactive Storytelling and Serious Gaming (A Play on Worlds)
22 Sara Magliacane (VU), Logics for causal inference under uncertainty
23 David Graus (UVA), Entities of Interest — Discovery in Digital Traces
24 Chang Wang (TUD), Use of Affordances for Efficient Robot Learning
25 Veruska Zamborlini (VU), Knowledge Representation for Clinical Guidelines, with applications to Multimorbidity Analysis and Literature Search
26 Merel Jung (UT), Socially intelligent robots that understand and respond to human touch
27 Michiel Joosse (UT), Investigating Positioning and Gaze Behaviors of Social Robots: People's Preferences, Perceptions and Behaviors
28 John Klein (VU), Architecture Practices for Complex Contexts
29 Adel Alhuraibi (UvT), From IT-BusinessStrategic Alignment to Performance: A Moderated Mediation Model of Social Innovation, and Enterprise Governance of IT"
30 Wilma Latuny (UvT), The Power of Facial Expressions
31 Ben Ruijl (UL), Advances in computational methods for QFT calculations
32 Thaer Samar (RUN), Access to and Retrievability of Content in Web Archives
33 Brigit van Loggem (OU), Towards a Design Rationale for Software Documentation: A Model of Computer-Mediated Activity
34 Maren Scheffel (OU), The Evaluation Framework for Learning Analytics

35 Martine de Vos (VU), Interpreting natural science spreadsheets
36 Yuanhao Guo (UL), Shape Analysis for Phenotype Characterisation from High-throughput Imaging
37 Alejandro Montes Garcia (TUE), WiBAF: A Within Browser Adaptation Framework that Enables Control over Privacy
38 Alex Kayal (TUD), Normative Social Applications
39 Sara Ahmadi (RUN), Exploiting properties of the human auditory system and compressive sensing methods to increase noise robustness in ASR
40 Altaf Hussain Abro (VUA), Steer your Mind: Computational Exploration of Human Control in Relation to Emotions, Desires and Social Support For applications in human-aware support systems
41 Adnan Manzoor (VUA), Minding a Healthy Lifestyle: An Exploration of Mental Processes and a Smart Environment to Provide Support for a Healthy Lifestyle
42 Elena Sokolova (RUN), Causal discovery from mixed and missing data with applications on ADHD datasets
43 Maaike de Boer (RUN), Semantic Mapping in Video Retrieval
44 Garm Lucassen (UU), Understanding User Stories - Computational Linguistics in Agile Requirements Engineering
45 Bas Testerink (UU), Decentralized Runtime Norm Enforcement
46 Jan Schneider (OU), Sensor-based Learning Support
47 Jie Yang (TUD), Crowd Knowledge Creation Acceleration
48 Angel Suarez (OU), Collaborative inquiry-based learning

2018 01 Han van der Aa (VUA), Comparing and Aligning Process Representations
02 Felix Mannhardt (TUE), Multi-perspective Process Mining
03 Steven Bosems (UT), Causal Models For Well-Being: Knowledge Modeling, Model-Driven Development of Context-Aware Applications, and Behavior Prediction
04 Jordan Janeiro (TUD), Flexible Coordination Support for Diagnosis Teams in Data-Centric Engineering Tasks
05 Hugo Huurdeman (UVA), Supporting the Complex Dynamics of the Information Seeking Process
06 Dan Ionita (UT), Model-Driven Information Security Risk Assessment of Socio-Technical Systems
07 Jieting Luo (UU), A formal account of opportunism in multi-agent systems
08 Rick Smetsers (RUN), Advances in Model Learning for Software Systems
09 Xu Xie (TUD), Data Assimilation in Discrete Event Simulations
10 Julienka Mollee (VUA), Moving forward: supporting physical activity behavior change through intelligent technology
11 Mahdi Sargolzaei (UVA), Enabling Framework for Service-oriented Collaborative Networks

12 Xixi Lu (TUE), Using behavioral context in process mining
13 Seyed Amin Tabatabaei (VUA), Computing a Sustainable Future
14 Bart Joosten (UVT), Detecting Social Signals with Spatiotemporal Gabor Filters
15 Naser Davarzani (UM), Biomarker discovery in heart failure
16 Jaebok Kim (UT), Automatic recognition of engagement and emotion in a group of children
17 Jianpeng Zhang (TUE), On Graph Sample Clustering
18 Henriette Nakad (UL), De Notaris en Private Rechtspraak
19 Minh Duc Pham (VUA), Emergent relational schemas for RDF
20 Manxia Liu (RUN), Time and Bayesian Networks
21 Aad Slootmaker (OUN), EMERGO: a generic platform for authoring and playing scenario-based serious games
22 Eric Fernandes de Mello Araujo (VUA), Contagious: Modeling the Spread of Behaviours, Perceptions and Emotions in Social Networks
23 Kim Schouten (EUR), Semantics-driven Aspect-Based Sentiment Analysis
24 Jered Vroon (UT), Responsive Social Positioning Behaviour for Semi-Autonomous Telepresence Robots
25 Riste Gligorov (VUA), Serious Games in Audio-Visual Collections
26 Roelof Anne Jelle de Vries (UT),Theory-Based and Tailor-Made: Motivational Messages for Behavior Change Technology
27 Maikel Leemans (TUE), Hierarchical Process Mining for Scalable Software Analysis
28 Christian Willemse (UT), Social Touch Technologies: How they feel and how they make you feel
29 Yu Gu (UVT), Emotion Recognition from Mandarin Speech
30 Wouter Beek, The "K" in "semantic web" stands for "knowledge": scaling semantics to the web

2019 01 Rob van Eijk (UL),Web privacy measurement in real-time bidding systems. A graph-based approach to RTB system classification
02 Emmanuelle Beauxis Aussalet (CWI, UU), Statistics and Visualizations for Assessing Class Size Uncertainty
03 Eduardo Gonzalez Lopez de Murillas (TUE), Process Mining on Databases: Extracting Event Data from Real Life Data Sources
04 Ridho Rahmadi (RUN), Finding stable causal structures from clinical data
05 Sebastiaan van Zelst (TUE), Process Mining with Streaming Data
06 Chris Dijkshoorn (VU), Nichesourcing for Improving Access to Linked Cultural Heritage Datasets
07 Soude Fazeli (TUD), Recommender Systems in Social Learning Platforms
08 Frits de Nijs (TUD), Resource-constrained Multi-agent Markov Decision Processes

09 Fahimeh Alizadeh Moghaddam (UVA), Self-adaptation for energy efficiency in software systems
10 Qing Chuan Ye (EUR), Multi-objective Optimization Methods for Allocation and Prediction
11 Yue Zhao (TUD), Learning Analytics Technology to Understand Learner Behavioral Engagement in MOOCs
12 Jacqueline Heinerman (VU), Better Together
13 Guanliang Chen (TUD), MOOC Analytics: Learner Modeling and Content Generation
14 Daniel Davis (TUD), Large-Scale Learning Analytics: Modeling Learner Behavior & Improving Learning Outcomes in Massive Open Online Courses
15 Erwin Walraven (TUD), Planning under Uncertainty in Constrained and Partially Observable Environments
16 Guangming Li (TUE), Process Mining based on Object-Centric Behavioral Constraint (OCBC) Models
17 Ali Hurriyetoglu (RUN),Extracting actionable information from microtexts
18 Gerard Wagenaar (UU), Artefacts in Agile Team Communication
19 Vincent Koeman (TUD), Tools for Developing Cognitive Agents
20 Chide Groenouwe (UU), Fostering technically augmented human collective intelligence
21 Cong Liu (TUE), Software Data Analytics: Architectural Model Discovery and Design Pattern Detection
22 Martin van den Berg (VU),Improving IT Decisions with Enterprise Architecture
23 Qin Liu (TUD), Intelligent Control Systems: Learning, Interpreting, Verification
24 Anca Dumitrache (VU), Truth in Disagreement - Crowdsourcing Labeled Data for Natural Language Processing
25 Emiel van Miltenburg (VU), Pragmatic factors in (automatic) image description
26 Prince Singh (UT), An Integration Platform for Synchromodal Transport
27 Alessandra Antonaci (OUN), The Gamification Design Process applied to (Massive) Open Online Courses
28 Esther Kuindersma (UL), Cleared for take-off: Game-based learning to prepare airline pilots for critical situations
29 Daniel Formolo (VU), Using virtual agents for simulation and training of social skills in safety-critical circumstances
30 Vahid Yazdanpanah (UT), Multiagent Industrial Symbiosis Systems
31 Milan Jelisavcic (VU), Alive and Kicking: Baby Steps in Robotics

	32	Chiara Sironi (UM), Monte-Carlo Tree Search for Artificial General Intelligence in Games
	33	Anil Yaman (TUE), Evolution of Biologically Inspired Learning in Artificial Neural Networks
	34	Negar Ahmadi (TUE), EEG Microstate and Functional Brain Network Features for Classification of Epilepsy and PNES
	35	Lisa Facey-Shaw (OUN), Gamification with digital badges in learning programming
	36	Kevin Ackermans (OUN), Designing Video-Enhanced Rubrics to Master Complex Skills
	37	Jian Fang (TUD), Database Acceleration on FPGAs
	38	Akos Kadar (OUN), Learning visually grounded and multilingual representations
2020	01	Armon Toubman (UL), Calculated Moves: Generating Air Combat Behaviour
	02	Marcos de Paula Bueno (UL), Unraveling Temporal Processes using Probabilistic Graphical Models
	03	Mostafa Deghani (UvA), Learning with Imperfect Supervision for Language Understanding
	04	Maarten van Gompel (RUN), Context as Linguistic Bridges
	05	Yulong Pei (TUE), On local and global structure mining
	06	Preethu Rose Anish (UT), Stimulation Architectural Thinking during Requirements Elicitation - An Approach and Tool Support
	07	Wim van der Vegt (OUN), Towards a software architecture for reusable game components
	08	Ali Mirsoleimani (UL),Structured Parallel Programming for Monte Carlo Tree Search
	09	Myriam Traub (UU), Measuring Tool Bias and Improving Data Quality for Digital Humanities Research
	10	Alifah Syamsiyah (TUE), In-database Preprocessing for Process Mining
	11	Sepideh Mesbah (TUD), Semantic-Enhanced Training Data AugmentationMethods for Long-Tail Entity Recognition Models
	12	Ward van Breda (VU), Predictive Modeling in E-Mental Health: Exploring Applicability in Personalised Depression Treatment
	13	Marco Virgolin (CWI), Design and Application of Gene-pool Optimal Mixing Evolutionary Algorithms for Genetic Programming
	14	Mark Raasveldt (CWI/UL), Integrating Analytics with Relational Databases
	15	Konstantinos Georgiadis (OUN), Smart CAT: Machine Learning for Configurable Assessments in Serious Games
	16	Ilona Wilmont (RUN), Cognitive Aspects of Conceptual Modelling

17 Daniele Di Mitri (OUN), The Multimodal Tutor: Adaptive Feedback from Multimodal Experiences
18 Georgios Methenitis (TUD), Agent Interactions & Mechanisms in Markets with Uncertainties: Electricity Markets in Renewable Energy Systems
19 Guido van Capelleveen (UT), Industrial Symbiosis Recommender Systems
20 Albert Hankel (VU), Embedding Green ICT Maturity in Organisations
21 Karine da Silva Miras de Araujo (VU), Where is the robot?: Life as it could be
22 Maryam Masoud Khamis (RUN), Understanding complex systems implementation through a modeling approach: the case of e-government in Zanzibar
23 Rianne Conijn (UT), The Keys to Writing: A writing analytics approach to studying writing processes using keystroke logging
24 Lenin da Nobrega Medeiros (VUA/RUN), How are you feeling, human? Towards emotionally supportive chatbots
25 Xin Du (TUE), The Uncertainty in Exceptional Model Mining
26 Krzysztof Leszek Sadowski (UU), GAMBIT: Genetic Algorithm for Model-Based mixed-Integer opTimization
27 Ekaterina Muravyeva (TUD), Personal data and informed consent in an educational context
28 Bibeg Limbu (TUD), Multimodal interaction for deliberate practice: Training complex skills with augmented reality
29 Ioan Gabriel Bucur (RUN), Being Bayesian about Causal Inference
30 Bob Zadok Blok (UL), Creatief, Creatieve, Creatiefst
31 Gongjin Lan (VU), Learning better – From Baby to Better
32 Jason Rhuggenaath (TUE), Revenue management in online markets: pricing and online advertising
33 Rick Gilsing (TUE), Supporting service-dominant business model evaluation in the context of business model innovation