Bachelor's thesis

Computing and evaluating pricing strategies in price comparison shopping

Raman Samusevich

May 2014

Advisor: Mgr. Viliam Lisý, MSc.

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Control Engineering

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Control Engineering

BACHELOR PROJECT ASSIGNMENT

Student: **Raman Samusevich**

Study programme: Cybernetics and Robotics
Specialisation: Systems and Control

Title of Bachelor Project: **Computing and evaluating pricing strategies in price comparison shopping**

Guidelines:

1. Study the problem of setting product prices by competing retailers in the environment of online price comparison engines.
2. Design a formal model capturing the most important features of the problem.
3. Implement a computational simulation of the proposed model allowing evaluating quality of different pricing strategies.
4. Survey the existing methods for setting prices in similar models originating from economics, game theory, and control theory. These methods will likely deal with substantially simpler models than the one designed in 2.
5. Design at least two different pricing strategies based on the models found in literature.
6. Experimentally evaluate the quality of the pricing strategies in the simulation and compare it to simple baselines.

Bibliography/Sources:

[1]Lin, Kyle Y., and Soheil Y. Sibdari. "Dynamic price competition with discrete customer choices." European Journal of Operational Research 197.3 (2009): 969-980.
[2] Martinez-de-Albeniz, V. and K. Talluri. "Dynamic Price Competition with Fixed Capacities." Management Science. 2011, Vol. 57, No. 6, pp. 1078-1093.
[3] Kocas_, C. "A Model of Internet Pricing under Price-Comparison Shopping." International Journal of Electronic Commerce. 2005, Vol. 10, No. 1, pp. 111-134.

Bachelor Project Supervisor: Mgr. Viliam Lisý, MSc.

Valid until the summer semester 2014/2015

L.S.

prof. Ing. Michael Šebek, DrSc.
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, January 20, 2014

Acknowledgement

I would like to thank my supervisor Mgr. Viliam Lisý, MSc. for regular and very useful consultations, helpful comments, and for the time Mgr. Viliam Lisý, MSc. kindly devoted to controlling my whole work and making it better. Next, I would like to thank Kristian Hengster-Movric, Ph.D. for his valuable advice on optimal control approach. I would like to thank Víctor Martínez de Albéniz,Ph.D. for sharing related study materials. Last but not least, I would like to thank my family for constant support during my work on the thesis.

Declaration

I declare that I worked out the presented thesis independently and I quoted all used sources of information in accordance with Methodical instructions about ethical principles for writing academic thesis.

Prague, May 21, 2014 ....

Abstrakt

Tato práce se zabývá cenovou soutěží na agregačních stránkách, které umožňují zákazníkům porovnávat různé nabídky produktu na základě cen. Agregační stránky se v dnešní době stávají široce populární. Na základě předchozího empirického výzkumu byl navržen obecný model chování obchodníků a zákazníků. S použitím existujících modelů cenové soutěže bylo navrženo několik strategií cenotvorby. Na základě rozšíření jednoho z modelů byla vyvinuta technika reagování na pozorované ceny konkurence. V kombinaci s vyvinutou technikou byly úspěšně použity metody teorie her, optimálního řízení a dynamického programování. Jako výsledek byly získány další strategie cenotvorby. Tyto strategie modelují kapacitu obchodníků a časovou strukturu cenové soutěže. Byla realizována výpočetní simulace soutěže. Taky byla navržena metoda jednotného hodnocení kvality různých strategií. Simulace ukázaly, že uvažovat pouze cenu konkurence může být prospěšné jen z krátkodobého hlediska, z dlouhodobého hlediska to nutno brát v úvahu více parametrů. V simulacích se strategie vycházející z učení podle herně-teoretické metodiky fictitious play v kombinací s dynamickým programováním ukázala jako obecně nejefektivnější.

Klíčová slova

(cenová soutěž, agregační stránky, teorie her, optimální řízení, dynamické programování)

Abstract

This thesis investigates price competition at comparison shopping engines which are becoming widely popular nowadays. Based on empirical related work a general model of retailers' and customers' behaviour is proposed. Using existing models of price competition several pricing strategies for the studied problem are obtained. One related model is extended and a technique for response to observed competitors' behaviour is suggested. The developed technique was successfully combined with methods of game theory, optimal control and dynamic programming. As a result, further pricing strategies were obtained. The strategies model retailer's capacity and the temporal structure of the price competition. A computational simulation of the competition was implemented and the method of unified numerical evaluation of different strategies was suggested. Using simulation it was found out that strategies based purely on observation of prices might be efficient only in short-term competition. To be successful in long-term competition, a retailer has to consider further characteristics of the problem. Using simulations the strategy based on fictitious play in combination with dynamic programming was found to be in general the most efficient. Factitious play is a game-theoretic learning rule.

Keywords

(price competition, shopbots, game theory, optimal control, dynamic programming, simulation)

Contents

1. Introduction

Development of information technology has influenced the forms of retailing. Online shopping has become widely popular among both adults and adolescents [1]. According to Czech Statistical Office, the internet retailing share of a total Czech retail was 5-6% in 2011 with revenues of 40-45 billion CZK. It is predicted that this share will grow. As the popularity of online retailing among customers is considerable, even big classic stores sell their goods via the Internet as well [2].

One reason for the popularity of online retailing is the possibility to compare different offers of a particular product and choose the most preferable one [3]. For the purpose of easier comparisons special engines have been developed. In the literature there are several terms describing essentially the same online comparison-shopping engines. These terms are: product comparison agents, price comparison sites, shopbots, comparison-shopping agents/engines, recommendation or buyer's agents, internet shopping agents and price aggregators [4, 5, 6, 7]. In this work a term *shopbot* will be used.

Shopbots are automated tools which provide customers with comparison of prices or other attributes of a particular product offered by different online shops [8]. Shopbots increase price transparency [7]. It leads to intensive price competition among retailers at shopbots [3]. Initially online retailers tried to avoid the price competition and did not join shopbots. However, nowadays retailers actively join them [3], because shopbots have become very popular among customers [4, 3]. Analysis of the price competition at shopbots and developing effective pricing strategies might be very important for successful online retailing.

In this work a general model of the price competition at shopbots is formulated. Based on existing related models of price competition, several pricing strategies are obtained. One existing model is extended in order to become as close to the general model as possible. A technique for learning from observation is developed for the extended model. The technique is successfully combined with methods of game theory, optimal control and dynamic programming. As a result, further pricing strategies are designed. The strategies consider retailer's capacity and the temporal structure of the price competition. Moreover, observation of adversaries' inventory levels is not required. A computational simulation of the price competition is implemented and performance of developed strategies is evaluated numerically. A set of experiments required for unified comparative evaluation of different strategies is proposed. In experiments it is shown that pricing strategies using observations of previous prices might be efficient only in short-term competition. In order to maximize profits in long-term competition, it is important to consider possible loyalty of customers, an inventory level, duration of sales and a total expected demand. It is shown that the strategy based on fictitious play in combination with dynamic programming is in general the most efficient.

The following section describes a structure of the work.

1.1. Outline of the thesis

In Chapter 2 the complex problem of pricing at shopbots is investigated systematically. Based on related empirical research key components and the most important characteristics of the price competition among retailers are discussed. A general model of retailers' and customers' behaviour is provided. In Chapter 3 related models of similar competitions on prices are surveyed. Different possible approaches to studying price competition are discussed. Limitations of these approaches are considered. Chapter 4 shows that it is possible to design pricing strategies based on the scientific models from the previous chapter. In the next Chapter 5 a more realistic extension of one model is developed based on the general model from Chapter 2. Using the developed extension further pricing strategies are designed in Chapter 6 based on methods of game theory, optimal control and dynamic programming. In order to evaluate the performance of different pricing strategies, a computational simulation of the price competition at shopbots is used in Chapter 7.

2. Price competition among retailers at a shopbot and its model

Each retailer, who participates in price competition at a shopbot, tries to maximise his profit. The aim of this work is to develop pricing strategies, which would maximise a retailer's profit. A real-world behaviour of the retailer is very complex. While selling a product at a shopbot the retailer has to deal with a lot of issues (transport and storing of the product, collecting information about adversaries and customers, updating a price etc.). It might be impossible to develop a pricing strategy which takes into consideration all real-world issues. The more real-world information would be taken into account, the better the performance of the strategy is expected to be. It is important to gather the most relevant information about shopbots, customers' behaviour at shopbots, and behaviour of retailers at a shopbot. This information organized systematically will provide a general model of the price competition at a shopbot. The model will apparently be too complex to obtain any pricing strategy. Models for developing pricing strategies will be significantly simpler than the general model. But after developing the general one it will be possible to understand limitations of simpler models better.

2.1. Functions of online comparison-shopping engines

The purpose of this work is to analyse price competition among retailers at shopbots. Services provided by shopbots shape the competition among retailers and should be inspected. On the other hand, neither technical realization of shopbots, nor algorithmic aspects of shopbots' functions would not be considered.

Shopbots provide customers with comparison of prices or other attributes of products available online. Shopbots are capable of collecting information from the retailers, storing and processing the stored information on prices or further product attributes [4]. Shopbots also allow customers to rate retailers. Shopbots store customers' ratings and reviews [9]. All information about offers and retailers is public [10]. Additionally, shopbots can provide retailers with analysis of the demand each retailer faces [11].

The information on prices and availability is updated periodically. The maximal frequency of price updates might be given by rules of the shopbot. For instance, according to the rules of a Czech shopbot heureka.cz [10], prices are updated several times per day. Google Shopping[1] shopbot advises customers to update information on prices and availability four times per day [13]. If four updates per day is not enough more frequent changes of prices and availability can be enabled [13]. Shopbots are especially interested in presenting up-to-date information on availability. Heureka.cz updates information on availability every ten minutes [14].

Even though many shopbots provide customers with comparison of various product attributes, price has always been the most common one. Historically the very first shopbot developed in 1995 compared only prices on available online CDs [15]. Nowadays many shopbots offer not only a comparison of prices but also a comparison of subjective customers' experiences [4, 10]. Heureka.cz in addition compares even such a specific piece of

[1] According to the most recent reports of CPC Strategy, a shopbots management company, Google Shopping is currently the top-performing shopbot [12].

information as a distance to a shop. Interestingly, it was shown that additional comparison information does not necessarily result in more visitors of a shopbot [16]. Some price comparison shopbots can compare total prices a particular customer would have to pay for a particular product. The total price includes a shipping cost and taxes. As an example, price comparison shopbot Google Shopping offers customers comparison of the total prices a customer would have to pay.

Deeper analysis of shopbots is not a subject of this work. The brief overview presented here should be enough for further analysis of a price competition among retailers at shopbots.

2.2. Behaviour of retailers at shopbots

2.2.1. Results of empirical research

Results of empirical research on pricing behaviour of competing retailers at shopbots are to be discussed.

It was shown that retailers periodically change their prices. Using a comprehensive set of empirical data it was found that top-ranked retailers adjust their prices quite often [3, 5]. Retailers who set the lowest prices are meant by the top-ranked retailers. Retailers with median positions 2nd - 5th had to make more than 3000 adjustment during a studied period (12 months), i.e. on average more than 8 adjustments per day [3]. Non-top retailers do not adjust their prices so often [3]. They might be big and very reputable retailers. It was predicted that shopbots should not affect pricing behaviour of reputable retailers very much [6]. On the average a retailer at the shopbot adjusted his price every 9 days [3]. Only 2% of all retailers at the studied shopbot have never been placed on the first page among the cheapest retailers and no single retailer managed to hold the first position [3].

At some point in time prices of different retailers at a shopbot vary significantly. In other words, it has been shown that prices at a shopbot are strongly dispersed [17, 3, 18]. Neither level of price dispersion, nor price level at shopbots are lower compared to the level of offline market [3].

Using empirical data, it was shown that retailers' behaviour at a shopbot can be described using a probabilistic pricing [5, 6], when every retailer has defined probabilities of setting each possible value of a price.

2.2.2. Set of prices

Possible values of a price for an arbitrary product are to be considered. One particular item from a retailer's list of offered goods or services is meant by a product. For instance, in the work the term *product* will be used for plane tickets on a particular date of flight, and for particular book, and for particular laptop model, and for hotel rooms on a particular date, etc. Let a *piece of a product* be a single unit of the product. A price for a single piece of the product will be meant by a price for a product.

Setting a negative price for a product is usually not desirable, as well as setting a price which is lower than a product cost[2]. If a retailer produces a product, then the product cost might be viewed as an average cost of the production and storing of one piece of a product. If a retailer does not produce a product, then the product cost might be viewed as a sum of the factory price, the shipping price, taxes, and the price of storing a single piece of a product. Let $\texttt{cst}$ denote the product cost, $\texttt{cst} \in \mathbb{R} \wedge \texttt{cst} > 0$. For additional remarks on the meaning of $\texttt{cst}$ in the general model see Subsection 2.2.4. A term "zero profit price", $p_{\texttt{zero profit}}$, will be used for a product price which equals to a product cost.

[2]Theoretically, though, both cases are possible in competitions similar to the one at shopbots [19].

In the online market there is the highest reasonable price $p_{\texttt{highest}}$ [5, 19, 6]. It is such a price, that no customer would buy a product at a higher price $p_{\texttt{H}} > p_{\texttt{highest}}$.

A set of actions available to every player is setting a reasonable price p, $p \in [p_{\texttt{zero profit}}, p_{\texttt{highest}}]$. Note, that in a case if $p_{\texttt{highest}} \leq p_{\texttt{zero profit}}$ a retailer cannot make any profit from participating in a price competition. Retailer are assumed to be rational and, as a consequence, $p_{\texttt{highest}} > p_{\texttt{zero profit}}$.

Also we assume that a retailer sets a price from an interval $[p_{\texttt{zero profit}}, p_{\texttt{highest}}]$ only if he has product pieces to sell. If a retailer has nothing to sell, he automatically quits a shopbot. A shopbot is not interested in displaying outdated information to customers [14, 13]. If a retailer has no product pieces at his storage, then his offer is automatically unavailable to customers. It means that no customer can buy from the retailer.

All retailers, who compete with each other at a shopbot, offer the same product. The product cost is assumed to be the same for all retailers. As a consequence, the price $p_{\texttt{zero profit}}$ is the same for all competing retailers.

2.2.3. Capacity constraints a retailer faces

We have discussed that a set of possible prices is constrained. In the real-world situations every retailer also faces capacity constraints. Usually a retailer has a product storage with a fixed capacity. For instance, a warehouse might be the storage. Also every retailer has some limited amount of available money, it also constrains a number of product pieces the retailer can buy. To sum up, every retailer has a maximal number of product pieces he is capable of buying and storing. Let this maximal number be called a *retailers' capacity*, as it was done in [20, 21]. Capacities of different retailers in general differ.

Let an *inventory level* be a number of pieces a retailer currently has. The notation is taken from [20, 21]. An inventory level cannot be negative. This is another capacity constraint each retailer faces.

2.2.4. Temporal structure of retailer's actions

If in real-world situations a retailer sells everything out, he cannot refill his inventory level with a new portion of products at any moment. In practice new portions are delivered periodically. As a result, a retailer faces a problem of selling a limited number of products over a finite time horizon [20, 21, 19].

At the end of the time horizon a retailer can refill an inventory level up to a number of products equal to the retailer's capacity. After the refilling the retailer faces the same problem of selling his capacity over the time horizon before the next refilling of the inventory level. As the problems are analogous, it is sufficient to consider only a time interval between two consequent refillings, as it was done in [20, 19, 21]. The time interval was called a *sales horizon* in [20, 21]. The same term will be used in this work.

During a sales horizon a retailer at a shopbot is involved in a dynamic price competition. He changes prices periodically [3]. It can be assumed that all retailers at a shopbot update prices in the same periods [20, 19]. When some retailer does not update his price the situation is identical as if he updated a price but set the same value as he had before the update. The frequency of price updates might be given by the rules of a shopbot [10]. Alternatively, it can be assumed that prices of all retailers are updated with a frequency of the most active retailers, which is observable by all retailers [3]. Every update of prices starts a new period of competition among retailers. Inventory levels of successful retailer decreases as the competition goes on.

Let a *total profit* of a retailer be equal to a sum of all retailer's profits during the sales horizon, the same definition of a total profit was used in [21][3]. If at some moment t a customer buys a piece of product from the retailer at price $\texttt{price}(t)$, then retailer's profit at that moment would be $\texttt{profit}_{\texttt{moment}} = (\texttt{price}(t) - \texttt{cst})$.

A period of competition and a sales horizon should be distinguished. A sales horizon is usually a longer time interval compared to a period of competition. It is often the case as products are not usually delivered more often than once per day, and usually several changes of prices take place during a day [3]. It is assumed that a sales horizon duration can be expressed as a multiple of a competition period, as it was done in [20, 19], see Fig. 3.1.1.

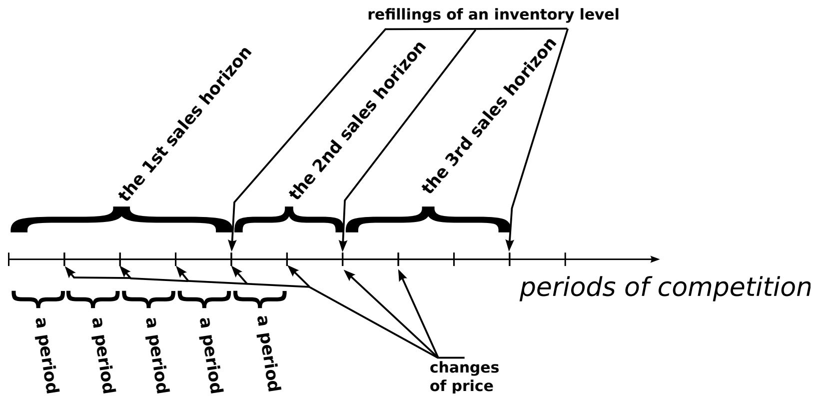

Fig. 2.2.1. Illustration of the difference between periods of competition and sales horizons

Note, that a retailer has to pay some holding cost for storing one item of a product during the sales horizon [21]. In the model it is assumed that the holding cost over one sales horizon is included in the product cost, cst. However, if a piece of a product is not sold during one sales horizon, then the retailer will have to pay additional holding cost for the next sales horizon. The length of the next sales horizon might be unknown. Nevertheless, in the model the cost for holding a piece of a product over the next sales horizon is assumed to be fixed, as it has been done in [21].

A comment on alternative definitions of the competition period is required. In related work periods of the competition were defined with arrivals of customers [20, 19]. In both [19] and [20] arrivals of customers were assumed to be observable by all retailers. If a customer arrives to any of the retailers, then a new period of the competition starts and retailers might want to change their prices.

In real-world situations a retailer usually cannot observe arrivals of customers to other retailers.

2.2.5. Information a retailer can observe

A retailer can observe several pieces of information.

a) Previous prices all other retailers at a shopbot might be observed.

b) A number of customers who previously bought from the retailer at some particular price might be observed.

c) In some cases it is possible to observe a capacity of other retailers. For instance, in a hotel business the initial capacities of hotel rooms are known. Nowadays hotels sell their rooms via shopbots as well. Another example is an airline business. A number of seats at a particular flight is usually known. There are many shopbots specifically for airline tickets.

[3] Sometimes a synonymous term *total revenue* is used [20, 21].

d) It might be possible to observe sales horizons of other retailers. For instance, in a case of airlines everybody usually knows a start date of sales of the tickets for a particular flight on a particular date. And the date of the flight is also known, it terminates a sales horizon.

Also it is possible to observe sales horizons of other retailers in a case of the hotel business. Hotels often sell reservations for their rooms for a particular weekend [22]. The sales of the reservations usually start a few months in advance. Sales horizon in this case is equal to a time interval between the start of sales and a date of the considered weekend. And all other competing hotels in the same area with similar rooms for the weekend can observe sales horizons of their competitors.

d) It might rarely be possible to observe other retailers' inventory levels. For instance, it is so in the case of an airline business.

When it is not the case, sometimes a retailer might be capable of an approximate estimation of other retailers' inventory levels, see [20, 19] for details.

2.2.6. Pseudo code of a retailer

The discussed retailer's behaviour during a single sales horizon will be formally described with pseudo code, see Algorithm 1. Note that after the end a sales horizon the retailer would pay a cost for holding the remaining pieces of a product. At the same time the retailer can refill his inventory level.

In the general model it is assumed that each customer buys just one piece of a product, see Subsection 2.3.3.

Note, that in Algorithm 1 function *ComputePrice(*`knowledge`, **Parameters***)* has to return only allowed values of prices defined in the Subsection 2.2.2.

Interaction between customers and retailers in the pseudo code is given as a test for customer's purchase from a considered retailer. Customers' behaviour is considered in the next section.

2.3. Customers' behaviour at shopbots

One of key goals of each retailer at a shopbot is to attract customers, as a retailer would earn nothing if no customer decides to buy from him. Therefore, in order to understand better which actions retailers should take, we need to examine factors that influence costumers' choices.

2.3.1. Offer attributes which influence a customer's choice at shopbots

Using a comprehensive set of empirical data it has been found that 90% of all clicks at the studied shopbot were obtained by retailers who had prices p such that $p \leqslant 1.05\ p_{\texttt{min}}$, where $p_{\texttt{min}}$ is the minimal price available at the shopbot [3]. So we can see that costumer choices are strongly influenced by prices.

It was also shown that having two similar offers with the same price more customers prefer a retailer with a stronger brand [7]. It should be noted that by a retailer's brand the retailer's reputation is meant. Even though a retailer's brand matters, still a product price is a primary determinant of choices for many customers [23].

2.3.2. Loyal and switching customers

In some works price competition among retailers at a shopbot is modelled as an all-or-nothing type of the competition, when all customers buy from retailers who offer the lowest

```
Input: Observations, see 2.2.5
Result: A retailer sets a price for the offered product
Parameters: the first period = period after a refilling of an inventory
            level, a capacity, an inventory level, a product cost = cst,
            a profit, a sales horizon
Variables: knowledge

initialization:
  knowledge = {capacities and sales horizons of other retailers, information on other
retailers' inventory levels, Observations};
while (current period - the first period) < sales horizon do
    if inventory level ≥ 1 then
        knowledge = knowledge + current period;
        price = ComputePrice(knowledge, Parameters);
        set a price price for the product;
        repeat
            if a customer buys a piece of the product then
                inventory level = inventory level - 1;
                profit = profit + price - cst;
                if inventory level = 0 then
                    quit the shopbot;
                end
            end
        until the end of the current period;
    else
        quit the shopbot;
    end
    knowledge = knowledge + Observations;
    current period = current period + 1;
end
```

Algorithm 1: Pseudo code modelling a retailer's behaviour at a shopbot

price [5, 19, 6]. These models of the price competition predict a demand jump for a retailer who starts to offer the lowest price. The demand jump has been observed using a set of empirical data [24].

In real-world situations if a retailer starts to offer the lowest price at a shopbot, he does not experience a jump from a zero demand to a total demand. For instance, in the empirical research [24] 45% of all clicks were obtained by a retailer who offered the lowest price[4]. More than a half of all customers did not choose the cheapest offer. In the work [24] it is explained by the fact that some customers have their preferred retailers at the shopbot [24]. Such customers are often called loyal customers or simply the *loyals* [24, 5, 6]. It is assumed that the loyals buy from their preferred retailers as long as the price is reasonable, i.e. as long as price $p \leq p_{\mathtt{highest}}$. Not all customers at a shopbot can be modelled as the loyals. Another type of shopbot customers are *shoppers*, or *switching customers*, or simply the *switchers*. The switchers always choose the lowest price [24, 5, 6].

The concept of the loyals relates to the concept of brand-sensitive customers. It has been shown that some customers are brand-sensitive, i.e. they would like to buy from reputable retailers even at prices higher than the minimal ones [7]. The concept of the

[4]Note, that it has been shown that clicks data can be used for identification of the demand characteristics [24].

loyals and the switchers also predicts that some customers would choose offers at prices higher than the minimal one. It is important to note that the empirical study [24] which has shown the credibility of the loyals/switchers model explains an existence of the loyals by brand-sensitivity of customers. Specifically, it has been shown that reputable online retailers which also have classic stores attract more customers compared to purely online retailers [24].

The retailer's brand can be understood as a general characteristic of the previous customers' experience. The brand is often expressed via forums, customers' reviews and rating systems [10]. The greater the retailer's brand is, the more satisfied the customers were with the retailer's services. And if the customers had been satisfied it is very likely that they became loyal to the retailer.

The behaviour of one customer in the general model is concisely described in the next section.

2.3.3. Summation on customers' behaviour

In the general model it is assumed that each customer buys just one piece of a product. The same assumption was made in models of price competition among retailers in [25, 19, 20].

Demand is usually decreasing in price. It is assumed that at the shopbot there is such a price $p_{\texttt{highest}}$, that no customer would buy a product at a higher price $p_{\texttt{H}} > p_{\texttt{highest}}$ [5, 19]. If all available prices are higher than $p_{\texttt{highest}}$, then a customer buys nothing and leaves. If a customer does not trust any of retailers at a shopbot due to the described loyalty, then the customer also buys nothing. In other cases a customer takes into consideration available prices and his own preferences and buys a piece of a product from some retailer. If a purely price sensitive customer faces several offers at the same price, then he chooses one offer at random [19].

2.3.4. Arrivals of customers

A customer's behaviour has been modelled. At different moments of time there are different numbers of customers at a shopbot. Arrivals of customers can be described as a stochastic process [19, 24, 20, 21]. A number of arrived customers during a particular period of the competition is a value of a discrete random variable. Let $\boldsymbol{Y}$ denote this random variable.

$\boldsymbol{Y}$ assigns a natural number of arrived customers to every possible period of the competition. Note that all periods of the same length which can take place are meant by possible periods of the competition. All these possible periods make up a sample space $\boldsymbol{Y}$ is defined on.

There are several stochastic processes which can describe arrivals of customers at a shopbot. The Bernoulli process is often used to model arrivals of customers ([26], p. 297). In real-world situations customers do not necessarily arrive one by one in discrete time periods. So arrivals of customers might be better modelled with the Poisson process which is a continuous-time analogue of the Bernoulli process ([26], p. 309). Both processes are used in the literature to model arrivals of customers. In [21] the Poisson process was used for analysis of a problem similar to ours. But in both [19] and [20] the Bernoulli process was used. It was argued that the Bernoulli process approximate the Poisson process well if one chooses an appropriate time scale [20, 19],([26], p. 311).

The Bernoulli process is easier to implement in numerical simulations of the competition at a shopbot. Studying of underlying stochastic process in detail is not the aim of this work. The goal of this work is to develop retailers' strategies, for this purpose the Bernoulli process of arrivals of customers would serve well.

The period of customer arrivals is not equal to the period of the competition in the model. In the model it is assumed that several customers might arrive during the single period of the competition.

The behaviour of a customer at a shopbot has been described in Subsection 2.3.3. In the model each customer performs the described behaviour as soon as he arrives to a shopbot. Pseudo code 2 captures in a concise manner a process of customers' arrivals. In the code `start` is the first period of the competition at a shopbot which would be considered.

```
Variables: current number of arrivals, a natural number k
for current period = start to infinity do
    current number of arrivals = Y(current period);
    k = current number of arrivals;
    while k > 0 do
        a new customer arrives;
        k = k - 1;
    end
end
```

Algorithm 2: Pseudo code modelling customers' arrivals to a shopbot

To conclude, the most important characteristics of the price competition among retailers at a shopbot have been underlined. The general model of retailers' reasoning and customers' behaviour is concisely given in Subsection 2.3.3 and in Algorithms 1 and 2. The model considers the most important real-world attributes of the price competition at a shopbot.

3. Related work on models of the price competition at shopbots

There are several possible ways to study the competition at shopbots. Game-theoretic methods can be used for this purpose, as retailers at shopbots might be viewed as competing rational decision-makers. Game theory is a study of strategic interactions among rational, independent and self-interested decision-makers [27]. Game-theoretic models of price competition are described in Section 3.1.

Another possible way to address the competition at shopbots is to use optimal control. Using variations in price, each retailer tries to control a number of sold products in such a way, that his profit is maximised. Section 3.2 describes related work in which optimal control methods were used to analyse dynamic price competition among several retailers.

3.1. Game-theoretic approach

Dealing with game-theoretic models related to the competition at shopbots some definitions from the field of game theory will be used. These definitions can be found in Appendix A.

3.1.1. Bertrand model

The basic economic model developed by Bertrand is one of the oldest models of price competition similar to the competition at shopbots.

Retailers at a shopbot offer the same product, i.e. all retailers sell an identical product. Selling a homogeneous product is an assumption of the Bertrand model [28]. As a consequence, the model assumes that the price $p_{\texttt{zero profit}}$ is the same for all retailers. In the Bertrand model customers are price-sensitive and buy from the retailer, who offers the lowest price [28]. Retailers' capacities are not considered in the model. Basic results of the model will be shown for a duopoly, i.e. a competition between two retailers [28, 29]. In the model it is assumed that every retailer can meet the total demand. The competition is modelled as a single-stage game.

It was shown that in the only Nash equilibrium of the Bertrand competition every retailer sets a price equal to the product cost, $p_{\texttt{zero profit}}$ [28]. Having a duopoly with retailers A and B the best response of a retailer B to a price p_A of a retailer A is

$$BR(p_A) = \begin{cases} p_A - \epsilon, & p_A - \epsilon > p_{\texttt{zero profit}}; \\ p_{\texttt{zero profit}}, & otherwise. \end{cases}$$

The result is taken from [29]. The only Nash equilibrium is given by $p_A^* = p_B^* = p_{\texttt{zero profit}}$ [29]. It is impossible to improve the profit by setting a price lower or higher than $p_{\texttt{zero profit}}$ if another retailer sets $p_{\texttt{zero profit}}$.

The result holds in the case of more retailers [30].

3.1.2. Bertrand competition in the context of shopbots

It was shown that under certain conditions the competition among retailers at shopbots might be the Bertrand competition, i.e. all competing retailers in the equilibrium would set the price $p_{\texttt{zero profit}}$ [18, 31]. The assumptions for this were: **a)** a low customer search cost and price-sensitive customers, **b)** a large number of retailers at a shopbot [31]. Nowadays customer search at shopbots is very fast and free, it can be viewed as a zero cost search. A majority of customers are price sensitive [23]. Price-sensitive customers are fully informed and can choose an offer at the lowest price. The number of retailers participating in shopbots is large nowadays. In this case a retailer's competitors can meet the total demand. To sum up, the price competition at a shopbot could take a form of the Bertrand competition. As a result, initially it was expected that in equilibrium all retailers would set the same low prices [31].

However, in empirical studies it was shown that prices of different retailers at shopbots vary significantly [17, 3, 18]. The price level at shopbots is not lower compared to the price level on the same product at classic stores [3]. For that reason further studies of the problem were required.

3.1.3. Alternative models of the competition

To explain an observed behaviour of retailers at shopbots, new models of the price competition were developed. Some of these models deal with a concept of loyal customers and switching customers, see Subsection 2.3.2.

Using the concept of switching and loyal customers, it was shown that an average price at a shopbot might not decrease comparing to offline market price [6, 5]. The models predict that retailers at shopbots would have mixed strategies. These theoretical results were verified using empirical data [6, 5].

A game-theoretic model considering loyal and switching customers will be described in Subsection 3.1.4.

The possibility of non-zero profit equilibrium in price competition similar to the one at shopbots can be explained in another way. In works [19] and [20] an extensive-form game is suggested. The real-world temporal structure of the competition and the retailers' capacity are taken into consideration. Extensive-form game models will be considered in sections 3.1.5 and 3.1.6.

3.1.4. Normal-form game model developed by Koçaş [5]

Koçaş [5] analyses price competition among retailers with different numbers of loyal customers. It was shown that only retailers with a small number of loyal customers, compared to a number of their addressable switching customers, would in equilibrium actively participate in price competition. The retailers were shown to adopt mixed strategies. In the model developed by Koçaş all competing retailers have similar small numbers of loyal customers compared to a number of switching customers they can address.

Let $p_{\texttt{min}}$ be the lowest price a retailer would ever consider to set. In [5] the price $p_{\texttt{min}}$ is calculated using simple reasoning. The retailer should be indifferent between selling pieces of a product to his loyal customers at $p_{\texttt{highest}}$ price or selling to both loyal and switching customers but at the $p_{\texttt{min}}$ price. In the general model the retailer's profit from selling a piece of a product at price p_i is given by $(p_i - \texttt{cst})$, where cst stands for a product cost[1]. Let χ be a number of retailer's loyal customers and let ψ be a number of his addressable switching customers. It is assumed that $\chi \geq 1$. Otherwise a retailer will act according to

[1]See 2.2.2 and 2.2.4 for details on product cost cst. Koçaş assumed a zero product cost and obtained slightly different formulas.

the Bertrand model. $(p_{\mathtt{min}} - \mathtt{cst})(\chi + \psi) = \chi \cdot (p_{\mathtt{highest}} - \mathtt{cst})$. Therefore,

$$p_{\mathtt{min}} = \mathtt{cst} + \chi \frac{(p_{\mathtt{highest}} - \mathtt{cst})}{\chi + \psi}. \tag{3.1.1}$$

As mentioned before, it was shown using empirical data that retailers' behaviour at shopbots can be modelled with probabilistic pricing [6, 5]. In the model developed by Koçaş mixed strategies are described in detail. It is assumed that prices are generated as values of some random variable K. The strategy of a competing retailer in the model of Koçaş is represented with a cumulative distribution function:

$$F_K(p) = 1 - \left(\frac{(p_{\mathtt{highest}} - p)\,\chi}{(p - \mathtt{cst})\,\psi} \right)^{\frac{1}{e-1}}, \tag{3.1.2}$$

where p is a price a retailer sets; e is a number of retailers who participate in price competition, it is assumed that $e \geq 2$; $p_{\mathtt{highest}}$ is the highest reasonable price at the shopbot. The formula was obtained similarly as in the work of Koçaş, only the product cost $\mathtt{cst}$ was assumed to be non-zero. F_K is defined on the interval $[p_{\mathtt{min}}, p_{\mathtt{highest}}]$. $F_K(p_{\mathtt{highest}})$ is defined as a left-sided limit, $F_K(p_{\mathtt{highest}}) = \lim_{p \to (p_{\mathtt{highest}}-)} F_K(p) = 1$.

To gain some intuition about the pricing behaviour suggested by the strategy described above, some reasonable values for the constants in the Eq. (3.1.2) would be considered. It is assumed that $\chi = 15$ and $\psi = 100$. Let $p_{\mathtt{highest}} = \$1000$, $\mathtt{cst} = \$100$. Hence, $p_{\mathtt{min}} = 100 + 15\frac{1000-100}{15+100} \approx \$217$. Let the number of competing retailers be $e = 3$ or $e = 30$. The cumulative distribution functions (3.1.2) for given values are plotted in Fig. 3.1.1.

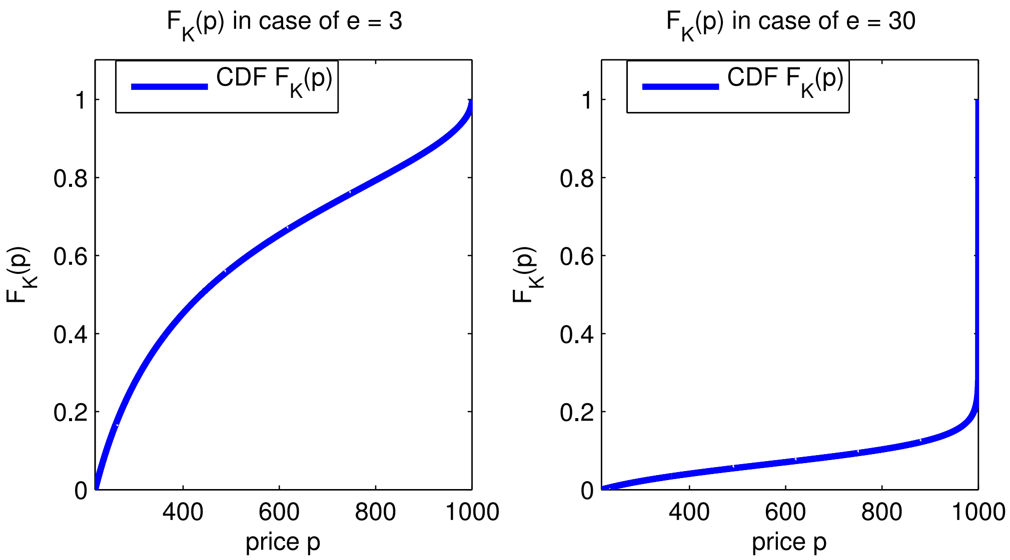

Fig. 3.1.1. The cumulative distribution function described with Eq. (3.1.2)

Using the cumulative distribution function (3.1.2), it can be shown that in the case of $e = 30$ competing retailers a probability of setting prices p such that $p \leq \$999$ is $P(p \leq \$999) \approx 25.9\%$. And in the case of $e = 3$ competing retailers $P(p \leq \$999) \approx 98.7\%$. Thus, the retailer has an incentive to sell to his loyals mostly if the number of competitors becomes large. However, if there are just several competitors, then a retailer actively participates in price competition.

To sum up, the model developed by Koçaş takes into consideration real-world differences in loyalty of customers, see Subsection 2.3.2. The model also deals with probabilistic pricing mentioned in 2.2.1. It explains the real-world price dispersion and price level, see

Subsection 2.2.1. However, neither the temporal structure of a retailer's behaviour, nor retailer's capacity constraints are considered.

It is possible to obtain pricing strategies based on the model. Due to the discussed limitations of the model, the strategies might not be efficient in cases of real-world problems.

3.1.5. Extensive-form game model developed by Martínez-de-Albéniz and Talluri [19]

A starting point of the work [19] is the Bertrand model of a duopoly with capacity constraints. It is called the Bertrand-Edgeworth model. If this model is extended to an extensive-form game, then there might exist a pure-strategy subgame perfect equilibrium with non-zero retailers' profits [25, 19].

Martínez-de-Albéniz and Talluri have considered in [19] an extensive-form game with perfect information between two retailers.

In the model each retailer has a fixed capacity, i.e. a given number of product pieces, he would like to sell over a finite time horizon. At most one customer could arrive during every stage of the game. Arrivals of customers are modelled as a stochastic process. All the results were obtained for a general stochastic process. Bernoulli stochastic process was pointed out as a possible option. In the model every stage of the game is called a *period of sales*. During every period before the end of the sales horizon each retailer sets a price. An expected number of future arrivals of customers is assumed to be known to both retailers. In the model customers are modelled as purely price-sensitive, so if a customer arrives he buys from a retailer who offers a lower price. If both retailers set the same price, then an arrived customer buys from any of them at random. It is assumed that every customer buys just one piece of product. It is also assumed that both retailers can observe inventory levels of each other. The last note about the model refers to a retailer's profit. In the model a product cost is assumed to be zero, so during one period of sales a retailer makes a profit equal to p if he sells one item at a price p. Non-zero product cost can be easily included in the model.

It was proved that in the model the unique subgame-perfect equilibrium in pure strategies exists. Technical proof can be found in [19]. Here an intuitive explanation will be provided.

The model assumes that customers always buy from a retailer with a lower price. Thus, the competition during a single period of sales is very similar to the Bertrand competition. However, it differs from the Bertrand model because each retailer can estimate an expected number of customers who will arrive until the end of the sales horizon, also inventory levels of both retailers are known to everyone. If none of the retailers can meet the total demand, then in the equilibrium a retailer with a larger inventory level would prefer to let another retailer sell everything out, rather than participate in a zero-profit Bertrand competition. After the smaller retailer sells everything out the bigger one becomes a monopolist and sets the highest reasonable price $p_{\texttt{highest}}$.

Equilibrium prices can be computed, see [19] for details. Before stating the results, it is necessary to introduce some notation.

Let T be the length of a sales horizon. A random variable which assigns a natural number of future arrivals of customers to every period t is denoted by $\mathbf{R}(t)$, $t \in \mathbb{N} \wedge t \leq T$. Let $P(\mathbf{R}(t) > a)$ be a probability that a value of random variable $\mathbf{R}(t)$ is greater than a, $a \in \mathbb{N}$. Let $\overline{\mathbf{R}}(t)$ be an expected value of the $\mathbf{R}(t)$.

It is assumed that one retailer has an inventory level x and another retailer's inventory level is y. In the equilibrium if a customer arrives, then he buys from a retailer with the lowest capacity at price p^* given by the expression

$$p^* = \begin{cases} (p_{\mathtt{highest}} - \mathtt{cst}) \cdot P(\mathbf{R}(t) \geq \min\{x, y\}), & x \neq y; \\ (p_{\mathtt{highest}} - \mathtt{cst}) \cdot P(\mathbf{R}(t) \geq x) \cdot \min\{1, \overline{\mathbf{R}}(t) - 2x + 2\}, & x = y. \end{cases} \tag{3.1.3}$$

Derivation of equilibrium price values can be found in [19]. Additionally, a non-zero product cost cst was included in the final expression (3.1.3).

The model predicts zero-profit and negative-profit prices in the considered duopoly. If each retailer can meet the whole demand on his own and retailers' inventory levels differ, then zero-profit prices are possible. In fact, the Bertrand competition takes place during each period of the sales horizon.

If both retailers have the same inventory level x and $2x > (\overline{\mathbf{R}}(t)+2)$, then negative-profit prices are predicted. Predicted for this case negative prices were explained. Each retailer has an incentive to pay for selling a few pieces of product because in this case he would become a retailer with a smaller capacity. So the competitor in the equilibrium would let him sell all remaining pieces of a product at some positive price.

The results have been generalised for a specific case of $n \geq 2$ retailers, $n \in \mathbb{N}$. The case when there are dominant retailers is considered. The dominant retailers have larger capacities than all non-dominant retailers together. It was shown, that if there is one dominant retailer, then in the equilibrium some smaller retailer sells a product at price $p = (p_{\mathtt{highest}} - \mathtt{cst}) \cdot P(\mathbf{R}(t) \geq \cdot x_1)$, where x_1 is a total capacity of all non-dominant retailers. If there are several dominant retailers with the same capacities, then negative-profit prices are possible in the equilibrium. It is the case if $x_\Sigma > (\overline{\mathbf{R}}(t) + 2)$, where x_Σ is a total capacity of all competing retailers.

To sum up, the model developed by Martínez-de-Albéniz and Talluri takes into consideration important real-world features of the competition at shopbots. Retailers' capacity constraints and real-world temporal structure of the competition has been modelled, including stochastic arrivals of customers. The main result of the work is computation of the unique subgame-perfect equilibrium for the case of duopoly. It is a stable solution to the competition. Existence of a Nash equilibrium was also shown for a specific case of oligopoly with dominant retailers. In that case in equilibrium all smaller retailers have to set the same price. However, it is known that prices of different retailers at a shopbot differ significantly [3]. Thus, the case of dominant retailers is not realistic.

The weak point of the model is the assumption of observability of retailers' inventory levels. In [19] it was discussed that for the case of duopoly inventory levels might be approximated if the total capacities are known, because in the equilibrium a retailer with smaller capacity would raise his price if a piece of a product is sold.

The model considers only the case of a duopoly. Extension to the case of more retailers is not general and cannot be used in real-world situations.

Another weak point of the model is a neglect of customers' brand-sensitivity. It was discussed that more complicated demand functions can be considered in order to model real-world brand-sensitive customers. However, in this case it is impossible to solve the problem analytically [19].

The analogous model with a more complex demand function exists. Its brief description will be given in the next section.

3.1.6. Extensive-form game model developed by Lin, and Sibdari [20]

The model developed by Lin, and Sibdari is essentially the same as the one developed by Martínez-de-Albéniz and Talluri. The differences in the model of Lin, Sibdari compared to the model of Martínez-de-Albéniz, Talluri will be listed.

a) There are in general n competing retailers in the model of Lin, Sibdari, $n \in \mathbb{N}$.

b) Multinomial logit (MNL) model is used to model choices of customers.

MNL model of the customer's behaviour considers a popularity of retailers. It is widely used in economic science literature [20]. If an arrived customer observes a price vector $\mathbf{p} = (p_1, p_2, ..., p_n)$, where p_i is a price of the i^{th} retailer, then the customer buys a product from the i^{th} retailer with a probability

$$P_i(\mathbf{p}) = \frac{e^{\alpha_i - \beta p_i}}{1 + \sum_{j=1}^{n} e^{\alpha_j - \beta p_j}} \tag{3.1.4}$$

and with a probability

$$P_0(\mathbf{p}) = \frac{1}{1 + \sum_{j=1}^{n} e^{\alpha_j - \beta p_j}} \tag{3.1.5}$$

a customer does not buy a product. β is called the price response coefficient and models price-sensitivity of customers. The parameter α_i models the popularity of the i^{th} retailer. Values of α_i and β has to be positive [21]. The lower the price p_i is, the higher the probability $P_i(\mathbf{p})$ is. The probability $P_i(\mathbf{p})$ is also higher if the popularity α_i is higher. It is always true that $\sum_0^n P_i = 1$.

In the model of Lin it is assumed that a retailer can set any positive price. If a retailer sets an infinite price, then he experiences a zero demand.

It has been proved that in the modelled competition a pure-strategy Nash equilibrium always exists. However, it might be not unique.

To sum up, the model developed by Lin and Sibdari considers brand-sensitivity of customers. However, the model becomes too complex and Nash equilibrium prices can be computed only numerically.

3.2. Related work based on optimal control

Methods of optimal control were used to analyse a dynamic pricing of perishable products by retailers with fixed capacities during a given time horizon [21].

A retailer tries to control a number of product pieces he sells using a price as a control variable. The number of his sales is influenced with other retailers' prices as well. Thus, an extension of a classical optimal control model should be used. Such an extension is a differential game. It is a type of optimal control problems with multiple and self-interested controllers [32].

In [21] price competition has been modelled as a differential game. A continuous time model was used. An inventory level was assumed to be a state variable in the model. It has been shown that under certain conditions a stable solution to the problem exists in the case of open-loop controls. Such a stable solution for a problem is Nash equilibrium for open-loop controls. Open-loop control is a pricing strategy in the case if retailers cannot observe inventory levels of their adversaries. Necessary conditions for existence of the equilibrium have been stated for different models of demand. For linear demand the equilibrium exists if a retailer can set such a price that he experiences a zero demand. For MNL demand the equilibrium was proved to exist always. Then in the work a problem-specific notion of Nash equilibrium was given, which deals with shadow prices. Shadow prices were defined as measures of the capacity externalities different retailers exert on each other. Conditions for the uniqueness of the defined Nash equilibrium were stated and can be found in the work [21]. It has been discussed that a problem of finding a solution to the stated differential game might be addressed as a finite-dimensional nonlinear numerical problem [21].

Note that it might be impossible to find an analytical solution to the differential game modelling the competition at shopbots due to the problem formulation. The issue should be discussed.

Let the simplest price competition between two retailers be considered. The first retailer tries to maximise his performance index $J_1 = \int_0^T L_1(x_1, x_2, u_1, u_2)dt$ over a time interval $[0, T]$, $x_1, x_2, u_1, u_2 \in \mathbb{R}$ for the sake of simplicity. The behaviour of the second retailer is analogous. The state equation is $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = f(x_1, x_2, u_1, u_2)$. In order to find the solution to the game, functions L_1, L_2 and f must be separable into $L_1 = L_{11}(x_1, x_2, u_1) + L_{12}(x_1, x_2, u_2)$, $L_2 = L_{21}(x_1, x_2, u_1) + L_{22}(x_1, x_2, u_2)$, $f = f_1(x_1, x_2, u_1) + f_2(x_1, x_2, u_2)$. In all related works available to the author analytical solutions to differential games similar to the one of interest are based on these assumptions [32], ([33], p. 278), ([34], p. 454).

However, even in the case of two retailers at a shopbot the assumptions do not hold. L_1 and f take the form $L_1 = u_1 f(u_1, u_2)$ and $f = f(u_1, u_2)$, where u_1 is a retailer's price and $f(u_1, u_2)$ is a demand function. If the demand function is separable, i.e. $f(u_1, u_2) = f_1(u_1) + f_2(u_2)$, then $L_1 = u_1 f_1(u_1) + u_1 f_2(u_2)$ is not separable due to a term $u_1 f_2(u_2)$. If the function L_1 is separable, i.e. $L_1 = L_{11}(u_1) + L_{12}(u_2) = u_1 f(u_1, u_2)$, then $f(u_1, u_2) = \frac{L_{11}(u_1)}{u_1} + \frac{L_{12}(u_2)}{u_1}$ is not separable due to a term $\frac{L_{12}(u_2)}{u_1}$. Indeed, the stated assumption of separability does not hold.

4. Pricing strategies based directly on models from related work

In the previous chapter possible models of price competition similar to the competition at shopbots were investigated. Based on these models pricing strategies for a retailer at a shopbot can be obtained.

4.1. Pricing strategy based on the Bertrand model

The Bertrand model of price competition is a single-stage game. Considering the repeating single-stage game a pricing strategy can be given as

$$p(t) = \begin{cases} p_{\min}(t-1) - 1, & p_{\min}(t-1) - 1 > p_{\texttt{zero profit}}; \\ p_{\texttt{zero profit}}, & otherwise, \end{cases}$$

where $p(t)$ is a price for a competition period t and $p_{\min}(t-1)$ is the minimal price at the shopbot during the previous period $(t-1)$. The strategy was given in [29]. For the sake of simplicity natural values of prices are considered. Natural-valued prices would suffice in the majority of real-world situations.

To make the strategy complete, the case of a monopoly has to be considered, i.e. the case when there are no adversarial retailers at a shopbot. In this case $p(t) = p_{\texttt{highest}}$.

For the purpose of numerical evaluations of the strategy's performance it is implemented in a Java class BertrandRetailer.java, see the attached CD.

4.2. Pricing strategy for a single-stage game based on the model developed by Koçaș [5]

The model of price competition at shopbots developed by Koçaş is described in 3.1.4. The same notation as in Subsection 3.1.4 will be used here.

In Subsection 3.1.4 pricing behaviour of a retailer in the equilibrium is given with a cumulative distribution function (3.1.2). Based on it the pricing strategy can be obtained.

First, it is needed to show that the cumulative distribution function (3.1.2) has an inverse function.

A derivative of the cumulative distribution function (3.1.2) with respect to price p can be computed:

$$\frac{dF_K}{dp} = \chi \cdot \frac{p_{\texttt{highest}} - \texttt{cst}}{\psi(e-1)(p-\texttt{cst})^2} \left(\frac{\chi\left(p_{\texttt{highest}} - p\right)}{\psi\left(p - \texttt{cst}\right)} \right)^{\frac{2-e}{e-1}}. \tag{4.2.1}$$

In the model it is assumed that a number of retailer's loyals is $\chi \geq 1$. Also it is assumed that a number of addressable switchers is $\psi > \chi$. As a consequence of the Eq. (3.1.1), $p_{\texttt{min}} > \texttt{cst}$, therefore the inequality $(p - \texttt{cst}) > 0$ holds for all prices. In Subsection 2.2.2 it was pointed out that $p_{\texttt{highest}} > \texttt{cst}$ for rational retailers. Rationality of a player is one of the main assumptions in game theory, therefore in game-theoretic model of Koçaş the inequality $p_{\texttt{highest}} > \texttt{cst}$ must hold. As a result, $(p_{\texttt{highest}} - \texttt{cst}) > 0$ for every retailer.

The inequality $p \leq p_{\texttt{highest}}$ holds for all prices, hence, $(p_{\texttt{highest}} - p) \geq 0$. In the model it is assumed that $e \geq 2$. Thus, $\frac{2-e}{e-1} \leq 0$. Hence, the factor $(p_{\texttt{highest}} - p)$ is in the dominator of the Eq. (4.2.1).

To conclude, all factors of $\frac{dF_K}{dp}$ in (4.2.1) are positive for $\forall p \in [p_{\texttt{min}}, p_{\texttt{highest}})$. For a boundary price $p = p_{\texttt{highest}}$ the left-hand derivative is

$$\begin{aligned} \frac{dF_K(p_{\texttt{highest}}-)}{dp} &= \lim_{h\to 0-} \frac{F_K(p_{\texttt{highest}}+h) - F_K(p_{\texttt{highest}})}{h} = \\ &= \lim_{h\to 0-} \frac{-\left((p-\texttt{cst})\,\psi\right)^{e-1}}{h\left((p_{\texttt{highest}} - p_{\texttt{highest}} - h)\,\chi\right)^{e-1}} = +\infty. \end{aligned} \tag{4.2.2}$$

F_K has a jump at $p = p_{\texttt{highest}}$.

As a result, the derivative $\frac{dF_K}{dp} > 0$ for all possible prices $\forall p \in [p_{\texttt{min}}, p_{\texttt{highest}}]$. Hence, F_K is increasing on the interval $[p_{\texttt{min}}, p_{\texttt{highest}}]$ ([35], p. 8). An increasing function has an inverse function ([35], p. 2). Function F_K maps the interval $[p_{\texttt{min}}, p_{\texttt{highest}}]$ to the interval $[0,1]$. Therefore, its inverse $F_K^{-1} : [0,1] \mapsto [p_{\texttt{min}}, p_{\texttt{highest}}]$. The inverse function is given as

$$F_K^{-1}(y) = \frac{p_{\texttt{highest}}\chi + \texttt{cst} \cdot (1-y)^{e-1}\psi}{\psi(1-y)^{e-1} + \chi}. \tag{4.2.3}$$

To obtain a pricing strategy for a retailer, The Inverse Transform Method can be used.

Theorem 4.2.1 (The Inverse Transform Method). *"Let a cumulative distribution function $F(x)$, $x \in \mathbb{R}$, have an inverse function $F^{-1}(y)$, $y \in [0,1]$. Let U be a random variable uniformly distributed on the interval $[0,1]$. Then F is a cumulative distribution function of $X = F^{-1}(U)$."*

The formulation was taken from ([36], p.6). The proof can be found in ([36], p.6).

Thus, in order to simulate the equilibrium pricing behaviour of a retailer in the model of Koçaş, it is needed to use a value y of a random variable U uniformly distributed over the interval $[0,1]$ and substitute y into Eq. (4.2.3). A uniformly distributed random variable can be modelled using a function of Java Math library `Math.random()`. The described technique provides a mixed strategy based on the model of Koçaş.

To use the strategy, knowledge about customer's loyals and addressable switchers is required. In the original model developed by Koçaş it is assumed that parameters χ and ψ are known to all competing retailers beforehand [5]. In all investigated related work it is assumed that characteristics of the demand, such as amounts of customers loyal to different retailers or parameters of the stochastic process of arrivals of customers, are known to all competing retailers before the start of price competition and estimation of the demand was never considered as a part of price competition [5, 19, 20]. The same assumption will be used in this work. Alternatively, it might be assumed that before the start of price competition all competing retailers have successfully estimated the characteristics of the demand. One possible method of the estimation can be found in Section 5.3. However, the meaning of the retailer's parameters χ and ψ in the case of repeating periods of the competition should be discussed.

The model developed by Koçaş is a single-stage game. The quality of the strategy based on the model will be evaluated using numerical simulation of the price competition at shopbots. In the simulation the single-stage game of Koçaş will be repeated. Arrivals of customers during different stages will be described using a stochastic process in accordance with the general model of the competition, see Subsection 2.3.4.

In the model developed by Koçaş it is assumed that during the single period of the competition there are χ customers loyal to our retailer and ψ switching customers. Both χ

and ψ are assumed to be known. As it was discussed in Subsection 2.3.2, χ is related to our retailer's popularity among customers. ψ might be related to characteristics of the product. If the product is just a book, then many customers might behave as switchers. But if the product is a laptop, then customers might get suspicious to buy the cheapest offer. To sum up, χ and ψ from the single-period model might be viewed as general characteristics of the structure of arrived customers during the multi-period competition. These parameters describe a relative share of customers loyal to different retailers and of switching customers. If there are m retailers at a shopbot and the popularity of the i^{th} retailer is characterized with χ_i, then it is assumed that an arrived customer is loyal to the i^{th} retailer with a probability

$$P_{i,\text{loyal}} = \frac{\chi_i}{\psi + \sum_{j=1}^{m} \chi_j}, \qquad (4.2.4)$$

where ψ characterises the number of switchers for the case of a particular product. An arrived customer is a switcher with a probability

$$P_{\text{switcher}} = \frac{\psi}{\psi + \sum_{j=1}^{m} \chi_j}. \qquad (4.2.5)$$

Let $\overline{\boldsymbol{Y}}$ be an expected number of customers arriving during a period of the competition. In this case an expected number of customers loyal to the i^{th} retailer is $\overline{\boldsymbol{Y}} \cdot P_{i,\text{loyal}}$ and an expected number of addressable switchers is $\overline{\boldsymbol{Y}} \cdot P_{\text{switcher}}$. The obtained expected numbers can be used in the simulation of pricing behaviour based on the model of Koçaş.

It completes the description of the mixed pricing strategy developed using the model of Koçaş.

For the purpose of numerical evaluations of strategies' performance the strategy is implemented in a Java class KocasRetailer.java, see the attached CD.

4.3. Pricing strategy for a multiple-stage game based on the model developed by Martínez-de-Albéniz and Talluri [19]

Another pricing strategy for a retailer at a shopbot can be obtained based on the model of Martínez-de-Albéniz and Talluri, see Subsection 3.1.5. In the case of duopoly an equilibrium pricing strategy was computed [19]. A non-zero product cost has been already included in the result expression (3.1.3) in previous chapter. It is possible to use the expression directly. One additional modification will be done.

In the model negative equilibrium prices are predicted. In real-world situations when a retailer tries to sell product pieces quickly he might discount and set prices lower than the product cost. However, negative prices are not common. It is not profitable for a retailer to pay not only the product cost but also additional amount of money. In order to make pricing strategy more realistic, it will be assumed that a minimal possible price is equal to zero.

The pricing strategy can be summarized as follows:

A retailer compares his own capacity and the competitor's capacity. If his capacity is greater, then he sets a price $p_{\texttt{highest}}$. Otherwise, a price is computed using the expression (3.1.3). If according to the Eq. (3.1.3) the equilibrium price would be negative, a retailer would set a price equal to zero.

To use the Eq. (3.1.3), it is required to know parameters of a stochastic process which describes arrivals of customers. In the work [19] the parameters are assumed to be known to both customers. As it was discussed in the previous section, the same assumption is used in this work.

In the work [19] the Bernoulli stochastic process was suggested to model arrivals of customers. The same stochastic process was chosen in the general model, see Subsection 2.3.4. In case of the Bernoulli process the binomial random variable describes a number of arrived customers during n independent periods of arrivals ([26], p. 298). Let n be a maximal possible number of customers during the competition. Periods of the competition in the model of Martínez-de-Albéniz and Talluri are defined with periods of possible arrivals of customers. As a consequence, the length of a sales horizon is n [19]. Thus, the model assumes that the maximal possible number of remaining customers until the end of the competition is $(n - t)$, where t is a current period. Let ξ be a probability of arrival of a customer during a single period of arrivals. As a result, an expected value $\overline{\mathbf{R}}(t) = \xi(n - t)$, see Subsection 3.1.5 for details on notation.

During l periods of the competition k customers arrive with a probability $P'(k) = \binom{l}{k}\xi^k(1-\xi)^{(l-k)}, k \leq l$.

Then $P(\mathbf{R}(t) > a)$ from the Eq. (3.1.3) can be computed as follows:

$$P(\mathbf{R}(t) > a) = \sum_{j=a+1}^{n-t} \left(\binom{n-t}{j} \xi^j (1-\xi)^{(n-t-j)} \right). \qquad (4.3.1)$$

It completes the description of the pricing strategy based on the model of Martínez-de-Albéniz and Talluri.

For the purpose of numerical evaluations of different strategies and their performance the developed strategy is implemented in a Java class MartinezTalluriRetailer.java, see the attached CD.

Note that the strategy was obtained for the case of duopoly. A heuristic extension of the strategy might be suggested for the case of several competing retailers. All competitors can be modelled as a single adversary who has a capacity equal to a total capacity of all adversaries.

Note on the model developed by Lin and Sibdari [20]

Lin and Sibdari developed another extensive-game model of price competition. The model might be viewed as an alternative to the model of Martínez-de-Albéniz and Talluri [19].

In the model of Lin and Sibdari it is impossible to obtain an analytical equilibrium solution to the formulated problem. Equilibrium pricing strategies were proved to exist and it might be possible to compute the strategies numerically for the given model of demand [20]. Details about the computation can be found in [20].

The MNL demand model is used. It enables to consider brand sensitivity of customers. On the other hand, it does not explicitly consider a real-world level of prices for a particular product. If a customer arrives and he has to choose among several retailers with different popularities, then in the model there is a non-zero probability that a customer would buy the product at arbitrary high prices. In order to use the model in simulations of the competition, it is not enough to know characteristics of the demand, such as amounts of customers loyal to different retailers and parameters of arrivals of customers. For every combination of competing retailers parameters of the MNL demand model have to be fitted in such a way that a level of equilibrium prices would make sense for the considered product.

The model of Martínez-de-Albéniz and Talluri explicitly takes into consideration a real-world price level for the case of a particular product. For the purpose of evaluation of a strategy based on an extensive-form game only the strategy obtained from the model of Martínez-de-Albéniz and Talluri will be investigated.

5. Multi-period model as an extension of the model developed by Koçaș [5]

Using empirical data it was shown that pricing behaviour of retailers at a shopbot can be described by probabilistic pricing, i.e. by using mixed pricing strategies [5, 6]. Mixed pricing strategies might arise due to the presence of loyal customers [24, 23]. In Section 4.2 the mixed pricing strategy based on these findings was developed. Unfortunately, the strategy takes into consideration neither temporal structure of the competition, nor the retailer's inventory level. However, in order to maximise the total profit from a fixed capacity over a given sales horizon, a retailer should take the temporal structure of the competition and capacity constraints into consideration.

The model of Koçaş can be improved by considering a multi-period competition in accordance with the general model from Chapter 2. In the case of a multi-period competition a retailer can observe pricing behaviour of all other retailers and take the observed information into consideration before setting a price for the next period.

It is required to develop a pricing strategy for one particular retailer. This retailer is named an *our retailer*. All other retailers at a shopbot can be viewed as adversaries of our retailer, they are named *adversarial retailers* or simply *adversaries*.

Based on an assumption that retailers adopt mixed-strategies it is possible to estimate a probability that our retailer has the lowest available price at a shopbot if he sets some arbitrary price. Based on this finding a demand function for our retailer can be approximated.

Demand is determined by customers' preferences. Loyal customers prefer to buy from a particular retailer at any price $u \in [p_{\texttt{zero profit}}, p_{\texttt{highest}}]$[1].

Due to the presence of switching customers a retailer experiences a jump in the demand when he starts to offer a product at a price lower than prices of adversarial retailers [24]. Thus, it is important to estimate our retailer's probability $P(u)$ of having the lowest price at a shopbot when he sets some reasonable price u, $u \in [p_{\texttt{zero profit}}, p_{\texttt{highest}}]$.

5.1. Probability of having the lowest price

If our retailer sets a price lower than all adversaries do, he will attract not only his loyals but also all switching customers at a shopbot. In order to understand our retailer's probability $P(u)$ of having the lowest price, it is important to understand pricing behaviour of adversarial retailers.

Adversarial retailers at a shopbot are numbered. Let there be q adversarial retailers at a shopbot, $q \in \mathbb{N}$. Each adversary at a shopbot has assigned an ordinal number i, $i \in \mathbb{N} \wedge i \leq q$.

Pricing behaviour of retailers at shopbots can be described using probabilistic pricing [6, 5]. A price which a particular adversarial retailer i sets can be viewed as a value of some discrete random variable $\mathbf{Var}_i$[2]. The random variable $\mathbf{Var}_i$ assigns to every possible period

[1] See Subsection 2.2.2 for details on possible prices.

[2] The prices a retailer can set in the real life belongs to some discrete set. So a discrete random variable is considered.

of the competition a reasonable price[3]. The probability mass function of the considered random variable $\mathbf{Var}_i$ depends on retailer-specific parameters [6, 5]. These parameters might be sizes of loyal and switching segments, see Subsection 2.3.2. The size of a retailer's loyal segment is related to retailer' popularity. The size of the switching segment might be determined by characteristics of the considered product. Neither retailer's popularity, nor characteristics of the product change too quickly. Thus, it is assumed that a size of the switching segment as well as sizes of loyal segments of all retailers at a shopbot are constant during the considered price competition [5]. As a consequence, it is assumed that every adversarial retailer sets prices according to some static probability mass function as long as he has some pieces of a product to sell. However, very often information on retailer's capacities is not public. The worst possible case when all adversaries have infinite capacities is assumed. Thus, all adversaries set prices according to their static probability mass function and no adversary ever quits the competition.

To estimate the mentioned probability mass functions, available observations of previous prices can be used. The set of all reasonable prices is $[p_{\texttt{zero profit}}, p_{\texttt{highest}}]$ for every competing retailer, see Subsection 2.2.2 for details. Let this set be called a *set of allowed prices* $= S_{allowed}$. Let $\texttt{k}$ be a number of values in $S_{allowed}$, $\texttt{k} \in \mathbb{N}$. Ordinal numbers can be assigned to elements of $S_{allowed}$ starting from the lowest price up to the highest one. Thus, the lowest price in $S_{allowed}$ can be denoted by $u_1 = p_{\texttt{zero profit}}$, the highest price is in this case $u_{\texttt{k}} = p_{\texttt{highest}}$ and $S_{allowed} = \{u_1, u_2, ..., u_{\texttt{k}}\}$. During some number of competition periods prices of each adversarial retailer are observed. For each adversary our retailer counts how many times every allowed price has occurred and divides the result by a total number of observations for the retailer. As a result, an estimation of the adversarial retailer's probability mass function is obtained.

Knowing estimations of all adversaries' probability mass functions it is possible to estimate probabilities $P_1, P_2, ..., P_{\texttt{k}}$ that a customer would buy a piece of a product from our retailer at prices $u_1, u_2, ..., u_{\texttt{k}}$ respectively. P_1 is a probability that an arrived customer would buy a piece of a product from our retailer at a price u_1. For the sake of compactness a situation when a customer buys a piece of a product from our retailer is named *a success*. Next, a mathematical formula for P_i, $i \in \mathbb{N}$, $i \leq k$ is derived.

The probability of success for price u_1, P_1, is equal to a probability that all adversaries set prices higher than u_1 plus a probability of success in case of ties, i.e. in the case when our retailer sets the lowest available price u_1 but he is not the only one who sets this price. A similar general note on probabilities of success was made in [5] without mentioning the ties.

Probability mass functions of random variables $\mathbf{Var}_i$ are assumed to be static and each adversary retailer uses only his own probability mass function while setting prices. Thus, for $\forall i, j : \; i, j \in \mathbb{N} \wedge i \leq q \wedge j \leq q$ $\mathbf{Var}_i$ and $\mathbf{Var}_j$ are independent random variables. Therefore, a probability that at the same time several adversarial retailers set prices higher than some allowed price u_f is equal to a product of probabilities that each adversary sets a price higher than u_f. Let $P_{j,l}$ be a probability that a retailer with an ordinal number j sets an allowed price u_l. In the case when our retailer sets an allowed price u_n a probability that all adversaries set higher prices is

$$P_{n\ \texttt{the only lowest}} = \prod_{j=1}^{q} \left(\sum_{l=n+1}^{k} P_{j,l} \right).$$

Note that $\forall j \in R : \sum_{l=k+1}^{k} P_{j,l} = 0$.

[3]Note that by possible periods of the competition all periods of the same length which can take place are meant. All these possible periods make up a sample space the random variable $\mathbf{Var}_i$ is defined on.

Possible ties should be taken into consideration. Let ties be broken at random. It means that an arrived customer would at random choose a retailer from those who offer the lowest price. During every period of the competition it is possible to have the same lowest price with just one adversary, or with several adversaries, or with all of them. In the case when our retailer has the same price as m adversaries an arrived customer would buy from our retailer with probability equal to $\frac{1}{m+1}$, as ties are broken at random. During some period of the competition it is possible that just one adversary has the same price as our retailer does, this one adversary might be any adversary at the shopbot. Also during a period of the competition any pair of adversaries can have the same price as our retailer does, and so on. In total, during some period of the competition our retailer might have the same price as adversaries in any subset of a set of all adversarial retailers at a shopbot. If our retailer has the same price as an empty subset of adversaries, then he is the only retailer who has a given price at a shopbot. Some notation should be developed. Let R be a set of all adversaries' ordinal numbers. Let $\mathcal{P}(R)$ be a power set of the set R.

The expression for P_n is derived in several steps. Let $P_{\texttt{helping}}$ be a probability of the case when our retailer sets an allowed price u_n, the price u_n is the lowest price at a shopbot, and each and only each adversarial retailer in a set O also sets the price u_n, $O \in \mathcal{P}(R)$. In this case all adversaries in a set $R \setminus O$ must have prices higher than u_n. Probability of such situation is

$$P_{\texttt{helping}} = \prod_{j \in \{R \setminus O\}} \left(\sum_{l=n+1}^{k} P_{j,l} \right).$$

Probability that when our retailer sets a price u_n there is a tie among and only among retailers in the subset O and our retailer is $P_{\texttt{tie}} = \prod_{i \in O} P_{i,n}$. Consequently, the probability of success in the case when our retailer sets a price u_n and there is a tie among and only among retailers in the subset O and our retailer is

$$P_{\texttt{success helping}} = \frac{1}{\mid O \mid +1} \left(\prod_{i \in O} P_{i,n} \right) \left(\prod_{j \in \{R \setminus O\}} \left(\sum_{l=n+1}^{k} P_{j,l} \right) \right),$$

where $\mid O \mid$ is a cardinality of the subset O. Note that we have chosen one particular element O of the power set $\mathcal{P}(R)$. In order to obtain the final formula for probability of success for a particular allowed price u_n, it is necessary to sum probabilities $P_{\texttt{success helping}}$ for all possible subsets $O \in \mathcal{P}(R)$:

$$P_n = \sum_{O \in \mathcal{P}(R)} \left[\frac{1}{\mid O \mid +1} \left(\prod_{i \in O} P_{i,n} \right) \left(\prod_{j \in \{R \setminus O\}} \left(\sum_{l=n+1}^{k} P_{j,l} \right) \right) \right]. \tag{5.1.1}$$

5.2. Estimated demand function

The method for computing probabilities of having the lowest price was suggested in the previous section. It can be combined with information on arrivals of customers and the retailer's popularity, which are assumed to be known before the start of the competition. Based on the knowledge about customers who are loyal to different retailers and about price-sensitive customers it is possible to compute a probability $P_{\texttt{our},\text{loyal}}$ that an arrived customer is loyal to our retailer, see the Eq. (4.2.4). Using the Eq. (4.2.5) a probability P_{switcher} that an arrived customer is a switcher can be estimated. Using an expected number $\overline{\mathbf{Y}}$ of customers during one period of the competition and the probabilities $P_1, P_2, ..., P_{\text{k}}$ from the previous section it is possible to obtain an estimation of the demand function. Let $D_1, D_2, ..., D_{\text{k}}$ be estimations of expected numbers of customers who would like to buy a piece of a product from our retailer at allowed prices $u_1, u_2, ..., u_{\text{k}}$ respectively.

Our retailer's loyal customers buy from him as long as he sets a price $p \leq p_{\mathtt{highest}}$. An expected number of arrived loyal customers is $\overline{\mathbf{Y}}_{\text{our}} \cdot P_{\mathtt{our},\text{loyal}}$. An expected number of arrived switching customers is $\overline{\mathbf{Y}}_{\text{our}} \cdot P_{\text{switcher}}$, but they will buy from our retailer only if he has the lowest price. Here the probabilities of having the lowest price at a shopbot are used.

As a result, for every allowed price u_i we obtain D_i as follows:

$$\forall i \in [1, \mathtt{k}] \subset \mathbb{N} : \; D_i = \overline{\mathbf{Y}} \left(P_{\mathtt{our},\text{loyal}} + P_i \cdot P_{\text{switcher}} \right). \tag{5.2.1}$$

Let $\mathbf{D}$ be a vector $(D_1, D_2, ..., D_{\mathtt{k}})$. It will be called a demand vector $\mathbf{D} = (D_1, D_2, ..., D_{\mathtt{k}})$.

5.3. Estimation of the demand structure

Analysing price competition, characteristics of the demand are assumed to be known to all retailers before the start of the competition and estimation of the demand structure is not considered to be a part of a pricing strategy, as it was done in related work [5, 19, 20]. However, one possible method for estimation of the demand structure is described in this section for the sake of completeness.

In the model of demand it is assumed that each arrived customer buys just one piece of a product [20, 19]. A customer who buys more pieces of a product is modelled as several customers. Numbers of arrived customers are modelled with the Bernoulli stochastic process.

Let $\boldsymbol{Y}$ be a random variable assigning a total number of arrived customers to every period of the competition, see Subsection 2.3.4. The Bernoulli stochastic process was chosen to model arrival of customers. Hence, $\boldsymbol{Y}$ is the binomial random variable ([26], p. 298). Let n be a maximal number of arrivals of customers during a period of the competition. The arrivals during one period of the competition can be visualized as a sequence of n coin tosses ([26], p. 297). Every two of these coin tosses are independent. If the outcome of a toss is head, then a customer arrives. And a probability of a head is ξ.

If during some period of the competition our retailer sets the lowest price at a shopbot, then both switchers and our retailer's loyals buy pieces of a product from our retailer. Customers loyal to some adversarial retailers would arrive to the shopbot also, but they never buy from our retailer. It was discussed that numbers of different retailers' loyals χ and a number of switchers ψ are viewed as characteristics of the total demand structure and describe relative shares of customers loyal to different retailer and switching customers, see Section 4.2. Let n_{our} be a maximal number of arrivals of customers who can potentially buy from our retailer. If there are m retailers at a shopbot and the popularity of the i^{th} retailer is characterized with χ_i, then

$$n_{\text{our}} = \frac{\chi_{\text{our}} + \psi}{\psi + \sum_{j=1}^{m} \chi_j} n, \tag{5.3.1}$$

where ψ characterises the number of switchers for the case of a particular product. As a consequence, our retailer can observe at maximum only n_{our} possible arrivals from a total maximal number of n. Still every possible arrival of interest occurs with the probability ξ. Let $\mathbf{Y}_{\text{our}}$ be a random variable determining a number of arrived customer who can potentially buy from our retailer, i.e. a number of our retailer's loyals and all switchers, which is at maximum n_{our}. Obviously, $\mathbf{Y}_{\text{our}}$ is also a binomial random variable. Parameters of this variable can be estimated.

The maximal possible number n_{our} of potential customers is equal to a maximal possible amount of arrived switchers together with the retailer's loyals. For the sake of simplicity the estimations of n_{our} and ξ are denoted by n_{our} and ξ as well. In order to estimate parameters of $\mathbf{Y}_{\text{our}}$ correctly, our retailer has to set the lowest price at a shopbot and attract switchers. He can observe numbers of arrived customers during different periods of the competition. Using these observed numbers the retailer can compute a sample mean $\overline{\mathbf{Y}}_{\text{our}}$. It estimates an expected value of a number of his potential customers arriving a period. Based on observations it is possible to compute the sample variance $S^2_{\mathbf{Y}_{\text{our}}}$. Using formulas of expected value and variance for the Bernoulli process, for instance from ([26], p. 298), it is possible to estimate n_{our} and ξ as follows:

$$\xi = 1 - \frac{S^2_{\mathbf{Y}_{\text{our}}}}{\overline{\mathbf{Y}}_{\text{our}}};$$

$$n_{\text{our}} = \frac{\xi}{\overline{\mathbf{Y}}_{\text{our}}}\ .$$

If our retailer sets a maximal price for some number of competition periods, then during each period only his loyal customers would buy from the retailer. The retailer can compute an average number $\overline{\chi}_{helping}$ of arrived loyals during one period, which is at the same time equal to

$$\overline{\chi}_{helping} = \overline{\mathbf{Y}}_{\text{our}} \cdot \frac{\chi_{\text{our}}}{\chi_{\text{our}} + \psi}.$$

$n_{\text{our}} = \chi_{\text{our}} + \psi$. Thus, it is possible to estimate

$$\chi_{\text{our}} = \frac{n_{\text{our}} \cdot \overline{\chi}_{helping}}{\overline{\mathbf{Y}}_{\text{our}}},$$

$$\psi = n_{\text{our}} - \chi_{\text{our}}.$$

To sum up, it was described how to estimate parameters n_{our}, ξ, χ_{our}, ψ of customers' arrivals modelled as the Bernoulli process. The meaning of the parameters can be summarized as follows: at maximum n_{our} customers can arrive during a period of competition and buy from our retailer. Customers arrives sequentially and one customer arrives with a probability ξ. If one of n_{our} customers arrives, then he will be loyal to our retailer with a probability of $\frac{\chi_{\text{our}}}{\chi_{\text{our}}+\psi}$ and a switching customer with a probability of $\frac{\psi}{\chi_{\text{our}}+\psi}$.

Therefore, the method estimates not only parameters of arrivals of customers, but also an expected number of customers who are loyal to our retailer, compared to an expected number of addressable switching customers.

The compared popularity of different retailers is expressed via shopbot's rating systems and is public information. Using it and the expected number χ_{our} of our retailer's loyals it is possible to estimate an expected number χ_i of customers who are loyal to the i^{th} adversarial retailer. If required, based on the Eq. (5.3.1) it is also possible to estimate a maximal total number n of all customers arriving during a period of the competition using estimations of ξ, n_{our}, ψ and χ_i for different retailers.

Note, that if some pricing strategy does not consider loyalty of customers and models all customers as switchers, then only n_{our} and ξ are required. n_{our} is assumed to be the total maximal number of arriving customers in this case.

To sum up, even though estimation of demand is not considered to be a part of price competition, it was shown how the demand can be estimated by a retailer. The estimation can be used in cases when there is no price competition as well. In this case in order to estimate a total number of potential customers, a retailer does not have to set the lowest possible price as at any reasonable price every potential customer would buy from our retailer.

6. Design of pricing strategies using the formulated multi-period model

Based on the developed multi-period model from Chapter 5 it is possible to obtain new pricing strategies for retailers at shopbots. The same notation as in the Chapter 5 will be used.

6.1. Strategy 1: fictitious play approach

A model of multi-period competition was developed. It allows our retailer to set a price for the next period based on experience gained so far. In other words the multi-period model enables our retailer *to learn*. One of basic game-theoretic learning rules is called a *fictitious play* ([27], p. 201). Fictitious play is such a learning rule, when the retailer assumes that his adversaries adopt mixed strategies. The retailer estimates these strategies based on previous actions of his adversaries. During every stage of the game the retailer plays a best response to the estimated mixed strategies of his adversaries ([27], p. 206). Thus, using fictitious play it is possible to develop a strategy computing a price for a single stage game. The strategy takes into consideration only previous observations of adversaries' prices and possible loyalty of customers. It might be viewed as an extension of the strategy based on the model of Koçaş [5].

Fictitious play suggests setting a price which is the best response to estimated strategies of other retailers. Estimations of the strategies of all adversaries were successfully included into the demand function, formulated in the Section 5.2. To every allowed price u_i a rational number D_i was assigned. D_i can be viewed as an expected number of items which would be sold on average at price u_i during one period of the competition, if the average is computed using a large number of competition periods.

A single-period best response is a price maximising an expected profit over one period. It can be computed as

$$u_{\texttt{optimal}} = \underset{u_i,\ 1 \leq i \leq k}{\arg\max} \left(D_i \left(u_i - \texttt{cst} \right) \right). \tag{6.1.1}$$

In accordance with fictitious play a retailer should guess strategies of his adversaries before the first period of the competition. After each observed action the retailer updates his believes about the competitors' strategies. As it can be seen on the example of strategies developed in Chapter 4, strategies of competitors might differ significantly. Initial guess might be misleading. To avoid it, a retailer adopting fictitious play strategy sets the highest price during the first period of the competition without any initial guess, but starting from the second period of the competition the retailer always plays a best response to approximated strategies of his adversaries based on previous observations.

Now Eq. (6.1.1) together with the demand function, formulated in Section 5.2, describe the pricing strategy based on fictitious play.

For the purpose of numerical evaluations of the strategy's performance it is implemented in a Java class Fictitious.java, see the attached CD.

If all adversaries have fixed mixed-strategies, then the pricing strategy (6.1.1) will converge to the best response for a single-period game as a number of observed periods of competition goes to infinity ([27], p. 206). However, the strategy considers neither the real-world temporal structure of the competition, nor capacity constraints.

6.2. Optimal control problem

One possible way to develop strategies, which would use fictitious play learning rule and at the same time consider both the retailer's inventory level and the temporal structure of the competition, is to use methods of optimal control. A rational retailer at a shopbot tries to control a number of sold product pieces during a given time horizon in a way which maximises his profit.

Customers, our retailer and adversarial retailers can be viewed as parts of a complex system. The system will be called a *shopbot system*. To develop a model of the shopbot system, it is important to describe the key characteristics of the system.

6.2.1. General description of the system and the problem

Our retailer sells a product. The product price is a primary determinant for customers' choices [7]. The lower price our retailer sets, the more customers buy from him and the faster our retailer's inventory level decreases. An inventory level, or in other words the number of product pieces our retailer currently has, describes a current state of the shopbot system. The inventory level can be viewed as a state variable of the system. The price can be viewed as a control variable. Demand describes how many pieces would be bought at a particular price, or in our retailer's case it describes how a particular control would change a current value of the state variable. The change of the state variable does not depend on its current value. Thus, the demand function fully describes dynamic behaviour of the system.

Even though a number of sold pieces does not depend on an inventory level, the inventory level limits a maximum number of pieces that can be sold. It is impossible to have a negative number of remaining pieces. Thus, our retailer experiences a constraint on the state variable. Price takes values from a finite set, thus the retailer experiences constraints on the control variable, see Subsection 2.2.2.

The aim of our retailer is to sell pieces of a product in a way which maximizes his profit over a sales horizon. The greater the profit is, the more satisfied the retailer will be. Therefore, the retailer's profit can be viewed as a performance index, the term *performance index* is taken from ([34], pp. 20, 112).

The problem our retailer faces is to find an optimal control for the entire sales horizon. The optimal control would be a function which maximises the total profit of our retailer, assigning to each competition period during the sales horizon a value of the control variable.

6.3. Strategy 2: fictitious play in combination with variational approach to optimal control

Relevant characteristics and components of the shopbot system have been underlined. The most important component of the system is the demand function. It determines a change of the state variable.

Possible state representation

Fictitious play suggests estimating behaviour of adversaries using previous observations of their actions ([27], p. 206). In Section 5.2 it was shown how estimations of the adversaries' strategies can be expressed using the demand vector $\mathbf{D} = (D_1, D_2, ..., D_\mathtt{k})$. To every allowed price u_i a rational number D_i was assigned. D_i can be viewed as an expected value of a number of product pieces which would be sold at price u_i during one period of the competition.

In order to capture the real-world temporal structure of the competition, a sequence of the competition periods was modelled. For this purpose it is needed to represent new expected inventory levels after every period of the competition. It is assumed that there is some natural number $x_\mathtt{initial}$ of items at the beginning of the first competition period. If an allowed price u_i is set during the first period, then after it an inventory level $x_1 = x_\mathtt{initial} - D_i$ will be obtained. Taking into account the meaning of D_i, x_1 has the meaning of an expected value of number of items which would remain after the first competition period if the price u_i is set during this period. Note, that the obtained demand function assigns to every possible price an empirical estimation of a number of bought items. In the state equation of the shopbot system $x[k+1] = f(x[k], u[k])$ function $f(x[k], u[k])$ is not expressed analytically, $k \in \mathbb{N} \wedge k \leq T$, T is a length of sales horizon. The stated problem has to be solved numerically in this case. Therefore, it is important to estimate a numerical complexity of simulations based on the obtained demand.

Numerical complexity of simulations based on suggested state representation

In attempts to work with the suggested state representation insurmountable numerical difficulties occurred. Even for simple problems with small initial capacities and short sales horizons the simulation of the system required too much time and memory. The time complexity and the memory complexity were considered in more detail in Appendix B.

The generation of states in a numerical simulation in case of a small initial inventory level $x_\mathtt{initial} = 2$ and short sales horizon ten periods long was examined experimentally. Even for such a simple case numerical complexity of the simulation was high. However, if the initial inventory level was increased to $x_\mathtt{initial} = 3$, then the numerical complexity of simulations using the suggested state representation became insurmountable.

6.3.1. Continuous demand approximation

It is difficult to work with the estimated demand $\mathbf{D}$. Ways how to make a problem formulation easier must be found. One possible approach is to use a continuous model of the demand. Price competition has been modelled as a continuous system in work [21]. Continuous control problems have simpler mathematical formulation than their discrete equivalents ([34], p.110). Furthermore, if an analytical solution to continuous formulation of the stated optimal control problem can be obtained, then computation of pricing strategy based on the solution would have extremely low computational complexity.

A continuous linear demand function is very common in applied economic analysis due to its simplicity [37, 21]. The demand, which our retailer experiences, will be modelled using a linear demand function. For every possible price u from a set $u \in [p_\mathtt{zero\ profit}, p_\mathtt{highest}]$ a linear demand function will return a number of customers who would buy a piece of product at that price from our retailer. To estimate parameters of this linear function, the estimated demand vector $\mathbf{D} = D_1, D_2, ..., D_\mathtt{k}$ will be used, see Section 5.2 for details on the demand vector. Values $D_1, D_2, ..., D_\mathtt{k}$ correspond to prices $u_1, u_2, ..., u_\mathtt{k}$. Parameters of the demand function can be estimated using the method of least squares. Let he linear demand function be $D(u) = \mathbf{a} \cdot u + \mathbf{b}$, where $\mathbf{a}$ and $\mathbf{b}$ are constants. The linear function $D(u)$ might assign negative value to large values of allowed prices. Let a value of the obtained linear function $D(u)$ be negative for all prices $u \geq p_\mathtt{zero\ demand}$, i.e. $p_\mathtt{zero\ demand} = -\frac{\mathbf{b}}{\mathbf{a}}$.To avoid a negative demand, constraints on control in our model must be updated.

Let $u_{\max} = \max\{p_{\texttt{highest}}, p_{\texttt{zero demand}}\}$. Now a set of all allowed prices can be given as $u \in [p_{\texttt{zero profit}}, u_{\max}]$.

Note that in the general model of the competition a retailer automatically quits a shopbot when he has nothing to sell. It is enabled by a shopbot in real-world situations [14].

In accordance with fictitious play, adversaries' strategies have to be guessed before the start of competition ([27], p. 206). It can be slightly modified. To avoid misleading initial guesses, the retailer would set the price $p_{\texttt{highest}}$ during the first period and starting from the second period of the competition would estimate the demand function based purely on previous observations. Note that the obtained function $D(u)$ will be always decreasing in price. For instance, if all retailers always set the highest price $p_{\texttt{highest}}$, then a probability that our retailer has the lowest price at a shopbot is equal to one for all prices except for $p_{\texttt{highest}}$. As a consequence, $D(u)$ is decreasing. If in another extreme case all retailers always set $p_{\texttt{zero profit}}$, then probabilities that our retailer's price is lower than any adversary's price are equal to zero for all prices except for the price $p_{\texttt{zero profit}}$, as ties are possible. Again, as a result, estimated $D(u)$ is a decreasing linear function. Hence, $\mathbf{a}$ is always negative. An expected number of arrived customers should be positive, otherwise it is pointless to participate in price competition. Hence, $\mathbf{b}$ is always positive.

Note that the obtained demand function is an approximate estimation of demand. It is assumed that an expected number of retailer's potential customers is always the same. However, if a current inventory level x is less than a maximal number, n, of customers who might buy a piece of a product from our retailer, then the expected number of potential customers would be

$$Y_x = \sum_{i=0}^{x} Prob_i \cdot i + \sum_{i=x+1}^{n} Prob_i \cdot x,$$

where

$$Prob_i = \binom{n}{i} \cdot p^i \cdot (1-p)^{n-i}$$

is a probability that i potential customers arrive, given by the chosen Bernoulli process for modelling arrivals of customers.

6.3.2. Model of the competition and problem formulations

In order to work with the continuous demand, a continuous model of the system has to be developed.

It is assumed that a sales horizon starts at time $t = 0$ and ends at time $t = T_{\texttt{final}}$, $T_{\texttt{final}} \in \mathbb{R},\ T > 0$.

Let $x(t)$ be our retailer's inventory level, i.e. a quantity of remaining pieces of the product at time t; $\forall t \in [0, T_{\texttt{final}}] \subset \mathbb{R} :\ x(t) \in \mathbb{R},\ x(t) \geq 0$. In this case $x(t)$ is the state variable. Inventory level is modelled as a continuous quantity, as it was done in work [21].

The capacity of our retailer is denoted by c, i.e. $x(0) = c$, $c \in \mathbb{R},\ c > 0$. Knowing the capacity and the state equation it is possible to express an inventory level at time t, see an equation for the state in the problem formulation below.

A price function is denoted by $u(\cdot) : [0, T_{\texttt{final}}] \mapsto [p_{\texttt{zero profit}}, u_{\max}]$. It assigns to each time $t \in [0, T_{\texttt{final}}] \subset \mathbb{R} \wedge x(t) > 0$ an allowed real-valued price from the interval $[p_{\texttt{zero profit}}, u_{\max}]$. If the retailer's inventory $x(t) = 0$, then he automatically quits a shopbot, i.e. the retailer's offer becomes unavailable to customers and no one can buy from him. Note, $u(\cdot)$ is a control.

The demand is denoted by $D(u(t)) \in \mathbb{R}$. $D(u(t)) = \mathbf{a} \cdot u(t) + \mathbf{b}$. $D(u(t))$ is a number of customers who would buy a piece of a product at price $u(t)$ from our retailer at time t, each customer buys just one piece.

Let $h(n) : [0, c] \mapsto [0; \mathtt{h} \cdot c]$ be a retailer's real-valued holding cost function, the term is taken from [21]. $\mathtt{h}$ is a specific constant for our retailer, $\mathtt{h} \in \mathbb{R}, \mathtt{h} \geq 0$. $h(n)$ is a cost of holding a product quantity n, $n \in \mathbb{R},\ n \geq 0$, over one sales horizon. $h(n) = \mathtt{h} \cdot n$.It is assumed that a holding cost over the first sales horizon is included into a product cost, $\mathtt{cst}$. However, if a retailer does not sell all his products during the sales horizons, then he will have to pay for holding the remaining inventory level during the next sales horizon. Therefore, in a stated optimal control problem a final weighting function is $\phi(x(T)) = -h(x(T))$. The term *final weighting function* is taken from ([34], p. 112).

Let $e(t) \in \mathbb{R}$ be our retailer's profit, or earnings, obtained at time t. Note that $e(t)$ is a *weighting function*, the term is taken from ([34], p. 112).

Let $e_{\mathtt{total}} \in \mathbb{R}$ be a total profit of our retailer over the sales horizon. Note that $e_{\mathtt{total}}$ is a performance index of the shopbot system. The optimal control problem is to find a control function $u^*(\cdot)$ which would maximise the performance index. For the sake of readability u might be used instead of $u(\cdot)$.

Now it is possible to state the problem formally.

$$
\begin{aligned}
& \forall t \in [0, T_{\mathtt{final}}]: \\
& \frac{dx(t)}{dt} = -(\mathbf{a} \cdot u(t) + \mathbf{b}); \\
\text{equation for a state :} \quad & x(t) = c - \int_0^t (\mathbf{a} \cdot u(\tau) + \mathbf{b}) d\tau; \\
\text{control constraints :} \quad & u(t) \in \left[p_{\mathtt{zero\ profit}}, u_{\max}\right]; \\
\text{state constraint :} \quad & x(t) \geq 0; \\
& \textbf{if } \mathrm{x(t)} = 0, \ \textbf{then } \mathtt{control\ are\ disabled\ by\ the\ shopbot}; \\
& e(t) = (u(t) - \mathtt{cst}) \cdot ((\mathbf{a} \cdot u(t) + \mathbf{b})); \\
& h(n) = \mathtt{h} \cdot n; \\
\text{performance index :} \quad & e_{\mathtt{total}} = -h(x(T_{\mathtt{final}})) + \int_0^{T_{\mathtt{final}}} e(\tau) d\tau; \\
\text{problem :} \quad & u^* = \arg\max\nolimits_u e_{\mathtt{total}}
\end{aligned}
\tag{6.3.1}
$$

6.3.3. Solution to the problem

There are several inequality constraints in the problem (6.3.1): constraints on the control variable and a constraint on the state variable. In general, an inequality constraint on a state variable makes the problem of finding an optimal control very difficult to solve analytically [38],[39].However, it is possible to find an analytical solution to the problem (6.3.1).

At the beginning the constraint on the state variable is neglected.

Necessary conditions for optimality

The necessary conditions for optimality of a solution to the problem (6.3.1) can be obtained. Derivation of these conditions can be found in Appendix C.

Using the same notation as in Appendix C, a Hamiltonian function is defined as

$$\begin{aligned} H\left(u\left(t\right), x\left(t\right), \lambda(t)\right) = \left(\mathbf{a}u(t) + \mathbf{b}\right)\left(u(t) - \mathtt{cst}\right) + \lambda(t)\left(-\mathbf{a}u(t) - \mathbf{b}\right) = \\ = \mathbf{a}u^2(t) + \left(\mathbf{b} - \mathtt{cst}\cdot\mathbf{a} - \lambda\left(t\right)\right)u(t) - \mathbf{b}\left(\lambda(t) + \mathtt{cst}\right), \end{aligned} \tag{6.3.2}$$

where costate $\lambda(t) \in \mathbb{R}$.

A costate equation

One of the necessary conditions for optimality of a solution to the problem (6.3.1) is a costate equation

$$\frac{\partial H}{\partial x} + \dot{\lambda} = 0. \tag{6.3.3}$$

Taking a partial derivative $\frac{\partial H}{\partial x}$ of the Hamiltonian (6.3.2),

$$\dot{\lambda} = -\frac{\partial H}{\partial x} = 0. \tag{6.3.4}$$

Thus, an optimal costate variable λ^* has to be a constant.

A boundary condition

Another necessary condition for optimality is a boundary condition for the studying problem:

$$\left(\frac{\partial \phi}{\partial x} - \lambda\right)|_T = 0.$$

Therefore, an optimal costate variable $\lambda^*(T)$ at final time T is

$$\lambda^*(T) = -\mathtt{h}.$$

An optimal costate variable has to be a constant. Thus, the value of the optimal costate variable is

$$\lambda^* = -\mathtt{h}. \tag{6.3.5}$$

Pontryagin's Maximum Principle

A set of admissible values for the control variable is an interval $[p_{\mathtt{zero\ profit}}, u_{\max}]$. Another necessary condition for optimality of a solution, therefore, is given by Pontryagin's Maximum Principle:

$$\forall u \in [p_{\mathtt{zero\ profit}}, u_{\max}]:\ H(x^*, u^*, \lambda^*) \geq H(x^*, u, \lambda^*). \tag{6.3.6}$$

The same notation as in Appendix C.3 is used.

Using the Hamiltonian (6.3.2), the expression (6.3.6) can be rewritten as

$$\begin{aligned} \forall u \in [p_{\mathtt{zero\ profit}}, u_{\max}]:\ \mathbf{a}\left(u^*\right)^2 + \left(\mathbf{b} - \mathtt{cst}\cdot\mathbf{a} - \lambda^*\right)u^* - \mathbf{b}\left(\lambda^* + \mathtt{cst}\right) \geq \\ \geq \mathbf{a}u^2 + \left(\mathbf{b} - \mathtt{cst}\cdot\mathbf{a} - \lambda^*\right)u - \mathbf{b}\left(\lambda^* + \mathtt{cst}\right), \end{aligned} \tag{6.3.7}$$

which is equivalent to

$$\forall u \in [p_{\mathtt{zero\ profit}}, u_{\max}]:\ \mathbf{a}\left(u^*\right)^2 + \left(\mathbf{b} - \mathtt{cst}\cdot\mathbf{a} + \mathtt{h}\right)u^* \geq \mathbf{a}u^2 + \left(\mathbf{b} - \mathtt{cst}\cdot\mathbf{a} + \mathtt{h}\right). \tag{6.3.8}$$

Let $y(u) = \mathbf{a}u^2 + (\mathbf{b} - \mathtt{cst} \cdot \mathbf{a} + \mathtt{h})$. According to (6.3.8), an optimal control u^* has to maximise the function $y(u)$ on the interval $[p_{\mathtt{zero\ profit}}, u_{\max}]$.

The minimal value of a control variable is $p_{\mathtt{zero\ profit}} = \mathtt{cst} > 0$. In the Subsection 2.2.2 it is shown that $u_{\max} > p_{\mathtt{zero\ profit}}$, otherwise a retailer choose not to participate in the price competition. An optimal control u^* can be computed.

A costate variable $\lambda^* = -\mathtt{h}$ is a constant, thus $\mathbf{z} = (\mathbf{b} - \mathtt{cst} \cdot \mathbf{a} + \mathtt{h})$ is also a constant. Therefore, the function

$$y(u) = \mathbf{a}u^2 + \mathbf{z}u$$

represents a parabola, as $\mathbf{a} \neq 0$. Note, that $\mathbf{a} < 0$. So, $y(u)$ is concave. As a consequent, $y(u)$ has a global maximum. Note that the parabola $y(u)$ has the same shape for all moments $\forall t \in [0, T_{\mathtt{final}}]$ because $\mathbf{a}$ and $\mathbf{z}$ are constants. Let u^*_{g} be an optimal control for the case when constraints on control are relaxed. Setting $\frac{dy}{du} = 0$,

$$u^*_{\mathrm{g}} = -\frac{\mathbf{z}}{2\mathbf{a}}.$$

The same result could be obtained using a stationary condition for an unconstrained case.

Zeros of the function $y(u)$ are $\{0, \frac{-\mathbf{z}}{\mathbf{a}}\}$.

$\mathbf{b}$ is positive, $\mathbf{a}$ is negative, $\mathtt{h}$ is non-negative. Hence, $(\mathbf{b} - \mathtt{cst} \cdot \mathbf{a} + \mathtt{h})$ is positive. Therefore, $\mathbf{z} > 0$. As a consequence, $\frac{-\mathbf{z}}{\mathbf{a}} > 0$. And $u^*_{\mathrm{g}} > 0$.

If $u^*_{\mathrm{g}} \in [p_{\mathtt{zero\ profit}}, u_{\max}]$, then an optimal control is equal to u^*_{g}. The value u^*_{g} is the same for all possible moments, $\forall t \in [0, T_{\mathtt{final}}]: \ u^* = u^*_{\mathrm{g}} = const$.

If $u^*_{\mathrm{g}} \notin [p_{\mathtt{zero\ profit}}, u_{\max}]$, then for every possible time of interest the function $y(u)$ might be either strictly increasing, or strictly decreasing on the interval $[p_{\mathtt{zero\ profit}}, u_{\max}]$. If $y(u)$ is strictly increasing on $[p_{\mathtt{zero\ profit}}, u_{\max}]$, then an optimal control is again the same for all possible moments $u^* = u_{\max}$. If $y(u)$ is strictly decreasing on the allowed interval of controls, then an optimal control is $u^* = p_{\mathtt{zero\ profit}}$ and is the same for all moments of time as the function $y(u)$ is the same during the sales horizon.

In general a form an optimal control can be given as

$$f(x) = \begin{cases} u^*_{\mathrm{g}}, & u^*_{\mathrm{g}} \in [p_{\mathtt{zero\ profit}}, u_{\max}]\,; \\ p_{\mathtt{zero\ profit}}, & u^*_{\mathrm{g}} < p_{\mathtt{zero\ profit}}; \\ u_{\max}, & u^*_{\mathrm{g}} > u_{\max}, \end{cases} \tag{6.3.9}$$

where $u^*_{\mathrm{g}} = -\dfrac{(\mathbf{b} - \mathtt{cst} \cdot \mathbf{a} + \mathtt{h})}{2\mathbf{a}}$.

To sum up, using necessary conditions for optimality it was proved that an optimal control u^* has to be a constant. Its value is given with Eq. (6.3.9). However, the constraint on the state variable has not been considered yet.

A constraint on the state variable and an optimal state trajectory

In the problem (6.3.1) a current value of the state variable is given by

$$x(t) = c - \int_0^t D(u(\tau))d\tau.$$

It was shown that an optimal control u^* has to be a constant. As a consequence, a corresponding optimal demand $D(u^*) = \mathbf{a} \cdot u^* + \mathbf{b}$ has to be a constant as well. Therefore, an optimal state trajectory would be given by

$$x^*(t) = c - \int_0^t D(u^*)d\tau = c - t \cdot D(u^*). \tag{6.3.10}$$

Thus, a graph of an optimal state trajectory would be a straight line. On a time axis the graph is bounded with the 0 and $T_{\texttt{final}}$. On an axis of an inventory level the upper bound is given with the initial inventory level c, i.e. $x_{\max} = c$, as demand cannot be negative. The lower bound is given with a constraint on state $x(t) \geq 0$, i.e. $x_{\min} = 0$. A point $[0,\ c]$ always belongs to an optimal state trajectory. However, there are several possibilities how the graph can reach a boundary of the allowed region $[0, T_{\texttt{final}}] \times [0, c]$. Four possible cases are depicted in Fig.6.3.1.

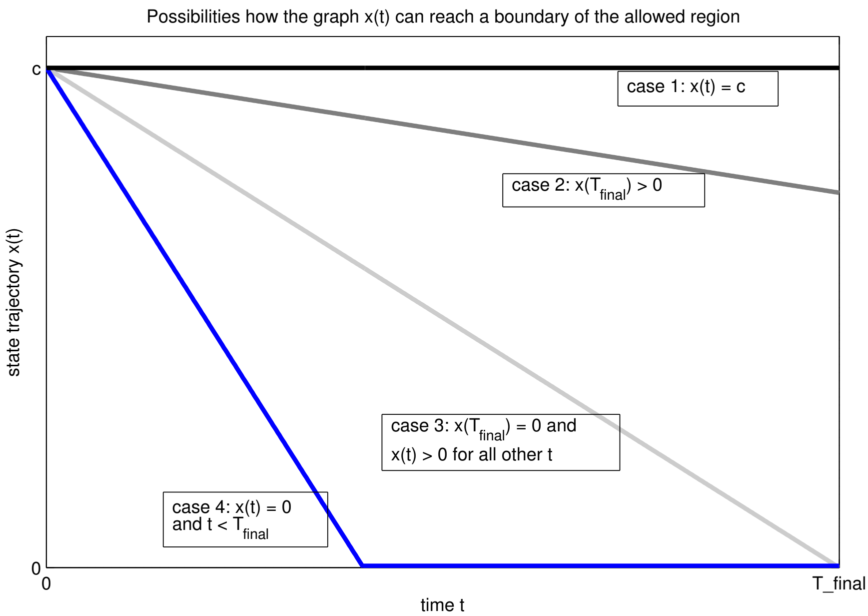

Fig. 6.3.1. Possibilities how the graph x(t) can reach a boundary of the allowed region

Case 1

The graph of an optimal state trajectory might stay on the boundary $x(t) = c$ during the whole period $[0, T_{\texttt{final}}]$. It is the case when a retailer does not participate in the competition but have some inventory level. In this case he will have to pay for holding his entire inventory c during future sales horizons. However, a rational retailer would prefer to sell at least some products even at the price $p_{\texttt{zero profit}}$ and pay a lower holding cost for the remaining inventory level, rather than not participate in the competition at all.

As a conclusion, a retailer with a positive inventory level would prefer to participate in the price competition and an optimal state trajectory would not stay on the boundary $x^*(t) = c$.

A possible optimal state trajectory depicted as the case 4 in Fig.6.3.1 will be examined next.

Case 4 compared to case 3

Let us examine the case when an optimal state trajectory reaches a boundary $x_{\min} = 0$ at time t, $t < T_{\texttt{final}}$, see the case 4 in Fig.6.3.1. Let u_4^* be an optimal control for the possible case 4. In this case all c pieces of a product would be sold out before the end of the sales horizon. This situation will be compared to the possible case 3 from Fig.6.3.1 when all c pieces of a product would be sold exactly at the end of the sales horizon. Let u_3^* be an optimal control for the possible case 3. From Fig.6.3.1 it is obvious, that $D(u_4^*) > D(u_3^*)$. Demand $D(u)$ is decreasing in price, consequently,

$$u_4^* < u_3^*. \tag{6.3.11}$$

As in both cases all pieces of a product were sold out, then a final weighting function

$$\phi(x(T_{\texttt{final}})) = -h(x_4^*(T_{\texttt{final}})) = -h(x_3^*(T_{\texttt{final}})) = -\texttt{h} \cdot 0 = 0$$

. Thus,

$$J_4^* = (u_4^* - \texttt{cst}) \cdot c$$

and

$$J_3^* = (u_3^* - \texttt{cst}) \cdot c.$$

Taking into consideration Eq. (6.3.11),

$$J_3^* > J_4^*.$$

Thus, u_4^* cannot be an optimal control and an optimal state trajectory cannot have the graph depicted as the case 4 in Fig.6.3.1.

To sum up, graphs depicted as cases 2 and 3 in Fig.6.3.1 are the only possible graphs of an optimal state trajectory. Therefore, the constraint on state is equivalent to an inequality

$$0 \leq x^*(T_{\texttt{final}}) < c.$$

It is equivalent to

$$0 \leq c - D(u^*) \cdot T_{\texttt{final}} < c \Leftrightarrow$$

$$\Leftrightarrow \frac{c}{\mathbf{a}T_{\texttt{final}}} - \frac{\mathbf{b}}{\mathbf{a}} \leq u^* < -\frac{\mathbf{b}}{\mathbf{a}}. \tag{6.3.12}$$

Note, that it might happen that $u_{\max} < \frac{c}{\mathbf{a}T_{\texttt{final}}} - \frac{\mathbf{b}}{\mathbf{a}}$, which is equivalent to an expression $c - T_{\texttt{final}}D(u_{\max}) > 0$. It is the case when even the lowest value of demand, $D(u_{\max})$, is so large, that if $u_{\max}$ is applied then a retailer will sell out all his capacity before the end of sales horizon and will be disabled to act by a shopbot. The length T of maximal duration of sales has to be updated.

$$c - TD(u_{\max}) \geq 0 \Leftrightarrow T \leq \frac{c}{D(u_{\max})}. \tag{6.3.13}$$

The maximal duration of sales is

$$T_{\max} = \min\left\{T_{\texttt{final}}, \frac{c}{D(u_{\max})}\right\}.$$

Considering the expression (6.3.12), let

$$u_{\min} = \max\left\{\frac{c}{\mathbf{a}T_{\max}} - \frac{\mathbf{b}}{\mathbf{a}}, p_{\texttt{zero profit}}\right\}. \tag{6.3.14}$$

$u_{\max} = \max\{p_{\texttt{highest}}, -\frac{\mathbf{b}}{\mathbf{a}}\}$, see the Subsection 6.3.1. The price $p_{\texttt{highest}}$ can be an optimal price. However, when $-\frac{\mathbf{b}}{\mathbf{a}} = p_{\texttt{zero demand}} \leq p_{\texttt{highest}}$, the price $p_{\texttt{zero demand}}$ cannot be optimal and has to be excluded from a set of considered prices. In real-world situations a price can take values from a discrete set $S_{allowed} = \{u_1, u_2, ..., u_{\texttt{k}}\}$. It will be assumed that if $p_{\texttt{zero demand}} \leq p_{\texttt{highest}}$, then $u_{\max'} = p_{\texttt{zero demand}} - 1$. Obtained precision would be enough in a majority of practical cases. Moreover, the discussed strategy is based on an approximate estimation of the demand. As a result, dealing with too precise price values might be pointless.

To conclude, an inequality constraint on the state variable was successfully transformed into new inequality constraints on control and a new duration of sales $T_{\max}$.

The pricing strategy

As a summation, the pricing strategy u^* would be presented.

$$
\begin{aligned}
u_{\max'} &= \begin{cases} -\dfrac{\mathbf{b}}{\mathbf{a}} - 1, & p_{\texttt{highest}} \geq -\dfrac{\mathbf{b}}{\mathbf{a}}; \\ p_{\texttt{highest}}, & \text{otherwise}; \end{cases} \\
T_{\max} &= \min \left\{ T_{\texttt{final}}, \frac{c}{D(u_{\max'})} \right\}; \\
u_{\min} &= \max \left\{ \frac{c}{\mathbf{a}T_{\max}} - \frac{\mathbf{b}}{\mathbf{a}}, p_{\texttt{zero profit}} \right\}; \\
u^*_{\mathrm{g}} &= -\frac{(\mathbf{b} - \texttt{cst} \cdot \mathbf{a} + \texttt{h})}{2\mathbf{a}}; \\
u^* &= \begin{cases} u^*_{\mathrm{g}}, & u^*_{\mathrm{g}} \in [u_{\min}, u_{\max'}]; \\ u_{\min}, & u^*_{\mathrm{g}} < u_{\min}; \\ u_{\max'}, & u^*_{\mathrm{g}} > u_{\max'}, \end{cases}
\end{aligned}
\tag{6.3.15}
$$

where a linear demand function $D(u) = \mathbf{a} \cdot u + \mathbf{b}$ is obtained as described in Subsection 6.3.1.

Sufficient condition for optimality

The optimal control was obtained using necessary conditions for optimality. The set of necessary conditions for the considered problem becomes a set of sufficient conditions if it is true that

$$\frac{\partial^2 H}{\partial^2 u} < 0,$$

see Appendix C.4.

$$\frac{\partial^2 H}{\partial^2 u} = \mathbf{a} < 0.$$

Note that it has been already shown that in our case Hamiltonian is a concave function of a control variable.

As a conclusion, the strategy (6.3.15) is an optimal solution to our problem (6.3.1).

To sum up, the pricing strategy for a retailer was developed using methods of continuous optimal control in combination with fictitious learning. It considers temporal structure of the competition, capacity constraints the retailer faces and future holding cost for product pieces which remain after the end of the competition. Previous observations of adversaries' prices are taken into account also. However, the observations are not used directly but rather are approximated with a continuous demand function. Possible inaccuracy of the approximation is a main weak point of the strategy. On the other hand, computation of the price according to the developed strategy has extremely low numerical complexity. It is the main advantage of the obtained strategy.

For the purpose of numerical simulations, the strategy is implemented as a Java class FictitiousOptimalController.java, see the attached CD for details.

Numerical methods

For the sake of completeness, note that there are several numerical methods solving continuous optimal control problem with state inequality constraints, as very often these problem are very hard to solve analytically.

The simplest approach is to use an integral penalty function which is described in ([33], pp. 242,240). If the performance index is J and the constraint on state $C(x) < 0$ is specified for times $[t_0, t_f]$, then it is possible to augment the constraint to the performance index as $J' = J + \mu \int_{t_0}^{t_f} (C(x))^2 \mathbf{1}(C) dt$, where $\mathbf{1}(C) = \begin{cases} 1, C(x) > 0; \\ 0, C(x) < 0, \end{cases}$ and μ is a real-valued constant which has to be positive if J is maximised and negative otherwise.

As an alternative to the penalty function method, a nonlinear programming approach was developed in [38]. The detailed description of the method can be found in [38].

6.4. Strategy 3: dynamic programming approach to optimal control

It is possible to develop another pricing strategy which would consider not only previous observations of the competitors' prices but also the temporal structure of the competition, the retailer's inventory level and the cost of holding the remaining items during the next sales horizon. Dynamic programming in combination with fictitious learning can be used to develop such a pricing strategy. Dynamic programming is "an alternative to the variational approach to optimal control" ([34], p. 260). Dynamic programming can be applied on discrete-time dynamic system even if the system dynamics contains some random parameter ([40], p. 2). In this case the optimal control problem is formulated as an optimization of an *expected* performance index

$$E\left\{\phi(x_N) + \sum_{k=0}^{N} L_k(u_k))\right\},$$

where L_k is a weighted function and ϕ is a final weighted function and $E\{.\}$ is an "expectation with respect to a joint distribution of the random variables involved" ([40], p. 2). In the case of such systems instead of a state equation *transition probability graphs* can be used to describe the system's evolution ([40], p. 9). "These graphs depict the transition probabilities between various pairs of states for each value of the control" ([40], p. 9). The transition probability graph can be obtained for the shopbot system. An inventory level is a state variable. Only non-negative integer values of the inventory level are allowed. Periods of arrivals of potential customers are considered instead of periods of price changes. Periods of arrivals are also considered in [19, 20]. For details on potential customers of our retailer see Section 5.3. If at maximum n potential customers can arrive during a single period of the competition and there are T competition periods in a sales horizon, then there are $n \cdot T$ periods when a potential customer can arrive. From now one in this section by a customer a potential customer of our retailer is meant. Let ξ be a probability that a customer arrives during a single period of arrivals of customers. Let $P_{\text{our,loyal}}$ be a probability that an arrived customer is loyal to our retailer. Let P_{switcher} be a probability that an arrived customer is a switcher, see Section 5.2 for details. And let P_{u_i} be a probability that our retailer's price u_i is the lowest one at the shopbot. Let us examine a case when at the beginning of period t of customers' arrivals our retailer has an inventory level x and sets a price u_i. During a period t no customer arrives with a probability $(1 - \xi)$. In this case before the next period of arrivals our retailer again has an inventory level x. With a probability ξ a potential customer arrives. In this case with a probability $P_{\text{our,loyal}}$ the customer might be loyal to our retailer and the retailer's inventory level at the beginning of the next period of arrivals would be $(x-1)$. At the same time the arrived customer might be a switcher with a probability P_{switcher}. With a probability $(1-P_{u_i})$ the retailer's inventory level at the beginning of the next period of arrivals remains x and with a probability P_{u_i} the retailer's inventory level becomes $(x-1)$ before the beginning of the next period.

To sum up, if during some period of arrivals of customers our retailer's inventory level is x, then it will remain the same if no customer arrives during the period or if an arrived customer is a switcher but our retailer's price is not the lowest available at the shopbot. In other cases at the beginning of the next period of customers' arrivals the retailer's inventory level would be $(x-1)$. Based on the given description of all possibilities, the transition probability graph can be obtained. It depicts evolution of the state variable from the start of a period of arrivals to the end of the period. For every state variable x, $x \in \mathbb{N}\}$ and for every price u_i from a set of allowed prices, $u_i \in S_{allowed}$ the transition probability graph of the state variable before and after the period of arrivals is depicted in Fig. 6.4.1.

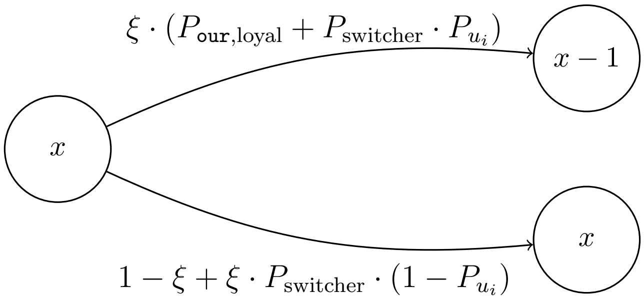

Fig. 6.4.1. Transition probability graph of the shopbot system with natural-valued state variable

Dynamic programming is based on Bellman's principle of optimality ([34], p. 260):

Theorem 6.4.1 (Bellman's principle of optimality)**.** *"An optimal policy has the property that no matter what the previous decision have been, the remaining decisions must constitute an optimal policy with regard to the state resulting from those previous decisions."*

To determine a set of potentially optimal strategies, Bellman's principle of optimality has to be applied backward in time starting from the final stage. The dynamic programming algorithm will be stated for a discrete-time system of the form $x[k+1] = f_k(x[k], u[k], \omega[k])$ with associated performance index

$$E\left\{\phi(x[N]) + \sum_{k=0}^{N-1} L_k(x[k], u[k], \omega[k])\right\},$$

where $x[a]$ is a state, $u[a]$ is a control from an allowed set $S_{allowed}$, ψ and L_k are any functions, ω is a random parameter and its distribution might depend on $x[a]$ and $u[a]$ ([40], p. 5).

Theorem 6.4.2 (The dynamic programming algorithm)**.** *"For every initial state x_0 the optimal performance index $J^*(x_0)$ is equal to $J_0(x_0)$, given by the last step of the following algorithm, which proceeds backward in time from period $N-1$ to period 0:*

$$J_N(x_N) = \phi(x_N),$$

$$J_k(x[k]) = \max_{u[k]\in S_{allowed}} E\left\{L_k(x[k], u[k], \omega[k] + J_{k+1}(f_k(x[k], u[k], \omega[k])\right\},$$

$k = 0, 1, ..., N-1$, where the expectation $E\{\mathbf{.}\}$ is taken with respect to the probability distribution of $\omega[k]$, which can depend on $x[k]$ and $u[k]$. Furthermore, if $u^*[k]$ maximizes the right side of the stated equation for each k, then the control policy $\{u^*[1], ..., u^*[N-1]\}$ is optimal."

The algorithm is taken from ([40], p. 23). Its key recursive equation is sometimes called a functional equation of dynamic programming ([34], p. 264). Note that equilibrium profits in [19, 20] were defined using the same recursive equation.

In case of price competition at shopbots a retailer cannot change his price during every period of arrivals of potential customers. There are n periods of possible arrivals in one period of the competition, therefore, a retailer changes his price once in n periods of customers' arrivals. Also note that according to fictitious play, probabilities P_{u_i} have to be updated after each period of the competition. As a result only a computed optimal price for the current period of the competition would be used. The state variable has to be non-negative. And if at the end of sales horizon a retailer has left l pieces of product, then he has to pay a holding cost $\mathtt{h} \cdot l$ for the next sales horizon.

Dynamic programming algorithm can be implemented as a recursive algorithm with a memory [41]. It is possible to provide pseudo code for computing an optimal price using fictitious learning and dynamic programming. Let $P_{u_i,\text{buy}} = \xi \cdot (P_{\mathtt{our},\text{loyal}} + P_{\text{switcher}} \cdot P_{u_i})$. It is a probability that one piece of a product will be bought from our retailer at price u_i during a period of arrivals of customers. For the sake of compactness, let x denote an inventory level. Let t denote a number of remaining periods of arrivals. Let $u(t-1)$ be a price from a previous period, $u(t-1)$ is required because a retailer can change his price only once during n periods of arrivals. Let j be a helping variable for counting periods of arrivals after the last change of price. If $j = n$, then a retailer can choose an optimal price, otherwise a price from a previous period is set automatically. An expected optimal profit in every period of choice will be stored in memory, so in every period of decisions optimization is carried out once. Optimal price can be computed by calling a function `optimal(`x_{current}, t_{current}, 0, n`)`. The function is given in Algorithm 3. Note, that for the sake of compactness it is assumed that an optimal price is automatically stored into the memory. It is not captured in Algorithm 3 in detail. The returning value of the Function 3 is an optimal expected profit.

To sum up, another pricing strategy for a retailer was developed. It also considers the temporal structure of the competition, the capacity constraints, and the future holding cost for remaining product pieces which remains after the end of the competition. Previous observations of adversaries' prices are taken into consideration as well. Computation of the price based on the presented algorithm might have a high numerical complexity if a number of remaining periods is large and there are many values of possible prices. However, the numerical complexity can be reduced using a simple heuristics. A situation when a retailer would like to sell l pieces during t periods is very similar to a situation when the retailer would like to sell $2l$ pieces during $2t$ periods. As a consequence, a situation when a retailer has x pieces to sell during T period and $T > 30$ will be approximately modelled by a situation when a retailer has $\lceil\frac{30x}{T}\rceil$ pieces to sell during 30 periods. Estimation of an inventory level $\lceil\frac{30x}{T}\rceil$ is rounded up in order to include cases when just one piece of a product is left. Rounding up forces a retailer to sell pieces of a product slightly faster than he would according to the developed dynamic programming algorithm.

For the purpose of numerical simulation, the developed algorithm for computing a pricing strategy is implemented in Java class FictitiousDynamicProgramming.java, see the attached

CD.

```
Function optimal(x, t, u(t - 1), j)
    if an optimal profit for the pair (x,t) is in the memory then
        return a value of an optimal profit from the memory;
    else
        if t == 0 then
            if t == t_current then
                price cannot be set;
            else
                return −h · x;
            end
        else
            if j == n then
                j = 0;
                profit = max_{u_i ∈ S_allowed} {P_{u_i,buy} · (u_i + optimal(x − 1, t − 1, u_i, j + 1)) +
                +(1 − P_{u_i,buy})optimal(x, t − 1, u_i, j + 1)};
                store results into the memory;
                return profit;
            else
                profit = P_{u(t−1),buy} · (u(t − 1) + optimal(x − 1, t − 1, u(t − 1), j + 1)) +
                +(1 − P_{u(t−1),buy})optimal(x, t − 1, u(t − 1), j + 1);
                return profit;
            end
        end
    end
end
```

Algorithm 3: Pseudo code of numerical generation of possible states

7. Evaluation of the quality of pricing strategies

7.1. Overview of the strategies

Several pricing strategies for a retailer at a shopbot have been obtained. It is required to understand limitations of different strategies. The general model of the price competition at shopbots, formulated in Chapter 2, underlines the most important aspects of the competition. The general model makes it possible to better understand weak points of strategies obtained in previous chapters.

First, a simple strategy based on a classical economical model of the Bertrand competition was presented [29]. The strategy is based only on price sensitivity of customers. It does not consider possible customers' loyalty to particular retailers. Temporal structure of the competition is not considered, neither is a retailer's inventory. At the same time the developed strategy uses observations of previous adversaries' prices.

It is obvious that the Bertrand model is too simple model and neglects some important features of the competition, as the model predicts that all retailers in equilibrium would set the same zero-profit prices [29]. The price dispersion at shopbots, however, is rather high [17, 3, 18].

Then a pricing strategy based on a slightly more sophisticated model of Koçaş was derived [5]. Possibility of customers' loyalty is included in the model. On the other hand no observations are considered.

Another pricing strategy is based on even more complex model of an extensive-form game [19]. Temporal structure of a retailer's actions was modelled. Also retailers' inventory levels are taken into account. The strategy uses observations of other retailers' capacities. It might be an unrealistic assumption.

Finally, the model of Koçaş has been extended and several pricing strategies based on that extension were designed. Precisely, the strategy based purely on a concept of fictitious play ([27], p. 206) was designed. This strategy not only models possible loyalty of customers, but also uses previous observations of adversaries' prices.

Combining fictitious play approach with methods of continuous optimal control another strategy was created. In addition to possible loyalty of customers and observations of previous prices, the strategy models the temporal structure of the competition, retailers' capacity constraints and a cost of holding pieces of a product which remain after the sales horizon. All these features of price competition are also taken into consideration in another developed strategy based on fictitious play in combination with dynamic programming. The difference between last two strategies is that continuous optimal control approach does not work directly with the observations, but encodes the observed information into linear function of demand. Dynamic programming works with observations directly without any approximations of the observed values. On the other hand dynamic programming approach computes a pricing strategy numerically and, therefore, has a higher computational complexity comparing to analytical solution of continuous optimal control.

Every strategy has its limitations. It is important to evaluate the performance of different strategies systematically.

7.2. Experimental evaluation of the strategies' quality

7.2.1. Simulation of the competition at a shopbot

For the purpose of the evaluation of strategies, a simulation of the price competition at shopbots was implemented using Java programming language. The simulation can be found on the attached CD. The implemented algorithms have been described in the work.

In the file Retailer.java the basic retailer's behaviour has been implemented in accordance with Section 2.2. The meaning of the class variables, such as a product cost, capacity, follows from a general model in Section 2.2. Competing retailers charge prices and can learn from previous observations. For these purposes class methods `chargingPrise()` and `learn()` are predefined and would be implemented in subclasses of Retailer.java. If a customer buys a product from a retailer, the class method `buyFrom()` is called.

Customers' behaviour described in Section 2.3 is implemented in classes CustomerLoyal.java and CustomerSwitcher.java. The only function of an arrived customer is to choose one offer from available alternatives. A loyal customer is loyal to a particular retailer and always buys from him, while a switching customer chooses one of the cheapest offers at a shopbot.

The stochastic process of customer arrivals to the shopbot is implemented in the file Evaluation.java, which contains implementation of other described processes at a shopbot. Number of arrived customers during a single period of the competition is modelled as the Bernoulli process with a defined maximal number n of customers per period and probability p that one customer arrives, see the Subsection 2.3.4. To model arrivals of customers, a pseudorandom number with a uniform distribution is generated n times. If the value of a generated number is lower or equal to p, than one customer has arrived [42].

Before the start of the competition an array of participating retailers is created. The basic simulation of one sales horizon is going on in a cycle, where each iteration of the cycle represents a single period of the competition. At the beginning of each competition period shopbot collects and stores prices from all retailers. In the simulation a retailer who has nothing to sell sets price $p = -1$. A shopbot deletes all offers with price -1 and sort the remaining retailers according to a price. All retailers observe prices of their adversaries. A number of arrived customers is simulated as discussed above. Numbers of switching customers and customers who are loyal to different retailers are proportional to popularities of the retailer in accordance with Subsection 2.3.2. Customers choose the preferable offers and a new period of the competition starts. Note that some retailers might have the same prices. Such retailers are sorted by a shopbot on base of an additional retailer's parameter `priority`. Initially it is equal to the retailer's id, which excludes the existence of two retailers with the same priority. After each period of the competition, the priority values of different retailers are shuffled. If during all periods of competition all retailers have the same prices, different retailers would be displayed to customers on the first position. A switcher chooses an offer on the first position. In the simulation non-zero product cost `cst` is modelled. If a customer buys a piece of a product from a retailer a price p, then the retailer's profit increases by a value $(p - \texttt{cst})$. A cost for holding pieces of a product which remains after the end of sales horizon is modelled in the simulation as well.

In the file Evaluation.java numerical experiments are also implemented, design of experiments is discussed below.

Different retailers' strategies were implemented in subclasses of a class Retailer.java according to strategies described in the work. The only strategy which was not discussed in the work is a static strategy. It is a simple strategy when a retailer does not change his price at all. The summation of files names and strategies implemented in those file is given in the Table 7.2.1.

File	Strategy implemented in the File
BertrandRetailer.java	Strategy based on the Bertrand model, Section 4.1
StaticRetailer.java	Strategy when a retailer does not change a price
KocasRetailer.java	Strategy based on the model developed of Koçaş,section 4.2
MartinezTalluriRetailer.java	Strategy based on the model of Martínez-de-Albéniz, Talluri, Section 4.3
Fictitious.java	Strategy based on fictitious play, Section 6.1
FictitiousOptimalController.java	Strategy based on fictitious play and optimal control, Section 6.3
FictitiousDynamicProgramming.java	Strategy based on fictitious play and dynamic programming, Section 6.4

Tab. 7.2.1. Summation of strategies implemented in different files on the attached CD

The last three strategies are based on the file DemandEstimator.java which contains implementation of the method for the demand estimation developed in the Chapter 5. Probability of ties inside any subset containing 1,2,3,4 and 5 retailers was considered, other possible ties were neglected due to the computational complexity. The neglected ties have a low probability.

7.2.2. Method for an experimental evaluation of the strategies' quality

The developed pricing strategies differ significantly. It is required to obtain a unified method for evaluation of a strategy's quality. For this purpose the performance of tested strategies is measured in head-to-head competitions between retailers who use different strategies. Using this approach a comparative evaluation of the developed strategies' quality can be obtained. However, it is pointless to use any of the complicated strategies if its performance is worse than a performance of some very simple strategy. As a simple baseline for comparison of different developed strategies static strategies are chosen, when a retailer always sets a constant price.

One simple static strategy is to always set the highest price $p_{\texttt{highest}}$ which can attract some customers. For cases when a total demand on a product is so high that all retailers sell all their capacities this static strategy is obviously optimal. Another simple strategy is to always set a price slightly higher than a minimal one and try to attract switchers, price $p_{\texttt{zero profit}} + 1$ was used to model this type of a static strategy in numerical experiments. In the simulation natural-valued prices are assumed. Thus, $p_{\texttt{zero profit}} + 1$ is the lowest positive-profit price.

Let every head-to-head competition be called a *match*. The performance of every strategy against other strategies should be evaluated in matches under different conditions which can occur in real world. Parameters of numerical experiments which would capture different real-world situations are discussed in the next section.

7.2.3. Parameters of numerical experiments

Parameters of different possible numerical experiments have to be discussed. The set of all parameters should capture all possible real-world situations.

When two retailers compete with each other, several cases are possible. A retailer and his adversary might be in the same position, the retailer might be in a better position or in a worse position. For instance, if one retailer has more loyal customers than another, then the first retailer can easier gain a higher profit [5, 3] and, therefore, is in a better position no matter what strategy he uses.

To uniformly evaluate performance of a strategy, all possible cases have to be tested in numerical experiments. Considering a retailer's popularity next three possibilities have to be tested: **a)** a tested retailer is more popular than each adversary,

b) a tested retailer is less popular than each adversary,

c) they both have the same popularity. By a tested retailer a retailer who adopts a tested strategy is meant. By the retailer's adversaries all possible adversaries are meant.

A number of addressable switching customers is usually considerably greater than a number of loyal customers [5]. In numerical experiments the number of switchers would be assumed to be ten times larger than a number of customers loyal to a more popular retailer. A number of switching customers was equal to $\psi = 50$ in all simulations.

In numerical experiments with different popularities of retailers a number of customers loyal to a more popular retailer was $\chi_{\text{popular}} = 5$, a number of loyal customers of a less popular retailer was $\chi_{\text{notpopular}} = 1$.

In numerical experiments with the same popularity of both retailers a number of customers loyal to a retailer was $\chi_1 = \chi_2 = 5$.

In accordance with the discussion in Section 5.3, numbers of loyals and switchers were viewed as general characteristics of the structure of arrived customers.

In real-world situations retailers differ in sizes of their capacities and lengths of sales horizons. It is important to estimate performance of every strategy in a case when a retailer has a given number of product pieces to sell during a particular sales horizon. Moreover, for a given pair of the retailer's capacity and sales horizon an adversary might have different combinations of his capacity size and sales horizon duration. However, the situation when an adversary would like to sell k pieces during future n periods exerts very similar constraints on the adversary as when he would like to sell $2k$ pieces during $2n$ periods. Thus, to model different situations when different retailers have different time and capacity constraints in numerical experiments of head-to-head competitions it would be assumed that both retailers have the same sales horizons and differ only in capacities.

Next situations were considered:

a) a tested retailer has a higher capacity than his adversary;

b) the retailer has a lower capacity compared to the adversary's capacity;

c) both retailers have the same capacity.

It is impossible to know for sure if a greater capacity is an advantage. It might be. However, if a retailer has too large capacity and a total demand for a product is very low, i.e. almost no customer arrives to a shopbot page, then the retailer has to pay considerable costs for holding his capacity and a retailer with a smaller capacity might be in a better position.

To make a conclusion, not only different capacities but also different cases of the total demand compared to the capacity levels have to be considered. For the case of different capacities there are several possible levels of the demand in relation to retailers' capacities. In the first case the demand is so low that even a retailer with a smaller capacity has at the beginning of sales horizon more product pieces than a total demand. Let this case of demand be called a *low demand*. Note, a total number of customers expected during a sales horizon is meant by the total demand. Also it is possible that the total demand enables a smaller retailer to sell all his capacity if he is the only seller. However, at the same time at the start of competition a larger retailer might have more product pieces than a total demand. This case would be called an *average demand*. If the total demand enables the largest retailer to sell all his capacity but is lower than a total capacity of both retailers, then the demand would be called a *high demand*. And it the total demand is such that both retailers can sell their capacities, then it would be called a *super high demand*.

To conclude, a structure of numerical experiments required for a rigorous evaluation of a strategy's performance was proposed.

7.3. Unified indices of performance and values of simulations' parameters

It is required to uniformly characterise performance of different strategies. For this purpose a *unified index of performance* should be used. The greater profit the strategy gains, the more efficient the strategy is.

Relative levels of retailers' popularity, relative capacity levels and demand level specify one particular contest situation for a tested strategy. Every tested strategy was put in different contest situations. For instance, one such contest situation can arise when the tested retailer is more popular than his adversary, and at the same time he has the same capacity as the adversary, and the demand is high. As an adversary's strategy, all strategies were taken one by one. Each match against each adversary under defined conditions was simulated 20 times. Average profit of the tested retailer against a given adversary was computed. After it a sum of average profits against all possible adversaries was computed. Note that a standard deviation of the total sum was obtained as a square root of a sum of variances of profits obtained in competitions against different adversaries. A variance of sum is equal to a sum of variances because each competition was independent on other competitions. Tables with profits of different strategies against all possible adversaries in each contest situation can be found on the attached CD in the folder experiments. For instance result profits for the mentioned contest situation can be found in the folder experiments/more popular than an adversary/the same capacity as an adversary/demand is high. Tested retailer is a row retailer in the contest tables on the attached CD. For the sake of the tables' compactness different designed strategies are labelled with numbers and abbreviations, see Table 7.3. In the Table 7.3 it shown what features of real-world competition every strategy considers.

The overall performance of the strategy should characterize how efficient the strategy is in general. A unified index characterising the overall strategy's performance should be designed. Some strategies might be more successful in particular contest situations, another might perform better in another situations. However, for the case when a retailer knows nothing about a real-world situation he would like to choose a strategy which is efficient in general, for different real-world situations. Thus, the unified index is intended to be a general parameter of the strategy's performance. Successes in different possible contest situations should have the same weight in the result index. In order to ensure this, maximal expected profits of the tested retailer have to be set equal in different competitions. Maximal expected profit is defined as a product of a maximal price and a maximal expected number of sold pieces of a product during the sales horizon. Maximal price is the same in all simulated competitions. Maximal expected number of sold pieces of a product is $\text{sold}_{\max} = \min\{c, E\{D\}\}$, where c is a retailer's capacity and $E\{D\}$ is an expected number of arrivals of customers during the sales horizon. Parameters of different contest situations must be chosen in such a way that $\text{sold}_{\max}$ is constant. Arrivals of customers are described

Labels	**File**	observation	loyalty of customers	time	capacity	holding cost
1/StH	StaticRetailer.java($p_{\texttt{highest}}$)					
2/StL	StaticRetailer.java($p_{\texttt{zero profit}}$)					
3/Bert	BertrandRetailer.java	✓				
4/Koc	KocasRetailer.java		✓			
5/MT	MartinezTalluriRetailer.java	✓		✓	✓	
6/Fic	Fictitious.java	✓	✓			
7/Opt	FictitiousOptimalController.java	✓	✓	✓	✓	✓
8/Dyn	FictitiousDynamicProgramming.java	✓	✓	✓	✓	✓

Tab. 7.3.1. Numbering of strategies in tables of results of numerical experiments

with the Bernoulli stochastic process with parameters n and ξ, where n is a maximal number of customers during a period of the competition. The length of a sales horizon was set to be equal to 30. Hence, a total expected demand is given as $E\{D\} = 30 \cdot n \cdot \xi$. Capacities and the demand parameters for simulation of different contest situations were chosen in such a way that sold_{max} is constant, see Table 7.3 for details on parameters of different situations. In the Table 7.3 c_{tested} and c_{ad} stand for capacities of a tested retailer and his adversary. Note that demand level is determined in relation to capacity levels of both retailers. In simulations a capacity c_{larger} of a larger retailer was always taken to be two times larger than a capacity c_{smaller} of the smaller retailer. The case of a low demand was always modelled as $E\{D\} = 0.5c_{\text{smaller}}$. The case of average demand was given with $E\{D\} = c_{\text{smaller}} + 0.5(c_{\text{larger}} - c_{\text{smaller}})$. The case of average demand should not be modelled if both retailers have the same capacity, as in this case total demand enables every retailer to sell his capacity and, hence, the situation is the same as the one described as a high demand. The case of high demand is modelled with $E\{D\} = c_{\text{larger}} + 0.5c_{\text{smaller}}$. And the case of super high demand is modelled as $E\{D\} = 1.5(c_{\text{larger}} + c_{\text{smaller}})$.

	$c_{\text{tested}} < c_{\text{ad}}$	$c_{\text{tested}} = c_{\text{ad}}$	$c_{\text{tested}} > c_{\text{ad}}$
low demand	$c_{\text{tested}} = 60,\ c_{\text{ad}} = 120,\ n = 2,\ p = 0.5$	$c_{\text{tested}} = 60,\ c_{\text{ad}} = 60,\ n = 2,\ p = 0.5$	$c_{\text{tested}} = 120,\ c_{\text{ad}} = 60,\ n = 2,\ p = 0.5$
average demand	$c_{\text{tested}} = 30,\ c_{\text{ad}} = 120,\ n = 3,\ p = 0.5$	not modelled	$c_{\text{tested}} = 40,\ c_{\text{ad}} = 20,\ n = 2,\ p = 0.5$
high demand	$c_{\text{tested}} = 30,\ c_{\text{ad}} = 60,\ n = 3,\ p = 5/6$	$c_{\text{tested}} = 30,\ c_{\text{ad}} = 30,\ n = 3,\ p = 0.5$	$c_{\text{tested}} = 30,\ c_{\text{ad}} = 15,\ n = 2,\ p = 0.625$
super high demand	$c_{\text{tested}} = 30,\ c_{\text{ad}} = 60,\ n = 5,\ p = 0.9$	$c_{\text{tested}} = 30,\ c_{\text{ad}} = 30,\ n = 4,\ p = 0.75$	$c_{\text{tested}} = 30,\ c_{\text{ad}} = 15,\ n = 3,\ p = 0.75$

Tab. 7.3.2. Parameters of numerical experiments for different possible cases of capacities and demand

The last remark is about a set of allowed prices. In experiments a product cost was assumed to be equal to 10, $\texttt{cst} = p_{\texttt{zero profit}} = 10$. The highest possible price was $p_{\texttt{highest}} = 100$. A holding cost per piece of product was considered to be `h`.

7.4. Result indices of performance

Every single match between retailers is documented, see the attached CD. Comparative tables of results for every contest situation are given. Graphs of mean retailers profits, prices and inventory levels are provided. See Appendix D for detailed description of the CD's contents. The most important is to analyse a general performance of each strategy.

Note that the nature of price competition might evolve in time. For that reason total set of all possible contest situations was simulated for two different durations of the competition. First, the short-term competition was considered. Its duration was a single sales horizon.

Total results of the whole large amount of contest situations are summarized in the Table 7.4.1.

	1/StH	**2**/StL	**3**/Bert	**4**/Koc	**5**/MT	**6**/Fic	**7**/Opt	**8**/Dyn	**sum**	**rank**
1	639 ± 15	400 ± 13	404 ± 11	399 ± 12	526 ± 13	396 ± 11	634 ± 15	403 ± 12	3801 ± 36	4
2	-9 ± 0	-15 ± 0	-19 ± 0	-10 ± 0	-13 ± 0	-19 ± 0	-10 ± 0	-9 ± 0	-104 ± 1	8
3	829 ± 13	-16 ± 2	495 ± 16	39 ± 5	581 ± 12	492 ± 16	827 ± 13	596 ± 12	3843 ± 34	3
4	139 ± 7	55 ± 5	65 ± 4	64 ± 5	101 ± 6	60 ± 6	128 ± 6	95 ± 7	707 ± 17	7
5	527 ± 10	322 ± 10	355 ± 11	327 ± 11	446 ± 10	364 ± 12	526 ± 9	380 ± 13	3248 ± 31	6
6	828 ± 12	-16 ± 2	548 ± 12	60 ± 14	592 ± 10	545 ± 12	831 ± 12	607 ± 11	3993 ± 32	2
7	637 ± 15	390 ± 12	401 ± 12	330 ± 11	523 ± 14	403 ± 11	631 ± 15	409 ± 12	3723 ± 36	5
8	840 ± 12	397 ± 12	538 ± 11	280 ± 16	592 ± 13	543 ± 12	836 ± 11	580 ± 11	4607 ± 35	1

Tab. 7.4.1. Total profits of a row retailer in all possible experiments. All values are divided by 100 for the sake of compactness.

It can be seen that the strategy Dyn, based on dynamic programming, appears to be the most efficient in short-term competition.

Therefore, consideration of key features of the competition can be helpful for maximising a total profit. However, it is obvious from the Table 7.4.1 that observation of adversaries' prices is the most important component of a short-term success. The strategy Opt, based on optimal control, considers key features of the competition as successful Dyn does. However, inaccuracy of approximation of the observations with a linear function results in the fifth position for the strategy Opt. At the same time a simpler strategy Fic is on the second position. The strategy Fic considers only observations of previous prices and possible loyalty of customers. The observations are analysed according to the technique developed in Section 5.1. It is worth noting that the top-ranked strategy Dyn is a direct extension of the second-ranked Opt. The performance increased approximately by 15% due to consideration of the capacity constraints, the temporal structure of the competition and the holding costs. The strategy Bert based purely on observations of prices is on the third position. However, its performance was not considerably better than the performance of the static strategy StH who sets the highest reasonable price. This static strategy is optimal in cases when demand is so high that both retailers will definitely sell all their capacities. Also the considered static strategy maximises the profit obtained from selling to loyal customers.

The strategy MT is placed on the sixth position. It does not consider observations of prices but rather observations of the capacity. It can be seen that pricing based on retailers capacities and total demand does not lead to maximal profits, but it provides approximately stable profits no matter what a competitor does.

The same holds for pricing strategies Opt and Dyn. It should be underlined that none of strategies at positions 2-5 dominates all others considerably. The second-ranked strategy Fic gained a profit greater than a profit of the fifth Opt only by 7.2%. At the same time the strategy Fic might end up with even negative profits when a competitor acts in accordance with the strategy StL. Thus, the performance of the strategy StL strongly depends on a type of a competitor.

The strategy StL might be very inefficient due to considered holding cost. Neither the mixing strategy Koc is efficient. It does not use any observations. However, the strategy Koc cut down prices of the strategies Bert and Fic. Note, that if the competition goes on for a longer time, then the strategy Fic can better estimate mixed-strategy of Koc and starts to response better. It can be seen from Table 7.4.2.

	1/StH	**2**/StL	**3**/Bert	**4**/Koc	**5**/MT	**6**/Fic	**7**/Opt	**8**/Dyn	**sum**	**rank**
1	2607 ± 28	1641 ± 24	1647 ± 23	1646 ± 23	2156 ± 27	1636 ± 23	2616 ± 28	1669 ± 24	15619 ± 71	2
2	4 ± 1	-15 ± 1	-36 ± 1	3 ± 1	-12 ± 0	-36 ± 1	4 ± 1	4 ± 1	-84 ± 1	8
3	3310 ± 25	-42 ± 2	339 ± 36	190 ± 9	2308 ± 22	1530 ± 20	3317 ± 25	2052 ± 20	13003 ± 63	6
4	594 ± 14	253 ± 11	297 ± 10	295 ± 9	446 ± 12	261 ± 12	557 ± 13	394 ± 17	3096 ± 35	7
5	2160 ± 18	1340 ± 20	1463 ± 22	1347 ± 21	1840 ± 21	1455 ± 23	2153 ± 19	1490 ± 21	13247 ± 59	5
6	3391 ± 22	404 ± 30	1339 ± 17	483 ± 56	2361 ± 20	1606 ± 18	3397 ± 25	2023 ± 19	15005 ± 81	4
7	2611 ± 26	1616 ± 25	1646 ± 24	1441 ± 24	2146 ± 27	1647 ± 22	2625 ± 29	1662 ± 23	15394 ± 71	3
8	3414 ± 27	1640 ± 24	1662 ± 19	1220 ± 35	2430 ± 21	1775 ± 19	3401 ± 24	2114 ± 20	17655 ± 69	1

Tab. 7.4.2. Total profits of a row retailer in all possible experiments. All values are divided by 100 for the sake of compactness.

Table 7.4.2 presents results of the simulation with the length of four sales horizons.

Strategy Dyn again outperforms all other strategies. It is obvious that the strategy Bert is not as efficient for long-term competition as it was for a case of short competition. It can be seen that competition between two Bert retailers leads to negative-profits, which are possible due to non-zero holding cost. At the same time the strategy Fic performs better against Koc or against itself as the time goes on.

In general, to be successful in long-term competition a retailer has to take loyal customers into consideration and try to earn as much as possible when demand is high. Results of the strategy StH suggest it. StH can be viewed as a good option in general, and it is especially efficient when demand is high or the retailer has a lot of loyal customers. The top-performed strategy Dyn also tries to maximise its profit from loyal customers when demand is high. The only unpleasant adversary for Dyn is Koc, which uses mixed strategies. Koc sometimes sets reasonably high prices. As a consequence, there are jumps in standard deviations of prices Koc set, see Fig. 7.4.1. And if Koc sets a high price, then Dyn tries to set a price slightly lower than the maximal one, see Fig. 7.4.1. As a result, if a Dyn's loyal customer arrives, then he buys at a lower price, then the maximal one. And Dyn does not maximise his profit from selling to loyal customers.

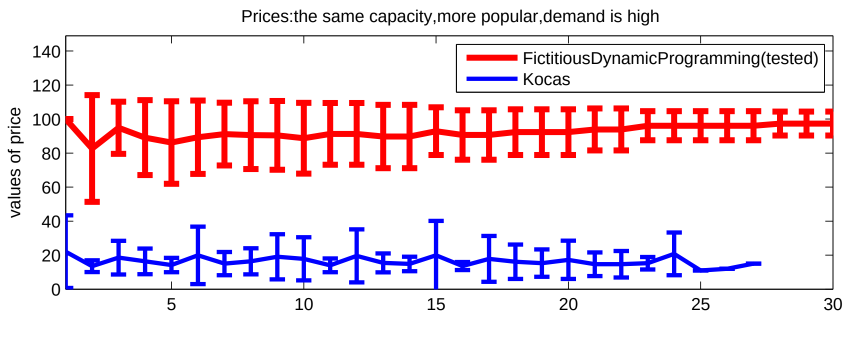

Fig. 7.4.1. Trajectories of prices for the case when Dyn is more popular and demand is high and his adversary is Koc.

Note that in a long term a strategy based on continuous optimal, Opt, is also efficient. Interestingly, due to its imprecise estimation of the demand function, mixed-strategy of Koc does not confuse Opt as much as Dyn.

8. Conclusions

In this thesis price competition among retailers at shopbots was investigated. The aim of the work was to understand the nature of the price competition, capture the most important aspects of it and design pricing strategies using methods of game theory and optimal control.

First, available empirical work on price competition at shopbots was surveyed. Based on it the most important features of the problem were underlined and included into a general model of the competition. The model captures key aspects of retailers' and customers' behaviour.

Next, existing related models of price competition were studied. Different models neglect different real-world aspects of the problem. Based on the models several pricing strategies were obtained. One strategy considers only observation of previous prices. Another strategy considers only possible loyalty of customers. The most complex strategy takes the capacity constraints and the temporal structure into consideration. The strategy requires observations of competitors' inventory levels.

The simple model considering possible loyalty of customers was extended and the temporal structure was included into the model. A technique for learning from observations of competitors' prices was developed for the extended model. Based on the technique and using a game theoretic learning rule further strategy was obtained. The strategy, therefore, takes into account previous observations and loyalty of some customers. Fictitious play was used as the learning rule.

Price competition was formulated as an optimal control problem in order to combine the estimations of adversaries' behaviour with consideration of a retailer's capacity and time. Using the developed learning technique and optimal control two further strategies were designed. One strategy was obtained based on variational approach to continuous optimal control. It was shown that the solution in this case can be obtained analytically. It is a strong point of the method. On the other hand, the continuous model does not capture the observed information precisely.

As an alternative, dynamic programming approach to optimal control was used. The strategy developed using dynamic programming considers the same real-world features of the competition and deals with previous observations directly. On the other hand, the implementation of the strategy has higher numerical complexity. However, a heuristic modification can be suggested in order to reduce numerical complexity.

An advantage of both strategies is that observation of adversaries' inventory levels is not required in contrast with the strategy from the related model. On the other hand, strategies are based on an assumption that competitors' pricing strategies can be modelled with probabilistic pricing [5, 6].

Performance of different strategies was evaluated numerically. For the purpose of the evaluation a computational simulation of the price competition was implemented. The simulation captures the structure of the competition from the developed general model and can be used to test various pricing strategies.

A set of numerical experiments for comparative evaluation of different strategies was proposed. The experiments enable determination of the most efficient strategy compared to all others in general case when nothing is known about demand and a competitor. Static strategies were included into a set of evaluated strategies as a baseline for comparison.

In numerical experiments it was found out that pricing strategies using observations of previous prices might be efficient only in short-term competition. However, even in short-term competition the strategy based on fictitious play in combination with dynamic programming outperforms all other strategies. When a competition goes on for a longer time, then in order to maximize profits, it is important to consider possible loyalty of customers, an inventory level, sales horizon and a total expected demand during sales horizon. In long-term competition the dynamic programming approach again provides in general the best results. A simple static strategy of setting the highest reasonable price performs well too. In a general case performance of the strategy based on fictitious play in combination with optimal control is roughly the same as performance of the static strategy.

There are several possible future research directions. Allowing retailers to choose their capacities a new type of competition can be obtained where retailers compete both on capacities and prices [43]. Another important extension of the considered problem might be a simultaneous competition on several products. It is possible that a customer attracted to a retailer's web page by the cheapest offer of one product would also buy additional products from the retailer [3]. It might be also important to examine when it is profitable for a retailer to invest in customers' loyalty.

It might be interesting to examine aspects of practical application of the most promising strategy based on dynamic programming. The strategy does not require any additional observations about competitors except their prices. However, it might be necessary to improve the model of demand and investigate in detail methods of demand estimation.

New numerical experiments for evaluation of strategies might be designed as well. The proposed method of evaluation is based on head-to-head competitions. It evaluates every strategy in a unified way, as all possible situations in relation to another strategy and demand are considered. However, it is computationally difficult to address all arising situations if all possible subsets of tested strategies are taken into consideration. Approaches to unified evaluation of multiple-player competition are to be designed.

Appendix A.

Game-theoretic definitions

A.1. Normal-form game

Definition A.1.1. *"The simplest game definition, known as a normal form, contains a tuple (N, A, u), where:*

- *N is a finite set of n players who are taking the action.*[1]
- *$A = A_1 \times A_2 \times ... \times A_n$ and A_i is a set of actions available to the i^{th} player.*[2]
- *$u = (u_1, u_2, ..., u_n)$ and $u_i : A \mapsto \mathbb{R}$ is a utility function of the i^{th} player. The utility function of a player maps every vector of actions $a \in A$ to a real-valued number. This number characterises a degree of the player's happiness about the outcome of the vector of actions a, which is also called an action profile. The greater the $u_i(a)$ is, the more satisfied the i^{th} player is about the outcome of the action profile a*[3] *([27], p. 118)."*

A normal-form game is very often seen as the most fundamental one. For some problems other game formulations might be more appropriate. Nevertheless, in game-theoretic works many other game formulations are very often represented using an "induced normal form" as well [27]. Normal-form game is also called a single-stage game to underline the fact, that the game has no temporal structure.

A.2. Extensive-form game with perfect information

In some cases it might be desirable to represent a temporal structure of the game. In this case a so-called extensive-form game formulation might be used. The extensive-form game explicitly deals with time. Very often an explicit temporal structure of the game is desired to model a game where players perform their actions one by one. In some cases, though, an extensive-form game where players act simultaneously might be useful as well. And a game is called a game with perfect information if every player knows all actions chosen by all other players previously when choosing his next action. Such an extensive-form game with perfect information is also called a multiple-stage game with observed actions [44]. All players act simultaneously at every stage of the game and know actions of other players during all previous stages. We will give here a formal definition of an extensive-form game with simultaneous actions, as some of the models of competition similar to the one at shopbots deal with this type of extensive-games.

Definition A.2.1. *"A finite extensive-form game is a tuple $G = (N, A, H, Z, \chi, \delta, u)$, where:*

- *N is a set of n players;*

[1] Analysing strategic interactions among retailers at shopbots, it is obvious that a finite set of players is a set of retailers at some shopbot.

[2] See 2.2.2.

[3] Let us note, that in the case of a competition among retailers at shopbots the greater the retailer's profit is, the happier the retailer is. Therefore, the retailer's utility function would be an amount of money the retailer earns in the defined game.

- *A is a single set of actions;*
- *H is a set of non-terminal choice nodes;*
- *Z is a set of terminal nodes, disjoint from H;*
- *$\chi = (\chi_1, ..., \chi_n)$, where $\chi_i : H \mapsto 2^A$ is the action function of the i^{th} player, which assigns to each choice node a set of possible actions for the player;*
- *$\delta : H \times A \mapsto H \cup Z$ is the successor function, which maps a choice node and an action to a new choice node or terminal node such that for all $h_1, h_2 \in H$ and $a_1, a_2 \in A$, if $\delta(h_1, a_1) = \delta(h_2, a_2)$ then $h_1 = h_2$ and $a_1 = a_2$;*
- *$u = (u_1, ..., u_n)$, where $u_i : Z \mapsto \mathbb{R}$ is a real-valued utility function of the i^{th} player on the terminal nodes Z."*

The definition was taken from ([27], p. 118) and has been slightly modified.

Note that from a graph theory point of view an extensive-form game is represented with a tree.

A.3. Players' strategies

Analysing the game among retailers at a shopbot, as a result we would like to understand actions the retailers would have the incentive to choose.

In game theory a set of all available player's choices is called a set of his strategies. One type of strategies is a pure strategy [27]. For a normal-form game it means that a player chooses one action and plays it. A selection of an action for every player is called a pure-strategy profile. A pure strategy for extensive-form game is a complete description of actions taken at every choice node of the game. Let us use the definition of pure strategies for an extensive-form game from ([27], p. 119):

Definition A.3.1. *"Let $G = (N, A, H, Z, \chi, \delta, u)$ be an extensive-form game. Then the pure strategies of the i^{th} player consist of the Cartesian product $\prod_{h \in H} \chi_i(h)$."*

We have already discussed pure strategies. Another type of strategies is a mixed strategy. Let us use the definition of mixed strategies for a normal-form game from ([27], p. 59):

Definition A.3.2. *"Let (N, A, u) be a normal-form game, and for any set X let $\prod(X)$ be the set of all probability distributions over X. Then the set of mixed strategies for the i^{th} player is $S_i = \prod(A_i)$. The set of mixed-strategy profiles is simply the Cartesian product of the individual mixed-strategy sets, $S_1 \times ... \times S_n$."*

Note that a pure strategy of normal-form game might be viewed as a special case of mixed strategies, where only one action has a probability 1, and other actions have a zero probability.

A.4. Equilibrium

If we know strategies of all retailers except the i^{th} retailer and assuming that the retailers are rational it is possible to predict a strategy of the i^{th} retailer. The i^{th} retailer would choose such a strategy which maximizes his own utility. Such a maximizing strategy as a response to a given set of other retailers' strategies is called a best response of the i^{th} player [27]. One of the most powerful and stable solution concepts in game theory is a Nash equilibrium which can be defined using the best response definition. Let us use definitions from ([27], p. 62):

Definition A.4.1. *"Let S_i be a set of possible strategies for the i^{th} player.*

Let $s_{-i} = (s_1, ..., s_{i-1}, s_{i+1}, ..., s_n)$ be a strategy profile without a strategy of the i^{th} player. The i^{th} player best response to the s_{-i} is a strategy $s_i^* \in S_i$ such that $u_i(s_i^*, s_{-i}) \geq u_i(s_i, s_{-i})$ for all strategies $s_i \in S_i$."

Definition A.4.2. *"A strategy profile $s = (s_1, ..., s_n)$ is a Nash equilibrium if, for all agents i, s_i is a best response to s_{-i}."*

Definitions of the best response and Nash equilibria are the same in case of extensive-form games as they are in case of normal-form games. However, there is a stronger notion of equilibrium for extensive-form games which is called a subgame-perfect equilibrium [27]. We will give here definitions taken from ([27], pp. 122-123).

Definition A.4.3. *"Given an extensive-form game G, the subgame of G rooted at node h is the restriction of G to the descendants of h. The set of subgames of G consists of all of subgames of G rooted at some node in G."*

Definition A.4.4. *"The subgame-perfect equilibrium of an extensive-form game G are all strategy profiles s such that for any subgame G' of G, the restriction of s to G' is a Nash equilibrium of G'."*

Appendix B.

Numerical complexity of simulations using state representation suggested in Subsection 6.3

In this Appendix the time complexity and the memory complexity of numerical simulations using state representation suggested in Subsection 6.3 are examined. First, number of different states generated is addressed. It provides an insight into the memory complexity of the simulation. Then the time complexity is mentioned.

In this Appendix the same notation is used as in Section 5.2. The obtained demand function is not expressed analytically. It is given by the demand vector $\mathbf{D} = (D_1, D_2, ..., D_{\mathtt{k}})$, where $\mathtt{k}$ is a number of allowed prices our retailer can set and D_i is an expected value of a number of customers who would buy a product piece from our retailer if he sets a price u_i.

B.1. Number of states generated during the simulation

Before starting the first period of the competition there is just one state given by a value of an initial inventory level. There is a finite number $\mathtt{k}$ of allowed prices which can be applied in every state. In some cases values D_i and D_j for different allowed prices u_i and u_j might be the same, for instance, it is the case if each adversarial retailer sets prices u_i and u_j with the same probability. The worst case will be assumed, when all possible values of demand D_i are different. Thus, after the first period of the competition there are $\mathtt{k}$ new possible states. Note that a constraint on the state variable values is not considered for now. Alternatively, it is assumed that $D_{\mathtt{min}} = \min\{D_i, i \in \mathbb{N} \wedge i \leq \mathtt{k}\}$, and that even for the last period of the competition with an ordinal number $\mathtt{last}$: $D_{\mathtt{min}} \cdot \mathtt{last} \leq x_{\mathtt{initial}}$. This assumption provides estimation of the memory complexity in the case of a large initial inventory level and short sales horizon.

During the first period of the competition $\mathtt{k}$ new states were generated. In every new state again $\mathtt{k}$ allowed prices can be applied. After it values of the state variable in some new states would merge. For instance, if a price u_w is applied in the first period of the competition and a price u_q is set in the second period, then at the beginning of the third period of the competition the value of the state variable will the same as if a price u_q had been applied in the first period of the competition and a price u_w had been applied in the second one. In fact, a number of states generated during the second competition period and described with different values of the state variable is equal to a number of all possible 2-combinations with repetitions from a set of allowed prices. Thus, a total number of possible states after first N competition periods is

$$\sum_{i=0}^{N} \binom{\mathtt{k} + i - 1}{i}. \tag{B.1.1}$$

Using Eq. (B.1.1), it is possible to estimate a number of generated states for some particular problem. It is assumed that for some product all allowed prices are natural numbers from an interval $[100, 200]$ and that some retailer has a sales horizon ten periods long. The case when $D_{\mathtt{min}} \cdot 10 \leq x_{\mathtt{initial}}$ is considered first, where $D_{\mathtt{min}} = \min\{D_i, i \in \mathbb{N} \wedge i \leq \mathtt{k}\}$. A number of different possible values of the state variable during the last competition period is

$$\binom{101+10-1}{10} \approx 4.7 \cdot 10^{13}.$$

A total number of different state variable values is

$$\sum_{i=0}^{60} \binom{101+i-1}{i} \approx 5.1 \cdot 10^{13}.$$

Numbers of new states generated during each period are depicted in Fig. B.1.1.

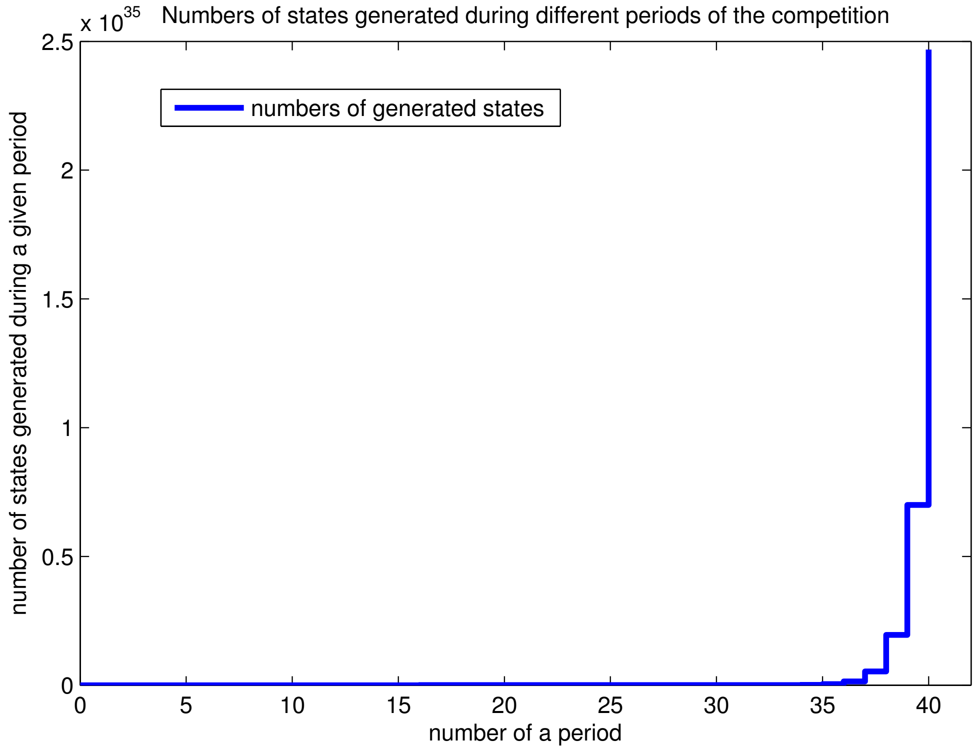

Fig. B.1.1. The numbers of new states generated during different periods of the competition

It can be seen that the growth in Fig. B.1.1 is approximately exponential. Therefore, if the initial inventory level is large and sales horizon is short, then the memory complexity of the simulation is exponential. The time complexity will be even higher, as in every generated state all allowed prices would be applied in order to generate states for the next period.

Note that a number of states generated in the simulation was examined, not a set of possible values of the state variable. This set might be smaller than a total number of all states generated in the numerical simulation. It might be possible to express a natural multiple of some allowed demand value $s \cdot D_i$, $s \in \mathbb{N}, s \leq \mathtt{maximum\ number\ of\ periods}$, as a non-trivial linear combination of the rest demand values with integer non-negative coefficients, every coefficient should be not greater than `maximum number of periods`. In fact, if we consider any pair of two different non-zero demand values $D_i,\ D_j,\ i,j \in \mathbb{N} \wedge i \leq \mathtt{k} \wedge j \leq \mathtt{k}$ it is always possible to express one particular natural multiple of D_i as a particular natural multiple D_j, i.e. $\forall i,j \in \mathbb{N} \wedge i \leq \mathtt{k} \wedge j \leq \mathtt{k}\ \exists \mathrm{c}, \mathrm{d} \in \mathbb{N} \wedge \mathrm{c} \neq \mathrm{d} :\ \mathrm{c} \cdot D_i = \mathrm{d} \cdot D_j$. It is so due to the fact that different allowed values D_i are in general rational numbers, this fact follows from equations (5.2.1) and (5.1.1). Different non-zero demand values $D_i,\ D_j$. $D_i,\ D_j$ are rational numbers. Thus, $D_i = \frac{\mathrm{b}}{\mathrm{a}} \wedge D_j = \frac{\mathrm{p}}{\mathrm{q}}$, where $\mathrm{b}, \mathrm{a}, \mathrm{p}, \mathrm{q} \in \mathbb{N}$ because all estimated demand values are non-negative and $D_i,\ D_j$ are assumed to be non-zero. Therefore, $\mathrm{c} \cdot D_i = \mathrm{d} \cdot D_j$, where $\mathrm{c}, \mathrm{d} \in \mathbb{N} \wedge \mathrm{c} = \mathrm{a} \cdot \mathrm{p} \wedge \mathrm{d} = \mathrm{q} \cdot \mathrm{b}$. And as $D_i \neq D_j$, it is obvious that $\mathrm{c} \neq \mathrm{d}$. If both c and d are less than a total number of competition periods, then two possible values of the state variable would merge, i.e. $x_1 = x_2$, where $x_1 = x_{\mathtt{initial}} - \mathrm{c} \cdot D_i \wedge x_2 = x_{\mathtt{initial}} - \mathrm{d} \cdot D_j$. However, in numerical simulations of the system states described with the same values x_1 and x_2 would not merge as they are considered at different periods of the competition, i.e. at different time. Note that the worst case was discussed. If an initial inventory level is small enough, then the number of generated states will decrease. It is so due to the constraint on a state variable value, the last has to be non-negative. In general, when an amount of required memory for an exponential algorithm is reduced by n times, where n is any natural number, the complexity of the algorithm remains exponential. In our case the value of a state variable decreases, or remains the same, if new states are generated, and there is a lower bound for the value the state variable. All possible values of a state variable will be in the interval $[0, x_{\mathtt{initial}}] \subset \mathbb{Q}$. The memory complexity and the time complexity were experimentally examined for a simple problem.

The initial inventory level $x_{\mathtt{initial}} = 2$ was considered. Allowed prices were assumed to be natural numbers from an interval $[100, 200]$. Sales horizon has duration of 10 periods.

A number of required states was examined numerically using the simulation of the price competition at shopbot, see Subsection 7.2.1. In the simulation there were 20 adversarial retailers who acted according to the developed pricing strategy based on a model by Koçaş. The maximal number of arrived customers during one period of the competition was equal to 10 in the simulation. The probability of an arrival of one customer was equal to 0.5. The main parameter of each retailer by Koçaş is number of his loyal customer. Parameters for different retailers were generated at random in the range between 1 and 5. The size of the switching customer segment was assumed to be ten times larger than the size of loyal segment. Before starting the generation of states, the retailer observed prices for 50 periods of the competition. Note, that some obtained values D_i for different prices were the same: during the simulation no retailer set a price lower than 113, so probabilities of success for prices lower than 113 were equal 1. Thus it was not the worst case when all possible values of demand are different.

In the described simulation 736656 states were added into the memory. The time complexity of an algorithm for the first estimation might evaluated by counting a number of required tests on data (number of testing `if` conditions) [45]. 74402256 times it was tested if a generated value was non-negative and at the same time had not been generated before. On the author's personal computer the simulation ran approximately half an hour and the default size of a java heap space was enough. However, if instead of $x_{\mathtt{initial}} = 2$ the value $x_{\mathtt{initial}} = 3$ was used, then the default size of a java heap space was not enough and the simulation ran out of the memory. After increasing the size of a heap space the simulation ran too long. After 44 hours of running the simulation was stopped. To sum up, it is insurmountably difficult to work directly with the state representation suggested in Subsection 6.3.

Appendix C.

Continuous optimal control: conditions for optimality of the solution

Let us show necessary conditions for optimality of a solution to an optimal control problem similar to ours, see the problem formulation (6.3.1). First, a general description of the problem is stated. The state variable and the control variable are scalars. The system is time-invariant and change of the state variable does not depend on a current value. A weighted function in the performance index does not depend on a current state. Also there is not an equality constraint on a final state. And both initial time and final time are known and fixed. Let us now give a general formulation of the problem. All derivations in this appendix are based on a work ([34], chapter 3).

C.1. Optimal control problem

Our system is described with a state equation

$$\dot{x}(t) = f(u(t)), \tag{C.1.1}$$

with a state variable $x(t) \in \mathbb{R}$ and a control variable $u(t) \in \mathbb{R}$.

Let the associated performance index be

$$J = \phi(x(T)) + \int_{t_0}^{T} L(u(t))dt, \tag{C.1.2}$$

where $[t_0, T]$ is a time interval we examine; $L(u(t), x(t))$ is a weighting function; $\phi(x(T))$ is a final weighting function.

We are going to consider an optimal control problem of finding such a control function $u^*(t)$ which would maximise the performance index J. $u^*(t)$ is a function which assigns to every moment $t \in [t_0, T]$ a control variable.

C.2. Derivation of the necessary conditions

Let us present here a brief derivation of necessary conditions for optimality based on the derivation from ([34], chapter 3). Insignificant distinction between the derivations is given by differences in problem formulations.

The performance index (C.1.2) has to be maximised. However, at every moment of time $t \in [t_0, T]$ Eq. (C.1.1) must hold. Thus, for every t we should adjoin the Eq. (C.1.1) to the performance index using a Lagrange multiplier. Alternatively, it can be done for the whole time interval of interest if we define an associated function $\lambda(t) \in \mathbb{R}$. $\lambda(t)$ is a function of time and is defined for $\forall t \in [t_0, T]$. $\lambda(t)$ is called a *costate variable*, or simply a costate. Thus, the augmented performance index is

$$J' = \phi(x(T)) + \int_{t_0}^{T} \left[L(u(t)) + \lambda(t)(f(u(t)) - \dot{x}(t))\right] dt. \tag{C.2.1}$$

Let us define a Hamiltonian function as

$$H(u(t)) = L(u(t)) + \lambda(t) f(u(t)). \tag{C.2.2}$$

Now the augmented performance index (C.2.1) can be rewritten as

$$J' = \phi(x(T)) + \int_{t_0}^{T} [H(u(t)) - \lambda(t)\dot{x}(t))] \, dt. \tag{C.2.3}$$

To find a maximum of J', it is necessary to find an expression for an infinitesimal change in J' caused by independent infinitesimal changes in all of its arguments. However, state variable $x(t)$ is a function of time. As a consequence, x and t are not independent and J' depends on both of them.

"Let us define the variation in $x(t)$, $\delta x(t)$, as the infinitesimal change in $x(t)$ when time t is held fixed" ([34], p. 111).

Let us now present here two relations from calculus of variations. The first one is

$$dx(T) = \delta x(T) + \dot{x}(T) dT, \tag{C.2.4}$$

where $dx(T)$ is an overall infinitesimal change in x when time t is fixed, $t = T$.

Another relation is called *Leibniz's rule* for functionals ([34], p. 111): *"If $x(t) \in \mathbb{R}^n$ is a function of t and*

$$J(x) = \int_{t_0}^{T} h(x(t), t) dt,$$

where $J(\cdot)$ and $h(\cdot)$ are both real scalar functionals (i.e., the functions of the function $x(t)$), then

$$dJ = h(x(T), T) dT - h(x(t_0), t_0) dt_0 + \int_{t_0}^{T} \left[\left(\frac{\partial h(x(t), t)}{\partial x} \right)^{\mathrm{T}} \delta x \right] dt." \tag{C.2.5}$$

Note that in the formula (C.2.5) $x(t)$ and $h(x(t), t)$ are slope vectors. Let us now use Leibniz's rule (C.2.5) to obtain an infinitesimal change in the augmented performance index (C.2.3):

$$\begin{aligned} dJ' = \frac{\partial \phi}{\partial x} dx \mid_T + \frac{\partial \phi}{\partial t} dt \mid_T + [H - \lambda \dot{x}] \, dt \mid_T - [H - \lambda \dot{x}] \, dt \mid_{t_0} + \\ + \int_{t_0}^{T} \left[\frac{\partial H}{\partial x} \delta x + \frac{\partial H}{\partial u} \delta u - \lambda \, \delta \dot{x} + \left(\frac{\partial H}{\partial \lambda} - \dot{x} \right) \delta \lambda \right] dt. \end{aligned} \tag{C.2.6}$$

For the sake of readability parameters of all functions are omitted.

Note that using integration by parts a term with $\delta \dot{x}$ can be eliminated:

$$\int_{t_0}^{T} [-\lambda \, \delta \dot{x}] \, dt = -\lambda \, \delta x \mid_T + \lambda \, \delta x \mid_{t_0} + \int_{t_0}^{T} \left[\dot{\lambda} \, \delta x \right] dt.$$

Let us substitute this result into (C.2.6). After it let us express $\delta x(T)$ and $\delta x(t_0)$ using the fact $\delta x(t) = dx(t) - \dot{x}(t) dt$ obtained from (C.2.4). Note that both t_0 and T are fixed and known. Also $x(t_0)$ is fixed and known. As a consequence, $dt_0 = dT = dx(t_0) = 0$. Let us take this into consideration. From the expression (C.2.4) we have that $dx(T) = \delta x(T)$ cannot be set to zero. As a result, for our case we obtain

$$dJ' = \left(\frac{\partial \phi}{\partial x} - \lambda \right) dx \mid_T + \int_{t_0}^{T} \left[\left(\frac{\partial H}{\partial x} + \dot{\lambda} \right) \delta x + \frac{\partial H}{\partial u} \delta u + \left(\frac{\partial H}{\partial \lambda} - \dot{x} \right) \delta \lambda \right] dt. \tag{C.2.7}$$

Our performance index J reaches its maximal value at the same point where the augmented performance index J' has a maximum. A maximum has to be a stationary point, it means that an infinitesimal change dJ' caused by independent infinitesimal changes in all of its arguments has to be zero. It means that in (C.2.7) coefficients of $\delta x, \delta u, \delta\lambda, dx \mid_T$ have to be zero. It gives us a set of necessary conditions for a maximum of our performance index. Let us denote by u^* an optimal control function. Let us sum up necessary conditions for optimality of a solution to our problem formulation together with the problem formulation and the definition of Hamiltonian function.

$$\begin{aligned}
& && \textbf{Model of the system:} \\
& && \forall t \in [t_0, T]: \\
& && \dot{x}(t) = f(u(t)); \\
& && J = \phi(x(T)) + \int_{t_0}^{T} L(u(t))dt; \\
&\text{problem formulation}: && u^* = \arg\max_u J; \\
&\text{optimal controller Hamiltonian}: && H(u(t)) = L(u(t)) + \lambda(t) f(u(t)). \\
& && \textbf{Necessary conditions for maximum:} \\
&\text{state equation}: && \frac{\partial H}{\partial \lambda} - \dot{x} = 0; \\
&\text{costate equation}: && \frac{\partial H}{\partial x} + \dot{\lambda} = 0; \\
&\text{stationary condition}: && \frac{\partial H}{\partial u} = 0. \\
&\text{boundary condition}: && \left(\frac{\partial \phi}{\partial x} - \lambda\right) \mid_T = 0.
\end{aligned} \tag{C.2.8}$$

C.3. Pontryagin's Maximum Principle

The stationary condition for optimality from (C.2.8) was obtained under assumption that a control variable is unconstrained. However, in many practical applications it is not the case and a control variable can take only *admissible* values from a limited set ([34], p. 232). Such a value of a control u that $\frac{\partial H}{\partial u} = 0$ may not belong to a set of admissible values. In this case a stationary condition has to be replaced with a more general one, which is called a Pontryagin's Maximal Principle: "the Hamiltonian must be maximised over all admissible controls u for optimal values of the state and costate" ([34], p. 232). All the rest necessary conditions for optimality of the solution hold. The theorem is named after a mathematician Pontryagin, who has proved it. Note that it might be called a Pontryagin's Minimum Principle, depending on a type of an optimal problem. Let us denote by U a set of admissible values of a control variable. u^* stands for an optimal control. Let us denote by x^* a corresponding optimal state trajectory and by λ^* a corresponding optimal costate trajectory. Now we can state a Pontryagin's Maximum Principle for our problem.

Theorem C.3.1 (Pontryagin's Maximum Principle)**.** *Having an optimal control problem (C.3.1)*

$$\begin{aligned} & \textbf{\textit{Model of the system:}} \\ & \forall t \in [t_0, T]: \\ & \dot{x}(t) = f(u(t)); \\ & J = \phi(x(T)) + \int_{t_0}^{T} L(u(t))dt; \\ \text{problem formulation}: \quad & u^* = \arg\max_u J. \end{aligned} \tag{C.3.1}$$

and defining a Hamiltonian as $H(u(t), \lambda(t)) = L(u(t)) + \lambda(t)f(u(t))$, *the necessary conditions for optimality of a control function are:*

$$\begin{aligned} &\text{state equation}: && \frac{\partial H}{\partial \lambda} - \dot{x} = 0; \\ &\text{costate equation}: && \frac{\partial H}{\partial x} + \dot{\lambda} = 0; \\ &\text{boundary condition}: && \left(\frac{\partial \phi}{\partial x} - \lambda\right) |_T = 0. \\ &\text{maximum principle}: && H(x^*, u^*, \lambda^*) \geq H(x^*, u, \lambda^*), \ \forall u \in U. \end{aligned} \tag{C.3.2}$$

C.4. Sufficient conditions for local maximum

We have obtained a set of necessary conditions for optimality. But without sufficient conditions for optimality necessary conditions might be misleading. For instance, consider Perron's paradox ([46], p.148): *"Let N be the largest positive integer and suppose that N ≠ 1. Then we have $N^2 > N$, which contradicts the property of N being the largest positive integer. Therefore, N = 1."* Perron's paradox highlights the danger of working only with necessary conditions. Note, that $N = 1$ is a necessary condition for optimality of the solution to a problem of finding the largest positive integer ([46], p.148).

A set of necessary conditions from Appendix C.3 would become a set of sufficient conditions for a maximum, if we ensure that the second variation of the augmented performance index $\delta^2 J' < 0$ for all variations $\delta u \neq 0$ ([34], p. 189), ([33], p. 182). As in our case the initial and the final states are fixed, we have

$$\delta^2 J' = \frac{1}{2} \int_{t_0}^{T} [\delta x \quad \delta u] \begin{bmatrix} \dfrac{\partial^2 H}{\partial^2 x} & \dfrac{\partial^2 H}{\partial x \partial u} \\ \dfrac{\partial^2 H}{\partial u \partial x} & \dfrac{\partial^2 H}{\partial^2 u} \end{bmatrix} \begin{bmatrix} \delta x \\ \delta u \end{bmatrix} dt$$

In our case Hamiltonian does not depend on a current state. As a consequence, $\dfrac{\partial^2 H}{\partial^2 x} = \dfrac{\partial^2 H}{\partial x \partial u} = \dfrac{\partial^2 H}{\partial u \partial x} = 0$. As a result,

$$\delta^2 J' = \frac{1}{2} \int_{t_0}^{T} \left(\delta u \ \frac{\partial^2 H}{\partial^2 u} \ \delta u \right) dt$$

In order for $\delta^2 J'$ to be negative for all $\delta u \neq 0$, $\delta u \, \frac{\partial^2 H}{\partial^2 u} \, \delta u$ has to be negative for all $\delta u \neq 0$. In other words, matrix $\frac{\partial^2 H}{\partial^2 u}$ has to be negative definite. Note, that a control variable in our case is just a scalar. So $\frac{\partial^2 H}{\partial^2 u}$ is a scalar as well. And it has to be negative to ensure that a solution given by (C.3.2) is an optimal solution to the problem we solve.

To sum up, the sufficient condition for optimality in our case is

$$\frac{\partial^2 H}{\partial^2 u} < 0.$$

Appendix D.

The attached CD contents

A computational simulation of price competition at shopbots and implementations of strategies obtained in the work. For the purpose of easier usage of the simulation, an archive of a NetBeans project *Simulation.zip* is provided. Source codes can be found in a directory *src* in the archive.

Results of all carried out numerical experiments for evaluation the strategies' quality. All results can be found in archives *Experiments_length of simulation.* All possible settings of numerical experiments are sorted into an intuitive system of directories. For instance, all results for a case when a tested retailer was less popular than his adversary, had a smaller capacity and demand was low can be found in the directory *../less popular than an adversary/smaller capacity than an adversary/demand is low.* There is a pdf file with a table of results in a directory of the considered case. Also there is a directory *graphs* in it. For every retailer there are several graphs. In a directory *profits* trajectories of the retailers profits together with adversaries' profits can be found. The graphs depict an average value of a retailer's profit based on repetitions of the same head-to-head competition. Standard deviations are depicted as well. Graphs of average prices and average inventory levels have a similar form.

Electronic version of the thesis in PDF.

References

[1] W. W. Hill and S. E. Beatty. “A model of adolescents’ online consumer self-efficacy (OCSE)”. In: *Journal of Business Research* 64 (2011), pp. 1025–1033.

[2] Český statistický úřad. *Využití internetu v maloobchodě.* Jan. 21, 2013. URL: http://www.czso.cz/csu/csu.nsf/91a943a39e1b11c6c1257236004a6f1d/9670c3fb22079be4c1257af70$FILE/c012113analyza.pdf (visited on 12/26/2013).

[3] H. Schneider and A. Sönke. *Retailer Competition in Shopbots.* 2007. URL: http://ssrn.com/abstract=1078505 (visited on 09/26/2013).

[4] Y. Wan, S. Menon, and A. Ramaprasad. “A classification of product comparison agents”. In: *Communications of the ACM* 50.8 (2007), pp. 65–71.

[5] C. Koçaş. “A Model of Internet Pricing under Price-Comparison Shopping”. In: *International Journal of Electronic Commerce* 10.1 (2005), pp. 111–134.

[6] G. Iyer and A. Pazgal. “Internet Shopping Agents: Virtual Co-Location and Competition”. In: *Management Science* 22.1 (2003), pp. 85–106.

[7] E. Brynjolfsson and M. D. Smith. “Consumer Decision-Making at an Internet Shopbot: Brand Still Matters”. In: *The Journal of Industrial Economics* 49.4 (2001), pp. 541–558.

[8] M. D. Smith. “The impact of shopbots on electronic markets”. In: *Journal of the Academy of Marketing Science* 30.4 (2002), pp. 442–450.

[9] Allegro Group CZ s.r.o. *Nápověda nákupní galerie HEUREKA.CZ: Ověřeno zákazníky.* 2014. URL: http://sluzby.heureka.cz/napoveda/certifikat-spokojenosti/ (visited on 04/26/2014).

[10] Allegro Group CZ s.r.o. *PODMÍNKY UŽÍVÁNÍ NÁKUPNÍ GALERIE HEUREKA.CZ.* URL: http://www.heureka.cz/direct/dokumenty/podminky-uzivani-nakupni-galerie-heureka-cz.pdf (visited on 03/16/2014).

[11] Allegro Group CZ s.r.o. *Nápověda nákupní galerie HEUREKA.CZ: Podpora Heureka služeb.* 2014. URL: http://sluzby.heureka.cz/napoveda/podpora-heureka-sluzeb/ (visited on 04/26/2014).

[12] T. Nguyen. *The Comparison Shopping Report Q4 2013: Top 10 CSEs.* URL: http://cpcstrategy.com/comparison-shopping/q4-2013/#7 (visited on 03/09/2014).

[13] Help Google Merchant Center. *Provide high-quality data.* URL: https://support.google.com/merchants/answer/188489 (visited on 03/16/2014).

[14] Allegro Group CZ s.r.o. *Nápověda nákupní galerie HEUREKA.CZ: Dostupnost produktu.* 2014. URL: http://sluzby.heureka.cz/napoveda/dostupnost-produktu/ (visited on 04/26/2014).

[15] K. W. Sadeddin, A. Serenko, and J. Hayes. “Online shopping bots for electronic commerce: the comparison of functionality and performance”. In: *International Journal of Electronic Business* 5.6 (2007), pp. 576–589.

[16] Y. Wan, S. Menon, and A. Ramaprasad. “The paradoxical nature of electronic decision aids on comparison-shopping”. In: *Journal of theoretical and applied electronic commerce research* 4.3 (2009), pp. 80–96.

[17] S. Larribeau and T. Penard. *Strategies and Price Competition on the Internet.* 2005. URL: http://ssrn.com/abstract=625602 (visited on 10/14/2013).

[18] M. D. Smith and E. Brynjolfsson. "Frictionless commerce? A comparison of Internet and conventional retailers". In: *Management Science* 46.4 (2000), pp. 563–585.

[19] V. Martínez-de-Albéniz and K. Talluri. "Dynamic Price Competition with Fixed Capacities". In: *Management Science* 57.6 (2011), pp. 1078–1093.

[20] K. Y. Lin and S. Y. Sibdari. "Dynamic price competition with discrete customer choices". In: *European Journal of Operational Research* 197 (2009), pp. 969–980.

[21] G. Gallego and M. Hu. "Dynamic pricing of perishable assets under competition". In: *Management Science* (2014).

[22] G. Perakis and A. Sood. "Competitive Multi-period Pricing with Fixed Inventories". In: *Working paper. Sloan School of Management, Massachusetts Institute of Technology.* (2004).

[23] U. Dulleck et al. "Buying Online: Sequential Decision Making by Shopbot Visitors". In: *Reihe Ökonomie/Economics Series, Institut für Höhere Studien (IHS)* 225 (2008). URL: http://ssrn.com/abstract=1307529 (visited on 09/05/2013).

[24] M. R. Baye et al. *Estimating firm-level demand at a price comparison site: accounting for shoppers and the number of competitors.* 2004. URL: http://ssrn.com/abstract=655461 (visited on 03/05/2014).

[25] M. Dudey. "Dynamic edgeworth-bertrand competition". In: *The Quarterly Journal of Economics* (1992), pp. 1461–1477.

[26] D. P. Bertsekas. *Introduction to Probability: Dimitri P. Bertsekas and John N. Tsitsiklis.* Athena scientific, 2002.

[27] Y. Shoham and K. Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations.* Cambridge University Press, 2009. URL: http://www.masfoundations.org/mas.pdf (visited on 07/05/2013).

[28] Chia-Hui Chen. *Lecture Notes for MIT Course Number 14.01: Principles of Microeconomics.* 2007. URL: http://ocw.mit.edu/courses/economics/14-01-principles-of-microeconomics-fall-2007/lecture-notes/14_01_lec28.pdf.

[29] Matthew Shum. *Lecture Notes for Caltech Course Number EC 105: Industrial Organization.* 2011. URL: http://www.hss.caltech.edu/~mshum/ec105/matt3.pdf.

[30] Gruber J. *Video Lecture 17: Oligopoly II, Bertrand Price Competition. MIT Course Number 14.01: Principles of Microeconomics.* 2011. URL: http://ocw.mit.edu/courses/economics/14-01-principles-of-microeconomics-fall-2007/lecture-notes/14_01_lec28.pdf.

[31] J. Y. Bakos. "Reducing buyer search costs: Implications for electronic marketplaces". In: *Management science* 43.12 (1997), pp. 1676–1692.

[32] A Bressan. "Noncooperative differential games". In: *Milan Journal of Mathematics* 79.2 (2011), pp. 357–427.

[33] A. E. Bryson. *Applied optimal control: optimization, estimation and control.* CRC Press, 1975.

[34] F. L. Lewis, D. Vrabie, and V. L. Syrmos. *Optimal control.* John Wiley and Sons, 2012.

[35] Josef Tkadlec. *Text přednášek k předmětu A3B01MA1: Matematika 1, FEL ČVUT.* 2012. URL: http://math.feld.cvut.cz/tkadlec/ftp/vyuka/ma1.pdf.

[36] N. Kropachera and A. Tihomirov. *Modelirovanie sluchajnyh velichin.* Novgorodskij gosudarstvennyj universitet imeni Jaroslava Mudrogo, 2004. URL: `http://www.novsu.ru/file/131661` (visited on 10/21/2013).

[37] J. T. LaFrance. "Internet Shopping Agents: Virtual Co-Location and Competition". In: *Management Science* 22.1 (2003), pp. 85–106.

[38] V. M. Becerra. "Solving optimal control problems with state constraints using nonlinear programming and simulation tools". In: *IEEE Transactions on education* 47.3 (2004), pp. 377–384.

[39] J. P. How. *Constrained optimal control.* URL: `http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-323-principles-of-optimal-control-spring-2008/lecture-notes/lec9.pdf`.

[40] D. P. Bertsekas. *Dynamic programming and optimal control.* Vol. 1. Belmont, MA: Athena Scientific, 2005.

[41] Demaine E. *Video Lecture 19: Dynamic Programming I. MIT Course Number 6.006: Introduction to Algorithms.* 2011. URL: `http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/lecture-videos/lecture-19-dynamic-programming-i-fibonacci-shortest-paths/` (visited on 04/11/2014).

[42] Kyle Siegrist. *Virtual Laboratories in Probability and Statistics, University of Alabama in Huntsville.* URL: `http://www.math.uah.edu/stat/bernoulli/Introduction.html` (visited on 03/15/2014).

[43] D. Acemoglu, K. Bimpikis, and A. Ozdaglar. "Price and capacity competition". In: *Games and Economic Behavior* 66.1 (2009), pp. 1–26.

[44] D. Fudenberg and J. Tirole. *Game theory.* Cambridge, Massachusetts, 1991.

[45] Fakulta elektrotechnická ČVUT. *Hodnocení algoritmů, algoritmy řazení a třídění. Slajdy předmětu A0B36PR1-Programování 1.* URL: `https://edux.feld.cvut.cz/courses/A0B36PR1/_media/lectures/13/36pr1-13_algoritmy.pdf` (visited on 04/20/2014).

[46] D. Liberzon. *Calculus of variations and optimal control theory: a concise introduction.* Princeton University Press, 2012.