TABLE OF CONTENTS

PREFACE

This thesis was written during studying at the "Machine learning and data science" master's program at ITMO University. This study focuses on dynamic pricing in petroleum retail by distributing through its network of gas stations. The topic of this thesis was proposed during the internship in the company Gazpromneft. The motivation for this study is to become a pioneer in applying dynamic pricing to the fuel industry.

I would like to express my gratitude first of all to my parents for always believing in me. I want to thank the IT&P faculty and the guys from the Machine Learning Laboratory of ITMO University for what I managed to learn from them during my studies at the master's program. I express my appreciation to my supervisor Andrey Filchenkov for the advice and the indicated errors concerning the thesis. I also want to thank my colleagues - Vadim Abbakoumov, Anton Kiselev, Alena Kuryleva for all kind of assistance in research and work. Thanks to all of my loved ones for their support and trust.

INTRODUCTION

For plenty of industries, pricing is a key element of the business itself. Examples of such business industries are various taxi services (Uber, Yandex.Taxi, Gett), booking services (Airbnb, booking.com), e-commerce services like Amazon.

Today customers can easily compare prices and choose the product that is most beneficial for them because of the wide spreading of e-business models and the implementation of digital technologies.

Dynamic pricing can be defined as the dynamic tuning of product or service prices in demand response on the market. Each seller wants to set such price for their products so that allows obtaining maximum profit, revenue or another target reward. In other words, dynamic pricing is a flexible approach to determining prices based on a plurality of factors and events occurring in the market.

The most well-known example [7] of dynamic pricing is Uber's «surge pricing» policy. This application for taxi riding suggests prices depending not only on trip distance but also on the traffic situation, current demand, the time, weather and take into account a lot of other conditions. Uber's popularity and its demonstrated advantages are results of dynamic pricing for consumers. This showed other projects the benefits of dynamic pricing and prompted its use.

E-commerce and retail are favorable industries for implementing and using dynamic pricing. Such platforms as Walmart, Amazon, Taobao sell millions and billions of products at present. It is impossible to set prices for such a number of products manually from time to time to be competitive. Amazon has developed its automatic pricing system, which can change prices every 15 minutes. Amazon pricing strategies were studied empirically in this research [6] and significant pricing factors were formulated.

Also, irrational pricing is one of the most negative factors affecting the consumer's perception. If the lower limit of the price is naturally limited to the cost price, then there isn't such a limit for the upper price range in the general case. The fact of non-optimal pricing policy can lead to long-term losses to the company, so the issue of pricing is relevant to this day. Excessively high prices can adversely affect customer attitude

relatively provided products or services in the future. One of the most negative buyer perceptions relating to pricing is unfair pricing policies which can lead to company losses in the long-term period [25].

The authors of this study [13] argue that the assumption of constant demand does not correspond to examples from real life. Therefore, market players should change their pricing policy from a static one to such a policy that could take into account the characteristics of producer resources, the state of current demand, competitor's prices. The market is constantly changing, which affects customer demand. Dynamic pricing helps to notice and use such changes.

Using dynamic pricing, sellers can increase their revenue by selling their products at prices adapted to customer demand, market conditions and the seller's offer at the time of the transaction [11]. Since a high increase in profitability can be achieved through improved dynamic pricing, the problem of optimal dynamic pricing is of great intellectual, economic and practical interest [22].

There are several different approaches to dynamic pricing implementation as the changing of prices in a marketplace. Personalized pricing is a type of price discrimination where sellers define different customer segments and set different prices for each of them. Amazon.com [1] experimented with setting different prices for its customers, it was found that such a way has a rich potential, but there are customer rejection risks. Unlike this approach to dynamic pricing, this study considers the price changes over time in a market without any assumptions about customer segmentation. This point of view on dynamic pricing concentrates on how the seller will use fluctuations in cumulative demand over time with a finite time horizon. In this article [11], the authors refer to such type of price setting over time as dynamic pricing.

As examples of the use of dynamic pricing in the petroleum industry, we can mention German and Danish retail markets [21], where prices are of intraday seasonal. Danish fuel company OK Benzin has increased margins by 4-5% without losing market share after applying dynamic pricing for its gas stations [5]. As can be seen from the example, the relevance of research and development in this area is expressed in potentially earned profit by fuel companies.

In the Russian Federation, dynamic pricing is not yet used in this area. Gazpromneft is a company that has set the task of studying this problem because dynamic pricing is one of the most important interests of the company. Also, the company is interested in becoming a pioneer in this field.

It is also worth noting that the prices of petroleum products are frozen by agreement between the government and oil companies [32]. Thus, pricing opportunities are limited from below by fuel costs and are limited from above by this agreement. In such difficult conditions, the use of dynamic pricing becomes even more relevant.

The purpose of this study is developing an agent model with a reinforcement learning approach that will control the pricing at the gas station.

Tasks for this study:

- review of existing solutions for dynamic pricing problem;
- building an environment for modeling the demand depending on time and price for a particular gas station for a specific petroleum product;
- development of a methodology demand surface reconstruction for a specific gas station;
- implement DQN architecture as the agent model which sets prices for a particular gas station for a specific petroleum product.

This work is organized as follows:

1. Review of approaches for dynamic pricing problem;
2. Theoretical background and problem formulation of reinforcement learning;
3. The practical part which contains simulation experiments.

CHAPTER 1. REFERENCE REVIEW

1.1. APPROACHES IN THE PROBLEM OF DYNAMIC PRICING

1.1.1. Demand curve modeling

The author [22] focuses on the modeling of the demand function which can be represented as a demand curve by price because of how demand depends on changes in the price. It is crucial for an agent model building in the dynamic pricing problem, but an accurate demand curve estimation that change in the time is a hard challenge. They consider that the buyers' decision of purchase based on several factors such as the price of the product itself, positive feedback rate and competitors' prices, and the demand curve is a time-dependent function with these factors. It is also assumed that the demand function structure is unknown to the seller. Based on these aspects, they suggest a feed-forward network model that gathers the information reflecting demand from the online data and predict the dynamic demand curve on the fly. Unlike traditional methods, the proposed neural network based method takes into account the relationship between several factors, such as product price, positive feedback, competitors' prices and customers behavior based on real data, instead of to create any preliminary assumptions about the demand model.

Universal approximation theorem [9] says neural networks can approximate any continuous function with any predetermined accuracy. Therefore, this characteristic of neural networks makes the proposed method capable of extracting meaning from complex or inaccurate data and can be used to model relationships that are too sophisticated to be noticed by humans or by using computer methods. Imitation experiments reflect the potential of this approach for predicting the demand curve in a dynamic and competitive environment.

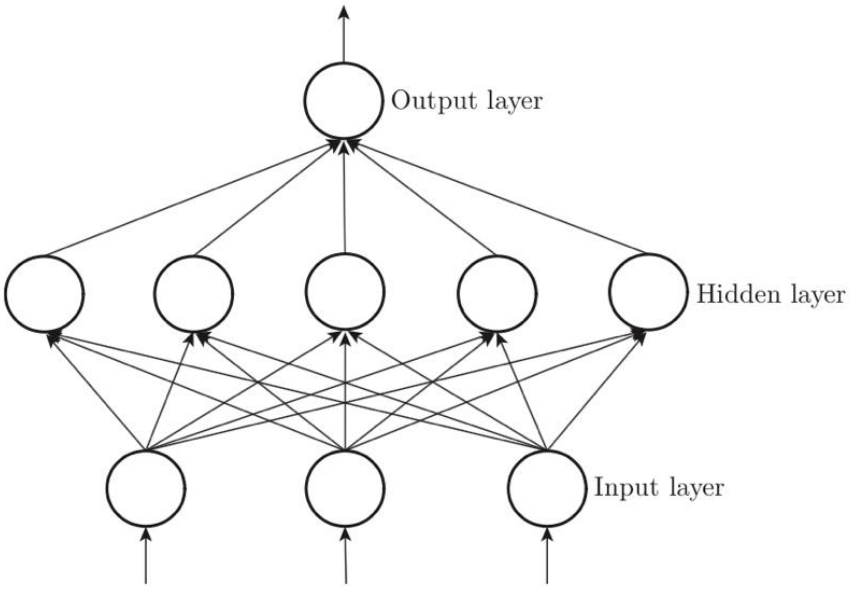

Figure 1. Neural network architecture

The three-layer feed-forward neural network (Fig. 1) with sigmoid activation function was proposed as a model with input parameters product price, feedback, competitors' prices (three units), with one hidden layer consists of five units, and customers purchase behavior as output (one unit).

1.1.2. Evolutionary algorithms

The authors have developed [29] adaptive dynamic pricing strategies and use an offline Evolutionary Algorithm (EA) to optimize their parameters although the strategies can deal with stochastic market dynamics fast enough online. They have developed two adaptive heuristic strategies of dynamic pricing in a duopoly where either firm owns a finite inventory of some type of product. They consider two scenarios. In the first, the mean of a number of customer stochastic estimation for every product is fixed across the whole selling period. In the second, the mean customer valuation for every product is changed in obedience to random Brownian motion. The parameters of both these strategies are then optimized by Evolutionary Algorithm (EA). The strategies rectify changes in the market and tune prices. Therefore, these strategies are adaptive.

From the one side, the strategies adaptive in decision making, from another side, their parameters are tuned in offline for more specific settings accordingly. Moreover, the strategies also outperform the derivative follower learning algorithm that sets the price in

the same direction (increasing or decreasing) while the revenue keeps increasing and then invert the direction of price change. These algorithms were successfully shown in the dynamic pricing problem [11, 17, 10]. The authors also compared their strategies with the Goal Directed (GD) strategy of [11]. Both strategies reach better results than GD one with the only initial price parameter, which is optimized for Brownian and non-Brownian cases.

In the final, they have shown that optimized adaptive strategies can achieve good results when different changes in the market environment on the inference phase. It means, they evaluate the strategy performance with optimized parameters for a given configuration, on a stochastically varied configuration. After that, the reached revenue is compared to the reached revenue by using the same strategy with optimized parameters for the varied configuration. It represents the regret of the wrongly estimated market structure. Demand function altering the by changing the customer/good ratio is a base of these configuration variations.

1.1.3. Kalman Filter

The authors of this article [27] suggest Kalman Filter can be a good solution to track hidden demand parameters. Its extensions promote exploration and as a result accelerate learning. Particle Swarm Optimization is a good choice for the dynamic pricing problem due to the exploratory nature of this method. When determining a corresponding balance between exploitation and exploration stages, also dynamic pricing has measurable related costs that must be taken into consideration. Since the environment dynamically changes over time, using only an exploitation strategy with time can produce worse results. On the other hand, exploration can extract valuable information about the nature of the market, which can have a positive effect on exploitation in the future. Exploration and exploitation are identical from the point of view of a customer because both concerned with setting a price. With a sufficiently strong exploration, there are risks of opportunity costs, since there is a deviation from the optimal price.

1.1.4. Reinforcement learning

The authors of this study [16] the problem of dynamic pricing in electronic retail markets with a duopoly. Multi-agent systems have some problems in learning when reward averages or discounts and only partial information about the opponent is available for the agent. To solving this problem, the authors use a combination of the performance potential idea with the simulated annealing *Q*-learning (SA-Q) and the win-or-learn-fast policy hill climbing algorithm (WoLF-PHC). The results of the simulation reflect that the WoLF-PHC algorithm outperforms the SA-Q algorithm in adapting the environment's change and in deriving better outcomes. There is a difficult tradeoff between exploration and exploitation. Excessive exploration will greatly decrease the learning performance and model convergence as a result, but superfluous exploitation will get stuck to local optimal solutions. The balance between exploration and exploitation is achieved by using the Metropolis criterion from simulated annealing algorithm in combination with the *Q*-learning.

The authors in their work [4] compare the dynamic characteristics in different price charts of varying complexity. The authors apply machine learning methods (reinforced learning) that implement a strategy that balances exploitation to maximize current profits comparing to exploration to increase future profits. The paper shows that the complexity of the price chart affects both the amount of geological exploration needed and the total profit received by the manufacturer. In general, simpler price charts are more reliable and produce less profit during training periods, even if more complex charts have higher long-term profit.

The authors of this study consider an automated market for electronic goods and they find out problems that have not been well studied previously. For example, a distinctive feature of information goods is flexibility. Marginal costs are insignificant and almost limitless bundling and unbundling of these items are possible, unlike physical goods. Therefore, producers can use complicated pricing schemes. However, the profit-maximizing design of a complex pricing schedule depends on a producer's knowledge of the distribution of consumer preferences for the available information goods. Customers' preferences are private. But there is an opportunity to recognize customer behavior pattern

through market experience. The authors of this paper compare dynamic performance across price schedules of varying complexity. They consider that the producer uses dynamic pricing strategies based on two machine learning methods: function approximation and hill-climbing. In both cases, the strategy balances the exploitation stage (maximization cumulative reward) against the exploration stage (investigation of the reward landscape to improve future rewards). They noted that the exploitation versus exploration tradeoff depends on the learning algorithms which used, and the complexity of the price profile that is offered. In common, complicated price schedules are less robust but more profitable in a long-period term. These results hold for both learning methods, though the relative performance of the methods quite depends on the choice of initial conditions and differences in the smoothness of the profit landscape for different price schedules. Their results can be applied for automated learning and strategic pricing in non-stationary environments, for example, consumer population increasing and decreasing, preferences shifting of individuals, or changes in competitors' strategies. They found out simple learning shows better results in the very uncertain markets when a pricing agent purpose is revenue maximization over a longer period than the immediate purchase period.

The authors of [24] propose a dynamic pricing algorithm for energy management in the hierarchical electricity market, which takes into account both the profit of the service provider and the costs of the client. Reinforcement learning is used to illustrate a hierarchical decision-making structure, in which the problem of dynamic pricing is formulated as a discrete final Markov decision-making process, and *Q*-learning is applied to solve this decision-making problem. Using RL, the seller can adaptively determine the retail price of electricity in the online learning process, which takes into account the uncertainty of demand profiles and the flexibility of wholesale electricity prices Simulation results show that this suggested DR algorithm, can increase SP profitability, decrease energy customers costs, find equilibrium for energy supply and demand in the market of the electricity, and enhance the electric supplying systems reliability, which can be considered for supplier and clients as a strategy where both sides are winners. Using RL for solving the dynamic pricing problem has several advantages. First, RL is a

model-free approach. It means that it does not need a specific environment model on which retail price actions can be selected. Instead, the relationship between retail price and reward can be learned by dynamic interaction with buyers. Second, RL is an adaptive approach which able to respond to a dynamically altering environment through ongoing learning and adaptation, considering the flexibility and uncertainty of the petroleum market.

In the study [25] the condition is set to maximize the price while maintaining the balance between income and equity. The authors demonstrate that RL provides two main functions to ensure equity in dynamic pricing: on the one hand, RL can learn from recent experience by adapting its pricing policy to difficult market conditions; on the other hand, it provides a compromise between short-term and long-term goals, therefore, integrating fairness into the core of the model. Given these two features, the authors propose the use of RL to optimize revenue with the additional integration of equity as part of the training process.

In pursuit of revenue maximization by dynamic pricing models, it is very important to take into account fairness and equality for avoiding unfair pricing to different customer groups. This paper shows how Reinforcement Learning (RL) techniques solve dynamic pricing problem so that prices are maximized with balancing between revenue and fairness. They show that RL has two major features to support fairness in dynamic pricing: firstly, RL allows use recent experience for learning for the pricing policy adaptation to complex market conditions; secondly, it finds out a balance between short and long-term objectives, hence integrating fairness into the model's kernel. These two features propose the RL approach for revenue optimization, with the fairness integration into the learning procedure by using Jain's index as a metric. While at the same time revenue maximization, tests in a simulated environment show a significant improvement in fairness. They have designed a synthetic environment for experiments with four different sensitivities of customers.

Cooperative learning of several adaptive agents against each other may lead to the increasingly dynamic behavior of the market and unexpected results. In this article, the authors show [18] that changes in price due to a simple price adaptation strategy. In

addition, they investigate several adaptive pricing strategies and their learning behavior in scenarios of co-learning with different competition levels. *Q*-learning is good enough at learning the best response strategies, but such learning is expensive. The authors note that the most efficient approach thought allows the agents to learn to fit their strategies using experience and observations from the past.

They developed the agent platform DMarks II for testing various models of markets and their emergent behavior. It is an agent framework with decentralized control and builds on peer-to-peer communication between its entities. Such design is well suited to model the common structure of the big future electronic markets. This framework provides agents' strategic behavior modeling from the bottom up, specifying how an agent makes its decisions, learns on its previous experience, explores the simulated market quantitatively. Various kinds of asynchronous multi-agent RL used to determine optimal seller strategies.

In this paper [8], the authors use reinforcement learning (RL) like a tool to apply dynamic pricing in an e-commerce market consisting of two competing sellers, which are sensitive to price and lead time-sensitive customers. Sellers, which provide the same products, compete on price to meet stochastically arriving customer demand and use classical inventory management. In such generalized conditions, the reinforcement learning approach has not been applied before. Sellers use automated pricing strategies (pricebots) that use RL-based algorithms to reset the prices at random intervals based on factors such as a number of back orders, levels of inventory, lead times of replenishment, and also the discounted cumulative profit maximization as an objective.

In the no information case, they show that a *Q*-learning pricing policy performs a derivative following (DF) policy. In the partial information case, they formulate the problem can be formulated in Markovian game fashion and use actor-critic architecture to solve dynamic pricing problem. They suppose their way to solve these problems is a promising approach to set prices dynamically in the situation with multi-seller markets where there is stochastic customer demand, there are customers who are price sensitive, and there is opportunity replenish sellers' inventory. Also, the authors considered a fairly general situation of an e-commerce market with two competing stores (that is, multiple

sellers). Each store consists an inventory of the same products and replenishes the inventory. Customers which sensitive to price and lead time-sensitive customers come into the system randomly.

The problem is what is the best way to determine the dynamic prices by a particular retail for revenue maximization, under the expected behavior of the buyers, the stochastic nature of the customers, inventory restrictions, and competition with the other seller. They have demonstrated how any seller can reach an equilibrium strategy in this non-cooperative, stochastic, dynamic pricing game.

1.1.5. Other approaches

In this paper [12], the authors suggest a framework for revenue optimization demand learning in association with dynamic pricing applied to monopolistic or oligopolistic firms. They have introduced a state-space model for this revenue optimization problem, which includes dynamic demand with game-theoretic nature and for nonparametric evaluating techniques the dynamic of the hidden variables of state. There are not strict model assumptions in this framework. They implemented a demand learning algorithm based on Markov chain Monte Carlo methods to evaluate parameters of the model, hidden variables of state, and functional coefficients in the nonparametric part. Such estimates allow to predict future price sensitivities and reach the pricing policy which will be optimal for the next time period.

They solved a revenue-maximizing problem on a monopoly market (one firm) and evaluated the performance of strategies of demand learning in the simulated environment. After that, they extend the framework to dynamic competition, the problem of which can be represented as a differential variational inequality. Numerical calculations demonstrate the efficiency and robustness of the demand learning approach. The demand function consists of price sensitivities which can be represented as a hidden state variable in a state-space model. One of the most important point of revenue maximization by demand learning is an estimating the pattern of the price sensitivities with high enough precision. The random walk assumption in [19] signifies that the historical price sensitivities don't provide information about its future changes, because it is supposed to the price

sensitivity is invariant in time. In other words, at time t price sensitivity equals the price sensitivity at time plus Gaussian noise. Such an assumption is totally impractical in real life, although provides ability of analytical tractability. They have made more general this assumption by learning price sensitivity from the historical sales data without any hypothesis about the parametric form of it. Usage of nonparametric functional-coefficient autoregressive (FAR) to represent the nonlinear time series of the price sensitivities increase precisely. They develop a Bayesian method based on MCMC algorithms for fitting model parameters, estimation of latent state variables, and functional coefficients jointly evaluation. Then, they used a simulated annealing algorithm for solving a monopolistic firm's pricing problem, and a fixed-point algorithm for a noncooperative competition problem. This paper suggested a demand learning strategy from an evolutionary game theory point of view and showed how to use this strategy problem in monopoly and oligopoly markets with flexible prices. FAR models are underlying in nonparametric techniques, therefore, which allows tracking the unobserved price sensitivities dynamics in model-free fashion. First, demand response to price changing is evaluated. After that that simulated annealing and fixed-point algorithm were used to reach the optimal pricing strategy in a monopoly and duopoly markets. The results of simulation showed that their provided method achieves more precisely price sensitivity estimates and predictions. This estimates and predictions are robust over a wide range of underlying state dynamics.

The authors of this paper [20] study the dynamic pricing problem on the monopolistic company that provides a perishable product or service and simultaneously learning the characteristics of customer demand. In the process of learning, the agent gathers the history of sales over a sequence of learning steps and forecast buyers demand by using an aggregating algorithm (AA) to a set of online stochastic predictors. Numerical implementation based on finite-sample distribution approximations which are periodically updated on the most recent sales data. After these are changed with a random step corresponding to the stochastic predictors. A simulation-based procedure integrated with AA optimizes the company's pricing policy. This paper proposes a general methodology because it is independent of any distributional assumptions. They show this

procedure on a demand model for a market in which customers know about dynamic pricing, are able to schedule their purchases strategically, and participate in a competition for a limited supply of a specific product. They formulate this demand model as a game-theoretic model of the buyer's choice and investigate the properties of its structure. Numerical experiments demonstrate the robustness of the learning procedure to deviations of the real market from the modeled market used in learning. The authors presented a solution to a dynamic pricing problem based on the new and robust procedure for the learning of customer demand characteristics. They illustrated that the suggested learning approach is robust in a statistical sense. The performance of the proposed method is next to the performance of optimal dynamic pricing policy for the exactly known demand model despite the use of an approximate model of consumer behavior in learning.

1.2. DYNAMIC PRICING IN THE PETROLEUM INDUSTRY

1.2.1. Example of Germany petroleum market

This article [21] analyzes the seasonal behavior of pricing at gas stations in the German retail gasoline market. You can observe high-frequency price cycles, as gas stations consistently lower prices for each other during the day, after which the price rises sharply in the evening. These asymmetric price cycles are compared to theoretical Edgeworth cycles, with the result that some differences and inconsistencies are revealed. The results of the empirical analysis suggest a strategy of intertemporal price discrimination between different types of consumers. Gas stations consistently cut each other throughout the day to attract consumers with price elastic demand. However, this phase of disruption stops at the same time increasing prices in order to use inflexible and inelastic at the price of consumers.

The dataset used includes prices for approximately 14,700 petrol stations in Germany from October 2013 to June 2015. The dataset contains all the changes in the reporting prices of gas stations for Super E5, E10 and Diesel fuels, as well as additional information about the brand, name, station address, and GPS coordinates. These data are complemented by the price of crude oil (Brent). The data was obtained from the German

Department of Fuel Market Transparency (MTU) and provided by the price comparison site «Spritpreismonitor».

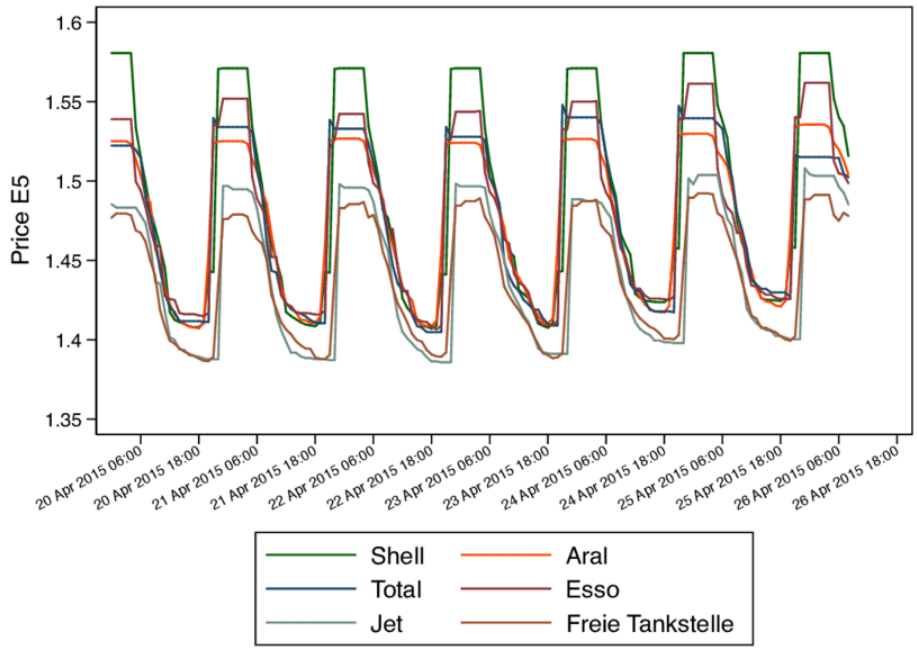

Figure 2. The price of fuel type E5 for Shell, Aral, Total, Esso, Jet and the so-called Freie Tankstellen (independent stations) gas stations in Hamburg for one week (from Monday to Sunday) in April 2015.

Shell, Aral, Total, Esso, and Jet are five major players in the German retail gasoline market that operate a nationwide network of petrol stations. Stations summarized under the term Freie Tankstelle are independently owned petrol stations that operate only at the regional level.

Repetitive daily price cycles can be observed for all brands. Prices drop during the day to six or seven in the evening. where all stations seem to increase their prices. This pricing behavior can be observed throughout the retail gasoline market in Germany and is not limited to the example presented in this analysis.

1.2.2. Solutions for dynamic pricing in the petroleum industry

In fact, dynamic pricing in the fuel industry is already in use, there are several companies provide such services, for example, a2i. Such solutions allow to increase in the prices of petroleum products during peak demand periods and cut prices in low demand periods. In the case of high demand with an optimal price rising, the margins (profits) growth will be greater than the lost profit due to the fact that some customers

reject to buy fuel because of the occurred price rising. In the low demand case with an optimal price reduction, the number of customers who purchase fuel at gas stations will increase, which will have a greater impact on the generated profit than the reduction in marginality by 1 liter of the petroleum product due to price cutting. The a2i company claims that PriceCast Fuel, its dynamic pricing solution, can increase the profitability of retailers of petroleum products by about 5 percent [28].

With PriceCast Fuel for managing the setting of the price for each product at each station within the regional corporate constraints, all gas station manager needs to do is to set the right local strategy that will give the best possible results in accordance with the regional budgets and goals. The local strategy may vary greatly from station to station. At some stations, the strategy may be to simply work with simple rules and fix the price to strictly follow the competitor with a specific price gap. At other stations, the strategy may be to optimize the price within the valid price range, for example, 2 euro cents relative to a competitor. However, for the third group of stations, the strategy may be to be a price leader and set the optimal price — with the desired balance between volume and margin — while remaining within the corporate limits.

With the help of artificial intelligence, PriceCast Fuel detects behavioral patterns in all available data related to sales and customer and competitor reactions with the frequency and accuracy that users of traditional pricing systems can only dream of. Obviously, knowledge of customer behavior and how and when they react to even the slightest changes, allows you to optimize your business. PriceCast Fuel discovers details that ultimately allow petroleum retailers to earn more by applying artificial intelligence to your historical transactions and real-time transactions [5].

One of the well-known examples [5] of industrial dynamic pricing implementation is the collaboration of one of the leading Danish fuel companies, OK Benzin, with a2i. The results of this data analysis were overwhelming, and the pilot experiment must confirm their promises. During tests of this experiment, it turned out that using PriceCast Fuel OK Benzin can increase margins by 4-5% without affecting its market share. After successfully completing the pilot test at the end of 2011, OK Benzin began a full deployment of PriceCast Fuel to manage prices at all its 700 stations.

1.3. PROBLEM STATEMENT

In our case, we want to use dynamic pricing to control the price of a specific petroleum product at a particular gas station in order to maximize profit. This gas station exists in the local market, where there are a local competitor and customers that form demand.

Purposes for dynamic pricing of petroleum products may be the following:

- profit maximization;
- revenue or sales maximization (concerned with the influence or market share expansion);
- redistribution of customer flows.

In this case, it is means use more attractable prices (with some discount) on the unpopular gas stations. It will increase revenue from such stations and make such stations more competitive on the local market. Usage a higher price on the gas stations with high demand will lead to mild demand decreasing and as a consequence of queues' reduction.

It is proposed to solve the dynamic pricing problem by reinforcement learning methods, which will be discussed below.

CONCLUSIONS ON CHAPTER 1

The result of Chapter 1 of this work is a review of the literature related to the dynamic pricing problem and its solutions. Also, this task was considered applying to the fuel industry and an example of the vendor's solution is given. And finally, the problem statement of the dynamic pricing is formulated intuitive and economics scene.

CHAPTER 2. THEORETICAL BACKGROUND

2.1. MARKOV DECISION PROCESS

The Markov Decision Process (MDP) is a process of consistent decision making under uncertainty for a fully observable environment with a Markov transition model and additional rewards. MDP is a useful framework for studying optimization problems solved using dynamic programming and reinforcement learning. MDP was known at least back in the 1950s [2]. The main study of decision-making processes according to Markov was given in [15].

The idea is that at any given time the process is in a certain state, and the agent of the system (the decision maker) can choose any action available in the current state. The system responds to the next point in time by transferring to a new state s', giving the agent some reward $\mathrm{Q_a(s, s')}$. The probability that the system enters a new state s', depends on the action chosen by the agent; this is due to the transition function $\mathrm{P_a(s, s')}$. Thus, the new state s' depends on the current state of the system s и and the action a chosen by the agent, but it does not depend on previous states and actions, in other words, the transitions satisfy the Markov property.

The difference between Markov decision-making processes and Markov chains is that in the first case, there are actions (determined by the agent choice) and rewards (providing motivation by the environment). If for each state of the system there is only one action and all rewards are the same, Markov's decision-making process is reduced to the Markov chain.

Mathematically, the Markov decision-making process is represented by four tuples:

- S – the finite states space of the environment;
- A – the finite action space, where A_s – the set of possible actions for the state S;
- $P_a(s, s') = \mathrm{P}(s_{t+1} = s' | s_t = \mathrm{s}, a_t = a)$ – the probability that the action a n the state s in the moment of time t will lead to the state s' in the next moment of time $t + 1$;
- $Q_a(s, s')$ – the expected immediate reward received after conversion from the state s to the state s' due to the action a.

The theory of Markov decision-making processes does not assert that A or S is finite, but often the basic algorithms using the MDP framework assume that they are finite.

Thus, the agent at each step receives the following tuple (s, a, r, s'), $r = Q_a(s, s')$.

In our problem formulation, the agent is a gas station, the system or environment with which the agent interacts is the local petroleum retail market, which generates the demand for gasoline products.

Many states reflect the current situation in the local market. An action space of the agent - a bunch of prices which sets, in the beginning, every specified time period (once in 1 day, 8 hours, 4 hours, 1 hour). The reward function is a profit of the gas station per period. A possible additional context that can be added: competitor prices, geo-location, physical parameters of the gas station (store availability, car wash, number of pumps), external parameters (traffic on the road, weather).

2.2. REINFORCEMENT LEARNING

Due to the unknown structure of the environment and the behavior of agents in the market, the authors considered to be correct to use reinforcement learning (RL) in modeling. The term RL implies one of the methods of machine learning in which the agent interacts with a certain environment. The RL does not request the specific information about the environment but uses an «amplifying signal» from it, which indicates whether the agent action was in the «right» direction or not. RL procedures were recognized as powerful and practical methods for solving MDP.

The problem of dynamic pricing comes to a problem of maximizing profit from the gas station. Using the terminology of RL the agent must earn as much reward as possible [30, 34]. This is the optimization problem with the objective function depending on the time horizon of the agent's activity. If the agent is given a certain time horizon h (a model with a finite horizon) to achieve the goal, then the agent's gain is represented as (1):

where r_t is the agent's reward at time t. If the time range is not given to the agent and in the context of the task it is assumed to maximize the reward for all the upcoming

time (model with an infinite horizon), then the discounted reward can be represented the following way (2):

$$R = \mathbb{E}[r_0 + r_1 + \cdots + r_h] = \mathbb{E}\left[\sum_{t=0}^{h} r_t\right], \quad (1)$$

where r_t is the agent's reward at time t. If the time range is not given to the agent and in the context of the task it is assumed to maximize the reward for all the upcoming time (model with an infinite horizon), then the discounted reward can be represented the following way (2):

$$R = \mathbb{E}[r_0 + \gamma \cdot r_1 + \gamma^2 \cdot r_2 + \cdots + \gamma^k \cdot r_k + \cdots] = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \cdot r_t], \quad (2)$$

here γ is a discount factor, which represents a fact that the reward value decreases over time.

Thus, in our task, at each step the agent tries to maximize the discounted R_t reward at time t, interacting with the environment (3):

$$R_t = \sum_{t'=t}^{T} \gamma^{t'-t} R_t \rightarrow \max \quad (3)$$

Besides the reward function R, it is important to determine the function of the expected total reward, which can be gained starting from a current state s with a certain action a. For MDP this can be formalized in the following form (4):

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}[R_t | s_t = s, a_t = a] = \mathbb{E}_{\pi}\left[\sum_{t'=t}^{T} \gamma^{t'-t} R_t \,\middle|\, s_t = s, a_t = a\right], \quad (4)$$

where π is a policy, a function that returns the probability distribution over the set of actions A for a given current state s.

Thus, the Q-function ($Q: S \times A \rightarrow R$) is a function that reflects the expected total reward if the agent is in the state s performs the action a.

In the current formulation of the problem, the Q-function is what needs to be evaluated, because knowing it the agent in the current state s can choose such an action a that maximizes $Q(s,a)$.

The optimum of an action-value function $Q^*(s,a)$ is the maximum expected reward for all $a \in A, s \in S$ (5):

$$Q^*(s,a) = \max_{\pi} \mathbb{E}[R_t | s_t = s, a_t = a, \pi] \quad (5)$$

The optimal action-value function conforms an important identity known as the Bellman equation (6):

$$Q^*(s,a) = \max_{\pi} \mathbb{E}_{s' \sim \epsilon}[r + \gamma \max_{a'} Q^*(s',a') \,|s,a] \quad (6)$$

In other words, if we know the optimal value of the action-value of the function $Q^*(s',a')$ at the next step s' for all possible actions a', then the optimal strategy is to choose the action a′ maximizing the expectation $r + \gamma \cdot Q^*(s',a')$. The basic idea establishing many RL algorithms is to estimate the action-value of a function, represented as the Bellman equation, which is updated iteratively, $Q_{i+1}(s,a) = \max_{\pi} \mathbb{E}[r + \gamma \max_{a'} Q_i(s',a') \,|s,a]$. Such algorithms converge to the optimal action-value function $Q_i \to Q^*$ as $i \to \infty$. In practice, this basic approach is absolutely unrealizable.

2.3. TEMPORAL-DIFFERENCE LEARNING

In fact, the goal of RL is not to evaluate the *Q*-function, but to determine the optimal policy for the behavior of the agent π^*.

Many modern approaches to learning are based on the principle of TD-learning [30, 34], which involves state learning based on the estimates for subsequent states: when making the next transition for the state-action pair from which we left, the value of the *Q*-function approximates the value of the *Q*-function for a pair of state-action, in which we have come. Nowadays, in practice, TD-learning turns out to be faster and more efficient than other strategies.

The article uses off-policy TD-learning *Q*-function, which is also called the *Q*-learning method, assuming the Bellman equation solved for the maximum (7):

$$Q(s_t,a_t) \coloneqq Q(s_t,a_t) + \alpha\left(r_{t+1} + \gamma \cdot \max_{\mathrm{a}} Q(s_{t+1},a) - Q(s_t,a_t)\right). \quad (7)$$

2.4. *Q*-LEARNING

The work uses the *Q*-learning method [31, 30, 34], which refers to the experiments of the type of training with reinforcement. *Q*-learning (Watkins, 1989) is a method that allows environmental agents to learn how to act optimally in situations that can be

represented as a Markov decision process. The method is incremental for dynamic programming, imposing limited computational requirements. It works by consistently improving its quality estimates. On the basis of the rewards that the agent receives from the environment, the utility function Q is formed, which further provides the agent with an opportunity not to randomly choose a strategy of behavior, but to take into account the experience of the previous interaction with the environment. One of the advantages of Q-learning is that it is able to compare the expected utility of the available actions without forming an environmental model. Used for situations that can be represented as a Markov decision-making process.

Q-learning is a model-free method that allows environment agents to learn how to act optimally in situations that can be represented as an MDP. The method is incremental for dynamic programming imposing limited computational requirements. It works incrementally improving its estimates. Given the reward that the agent receives from the environment, the function Q is formed, which allows the agent not to choose a behavior strategy randomly, but to analyze the experience of the previous interaction with the environment. One of the advantages of Q-learning is that it is able to compare the expected reward of the available actions without having a model of the environment. This method is used for situations that can be represented as an MDP. The optimal policy is represented by the formula (8):

$$\pi^*(\mathrm{s}) = \arg\max_{\mathrm{a}} Q^*(s, a). \quad (8)$$

Sutton, Barto present [30] the Q-learning algorithm for finding a policy that is close to optimal.

Often in practice, as in our case, the number of state-action pairs (s, a) is huge (if not infinite), so it is not possible to train the function Q^* on all possible inputs (s, a). In this case, the function $Q(s, a)$ can be represented as a parametric machine learning model $Q(s, a; \theta)$ which accepts features of s and a as an input. In this formulation $Q(s, a; \theta)$ is a function of the features of s and a which returns a real number (agent's expected reward). The parameters θ of this function can be estimated using machine learning models. According to the TD-learning principle, every transition $(s_t, a_t, r_{t+1}, s_{t+1})$ in the

input for learning the model, then each learning step looks like this: the agent performs an action a from a state s for the transition to the state s′, receiving the reward r and then taking one step of learning the function $Q(s, a; \theta)$ with the input (s, a) and the output $\max_{a'} Q(s, a; \theta) + r$.

Algorithm 1. Q-learning [30, 34] (off-policy TD control) for estimating $\pi \approx \pi^*$.

Hyperparametrs: learning rate $\alpha \in (0,1]$, $\epsilon > 0$.
Initialize $Q(\mathrm{s}, \mathrm{a})$, for all $s \in S$, $a \in A$, except that $Q(terminal, \cdot) = 0$.
Loop for each episode:
    Initialize S;
    Loop for each step in episode:
        Choose A from S using policy derived from Q (e.g., ϵ-greedy)
        Take action A, observe R, S'
        $Q(S, A) \leftarrow Q(S, A) + \left[R + \gamma \cdot \max_a Q(S', a) - Q(S, A)\right]$
        $S \leftarrow S'$
    Until S is terminal.

2.5. DEEP Q-NETWORK

At their study, V. Mnih et al. [26] present the ideas that formed the basis of the Deep Q-Network algorithm (DQN), and also propose the algorithm itself. The authors believe that it is appropriate to use neural networks, namely deep *Q*-networks (DQN), as a machine learning model for approximating $Q(s, a)$.

The idea of using *Q*-network is reduced to a non-linear approximation (9):

$$Q(s, a; \theta) \approx Q^*(s, a). \tag{9}$$

In practice, the approximation is performed by minimizing loss function $L_i(\theta_i)$, which is represented in the form (10) for each iteration i:

$$L_i(\theta_i) = \mathbb{E}_{s,a \sim p(\cdot)}\left[\left(y_i - Q(s, a; \theta)\right)^2\right], \tag{10}$$

where $y_i = \mathbb{E}_{s' \sim \epsilon}\left[r + \gamma \max_a Q^*(s', a'; \theta_{i-1})\right]$ is the target value for *i*-th iteration, $p(s, a)$ is the probability distribution over the states s and the actions a, called the behavior distribution.

Gradient respect to the weights (*Q*-learning algorithm) is represented as follows (11):

$$\nabla_{\theta_i} L_i(\theta_i) = \mathbb{E}_{s,a \sim p(\cdot); s' \sim \epsilon}\left[\left(r + \gamma \max_a Q^*(s', a'; \theta_{i-1}) - Q(s, a; \theta_i)\right) \nabla_{\theta_i} Q(s, a; \theta_i)\right]. \quad (11)$$

Algorithm 2. Deep *Q*-learning with experience replay [26, 34].

Initialize replay memory D to capacity N
Initialize action-value function Q with random weights
Loop for each episode
    Get state s
    Loop for each step in the episode:
        With probability ϵ select a random action a_t otherwise select $a_t = \max_a Q^*(s, a, \theta)$
        Execute action a_t in environment and observe reward r_t and state s_{t+1}
        Store transition (s_t, a_t, r_t, s_{t+1}) in D
        Sample minibatch of transitions (s_j, a_j, r_j, s_{j+1}) from D
        Set $y_j = \begin{cases} r_j, & \text{if episode terminates at step j + 1} \\ r_j + \gamma \cdot \max_{a'} Q(s_{j+1}, a'; \theta^-), & \text{otherwise} \end{cases}$
        Perform a gradient descent step on $\left(y_j - Q(s_j, a_j; \theta)\right)^2$ with respect to network parameters θ

CONCLUSIONS ON CHAPTER 2

The result of Chapter 2 of the thesis is a mathematical formulation of the dynamic pricing problem. Also proposed is a relevant solution for this problem statement – *Q*-learning as a DQN designed agent model.

CHAPTER 3. PRACTICAL PART

3.1. APPROACHES FOR MODEL'S TRAINING AND VALIDATION

3.1.1. Learning on the demonstrations

Learning and validation of historical data are provided in the following articles [14, 23]. In this study [14], the agent may access data from previous control of the system. The authors have presented an algorithm, Deep *Q*-learning from Demonstrations (DQfD), that uses small sets of observed data for the learning process acceleration even from relatively small numbers of demonstration data and is able to automatically assess the necessary ratio of demonstration data on the learning stage due to the mechanism of prioritized replay. The authors of this research [23] talk about such a learning approach in the dynamic pricing problem on the E-commerce platform. They have conducted offline and online experiments, and in both cases, they found that DQN and DDPG trained on demonstrations outperforms previous pricing policies.

There are some advantages to such an approach. Firstly, models trained on the observed real-life data. Secondly, there is a possibility for empirical evaluation of model performance by A/B -testing on the real gas station.

Also, disadvantages exist too. The amount of data may be limited to the one specific on the particular gas station. But in the opposite case, if we have more than enough data about sales, such data may contain structural changes that have occurred over a long time. Then observations from early periods of time are not homogeneous with respect to observation from late periods of time. Either, the historical price range of the specific petroleum product of the particular gas station may be too narrow. In other words, if an agent explores the tight price interval, it will affect negatively on the agent’s exploitation. And finally, the results of the empirical assessment agent's work are associated with monetary and reputational risks if the agent's pricing strategy is not relevant to customers' demand.

3.1.2. Environment simulation

When solving the problem of dynamic pricing, the using of a virtual demand environment is a frequent enough approach.

The authors [11] in the article presents the Learning Curve Simulator, a market simulator developed which helps analyzing agent strategies of pricing in markets under limited time horizons and stochastic customer demand. Current simulator based on imitation approach and promotes agent pricing strategies understanding. An analysis of different pricing strategies under varying market conditions helps a seller make use of customers demand fluctuations get more revenue and sell more units of products.

Simulated marketplaces are able to model diverse and complex scenarios, unlike theory-based solutions which are often difficult to apply to real-world markets because of the oversimplifying assumptions that typically are made in designing a theoretical model.

IBM researchers have made significant results [17] in examining customer and seller agent-driven markets using by simulation of information goods markets. They have found out some probable traps of using dynamic pricing, for example, price wars. In their analysis, they proposed four different agent pricing strategies: game theoretic, derivative following, myopically optimal (dynamic programming), and *Q*-learning. The derivative following strategy was adopted for finite markets and was analyzed in their study.

They suppose that sellers will develop an intuitive understanding of the theoretical findings by using a simulator before conducting pricing experiments. And after that use this knowledge to build a more sophisticated pricing policy. The authors propose that sellers use the Learning Curve Simulator to study complex pricing strategies.

The authors [29] have implemented a simulation software that simulates dynamic pricing strategies in different markets with an oligopolistic nature. The EA optimizes the strategy's parameters depending on the fitness criteria (the earned revenue) by interacting with this simulator as a black box. The main profit of using simulations is that we are able to apply complicated models that are too difficult for a theoretical approach.

[27] Parameterized model demand is one of the most frequent applied approaches to the problem. As the true market demand is unknown, the model's parameters are

assumed hidden; a price is set and revenue is observed, but the demanding nature at every price point is unknown in general. This model is useful to Bayesian reasoning because allow the use of the Kalman Filter for tracking its hidden parameters. The Kalman Filter is proposed a prior (and generally subjective) the hidden parameters representation, which it tunes on the observations for a relevant understanding of those parameters. This information is usually used to choose the best price for the next time period. The best price in this context means such a price which optimizes the expected revenue. It is a cyclic process of the new price setting and actual revenue observing.

The authors of this research [25] have developed a synthetic environment to experiment with four different customer sensitivities. They defined four different customer groups whose price bidding behavior modeled by logit-function. RL agents interacted with demand generated by randomly formed customer population.

One of the main advantages of using a simulated environment is the absence of monetary and reputational risks. The second reason to use it is an opportunity for training and testing multiple models. For example, in real life, there may be a lack of entities available for A/B-testing. Simulated environment solves these troubles. But at the same time using a simulated environment as a playground imposes certain limitations. If it were possible to create such a model that predicts the demand for the price range good enough, then the solution of the problem would be choosing the price that yields the highest value of the objective function. In this case, the task comes to an optimization problem with respect to the model's parameters. But demand is too complicated to be model-based. For these reasons, it was decided to build a model-free demand simulation.

Our observed data can be characterized as cumulative demand per period for the chosen product by specific price and time. It means for each time step there is the only one observed value of aggregated demand by the only one fixed price. But for our case, we want to have the opportunity to choose different prices on each time step. For these reasons, it was decided to build a model-free demand simulation based at the observed historical data.

3.2. DATA DESCRIPTION

The company provided access to anonymized data describing the aggregate hourly sales of a group of similar gas stations. In addition to the sales series, the historical prices and competitors' prices for each station in the group were available. Data obtained for a six months period.

3.3. DEMAND SURFACE RECONSTRUCTION

For agent training and validation, there has to be an environment to interact with. There are certain risks in using a real gas station as a platform for model training and testing. For this reason, it is proposed to use a demand simulation for a specific product for a particular gas station as an environment.

In this study, we propose to compute a specific demand curve for a particular product for each period of time. We have noticed that demand curves differ from each other in time and have seasonality (the time series of hourly sales for the week for 5 different gas stations is shown at Fig. 3). We assume that the demand curve depends not only on price but also on time factor. This means that demand is a function of price and time. In this case, we can represent demand as a surface (with price and time axes).

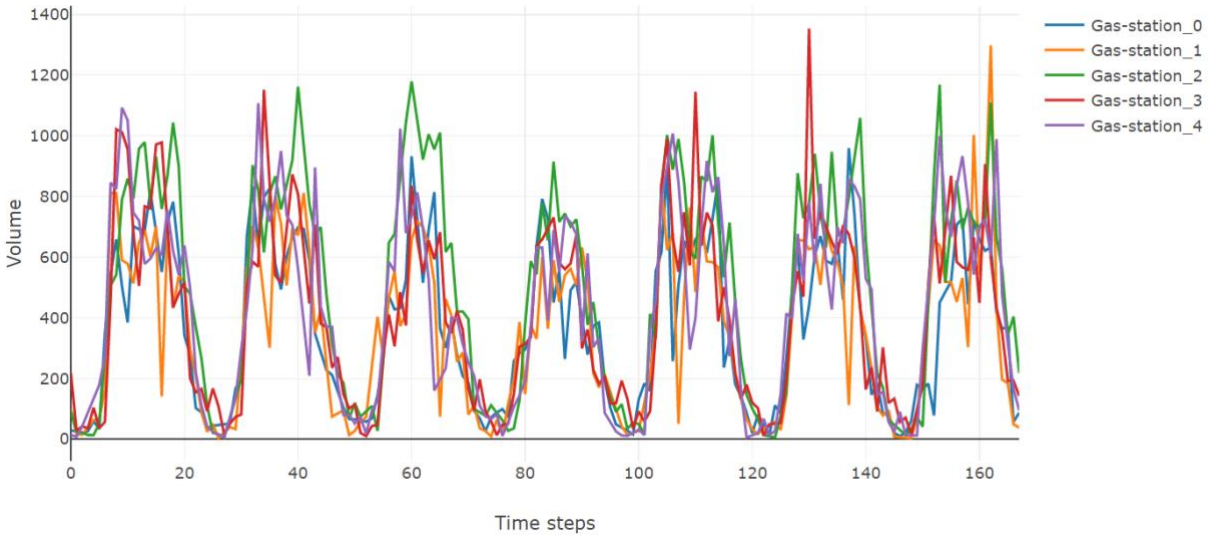

Figure 3. Time series of sales

When modeling the environment, it is important to consider that prices in each gas station change over time (Fig. 4):

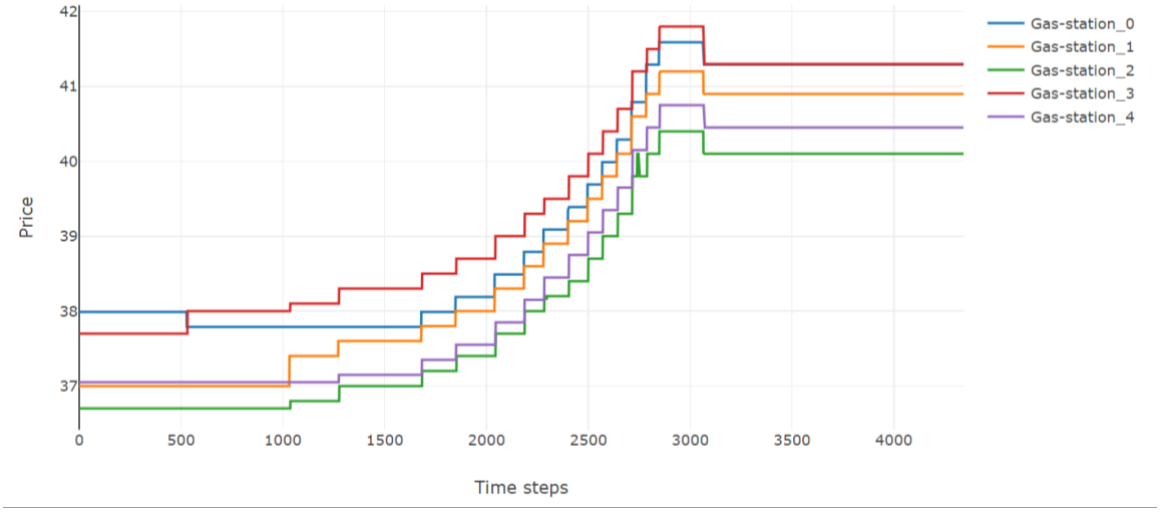

Figure 4. Price changes of five gas stations over a long period

Combining the dynamics of sales (Fig. 3) and price changes (Fig. 4), we got the sales curve in space depending on time and price (Fig. 5).

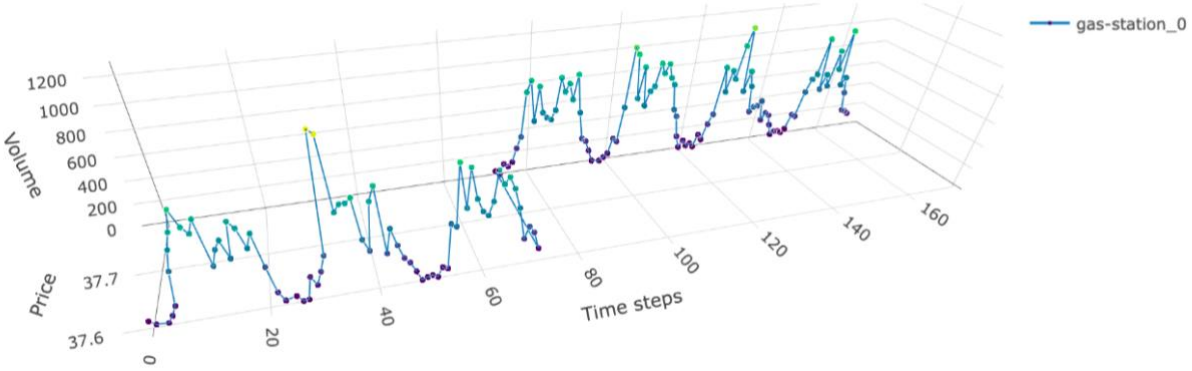

Figure 5. Sales curve in space depending on time and price

There is a problem with such demand representation – for every time step (an hour in this study) there is only one sales value corresponding to only one specific price value. This is an obstacle for the approximation of the demand function for a given hour. The approach to solving this problem proposed below.

We considered aggregated hourly demand for specific petroleum product on the particular gas station as the following function:

$$Q_d = f(p, p_c, s_h, s_d, s_{other}, t), \tag{12}$$

where

- p – price for petroleum product on our gas station;
- p_c – price for petroleum product on competitor gas station;

- s_h – seasonal hourly components;
- s_d – seasonal daily components;
- s_{other} – other seasonal (monthly) components;
- t – trend component.

Firstly, we made the transition from absolute to relative price by the formula:

$$p_r = \frac{p}{p_c} \tag{13}$$

Such price transformation helps avoid the effects of inflation, because in general for customers absolute the value of the price less important than how this price relates to the competitors' prices. For this reason, the one chosen absolute price can be perceived differently on the different local markets, which corresponds to different gas stations. It will affect the demand at a given price. But equal values of the relative price with respect to competitor on the two gas stations on the same period of time can describe similar conditions on the local markets. For example, we set equals prices for two gas stations and on the first station price is lower than the competitor's price, but on the second station, there is an opposite situation – a price higher than the competitor's one. At the first gas station demand will increase by the low price, at another gas station demand will reduce because of expensive fuel – there are different conditions in these markets. On the contrary, two same relative prices at both gas stations show that our petroleum is more expensive or cheaper than the competitor's one – there are two similar conditions in these markets. When clients make a choice in the local market, they are interested in choosing the cheapest of all alternatives acceptable for them. For this reason, the ratio of the prices is better to use than the absolute value of the price. At any time, it is possible to return to the absolute prices, knowing the competitor's price:

$$p = p_r * p_c \tag{14}$$

Then demand function can be represented as:

$$Q_d = f(p_r, s_h, s_d, s_{other}, t) \tag{15}$$

We propose to eliminate s_{other} and t components from the time series of sales. For example, linear regression is suitable for this task. After this procedure demand function depends on relative price, hour and weekday:

$$Q_d = f(p_r, s_h, s_d) \quad (16)$$

Every point in such a cleaned sales series defines by price and hour of the week (the combination of s_h and s_d) In this case we propose put this cleaned sales series into space of price axis and one week by time axis (Fig. 6). Then for every hour in a week, there are several points reflecting demand.

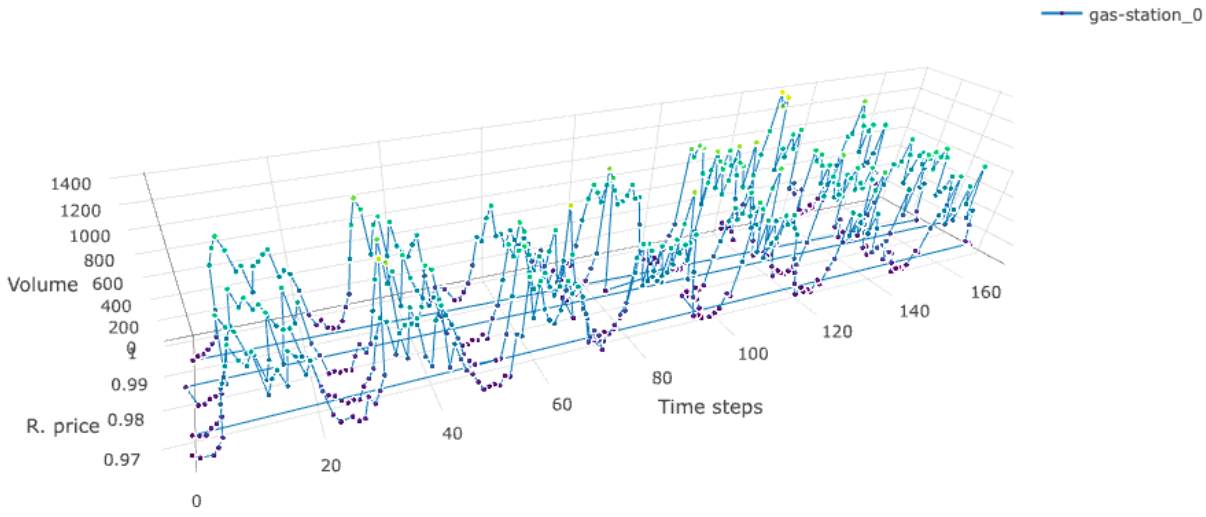

Figure 6. Demand cleaned of seasonal monthly and common trend components placed into the time and relative price space.

It is also assumed that if the time series of sales of two gas stations are similar (the task of determining similar sales series is outside the scope of this study) among themselves, it follows that the demand functions in local markets will return close demand values for all other fixed parameters. Therefore, it is proposed to consider a group of similar in sales series. In this case, each time point will correspond to several demand values corresponding to certain price value. It also allows you to increase the number of observed demand points depending on the price for each time slice.

The next step in modeling is to combine the dynamics of sales of several similar gas stations (Fig.7).

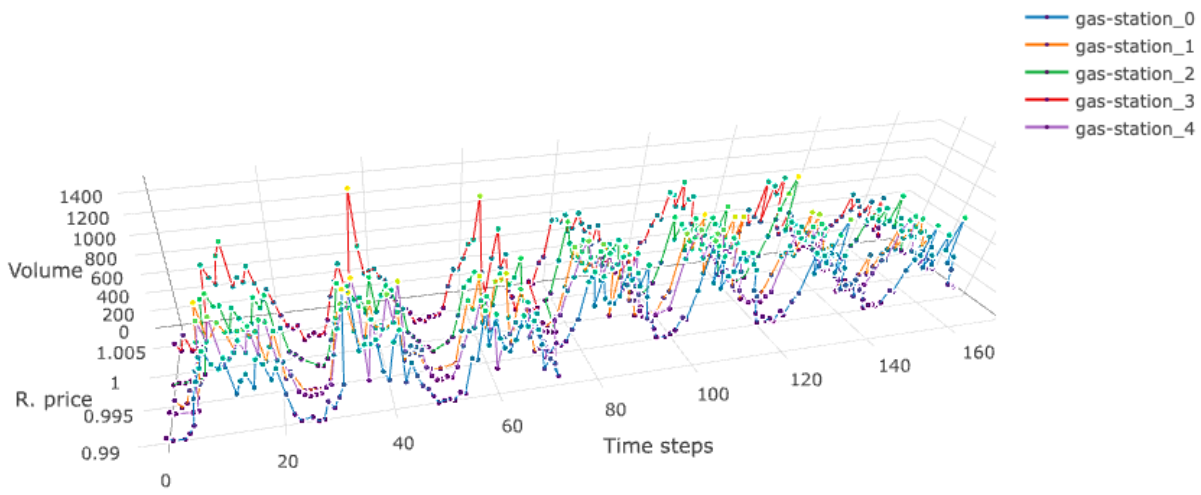

Figure 7. Sales dynamics at five gas stations

In this way, a point cloud composed of multiple weekly periods across multiple gas stations can be displayed on a one-week space (Fig.8). Each point reflects the demand for a particular petroleum product at a fixed price at each time point during the week.

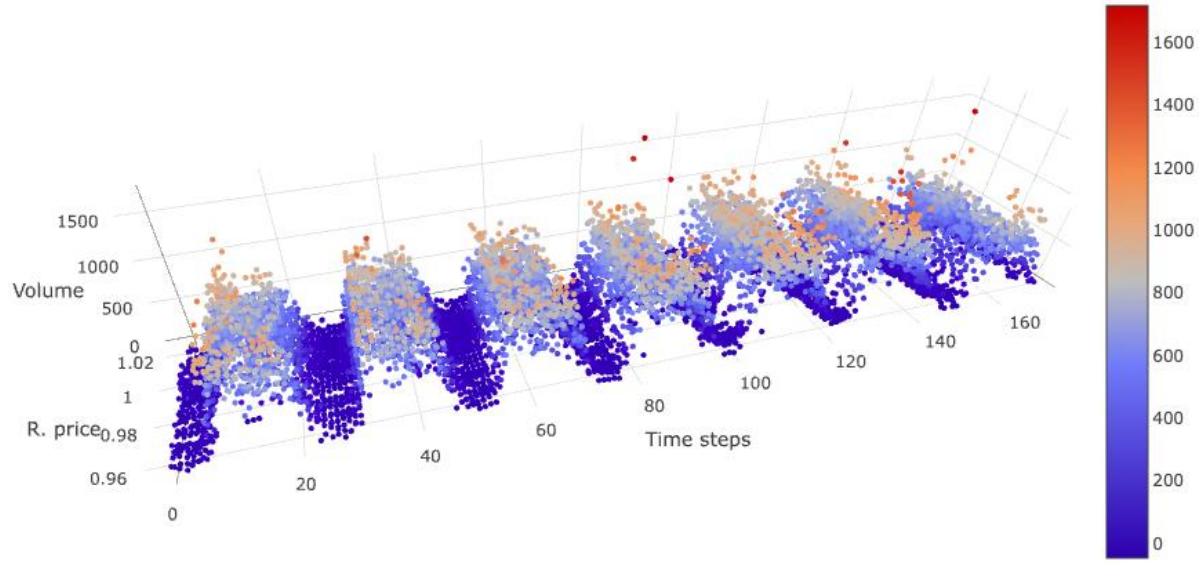

Figure 8. Weekly placement of points on the demand surface

Demand curve building is considered in classical economic theory [33]. Demand curve depending on a price describes multiple options for the volume of sales of a specific product in a particular market to the same group of buyers in a given period of time, but at different price levels. This curve is built on the assumption that if we are able to connect the points corresponding to the actual demand values at certain price levels by line. Such an approximation will describe those demand quantities that may arise in the market at all price values in the price range. Since sales are aggregated by the hour, then this point

cloud forms 168 (24 hours * 7 days) time slices. For each of these slices, a demand curve can be constructed with a predetermined functional form.

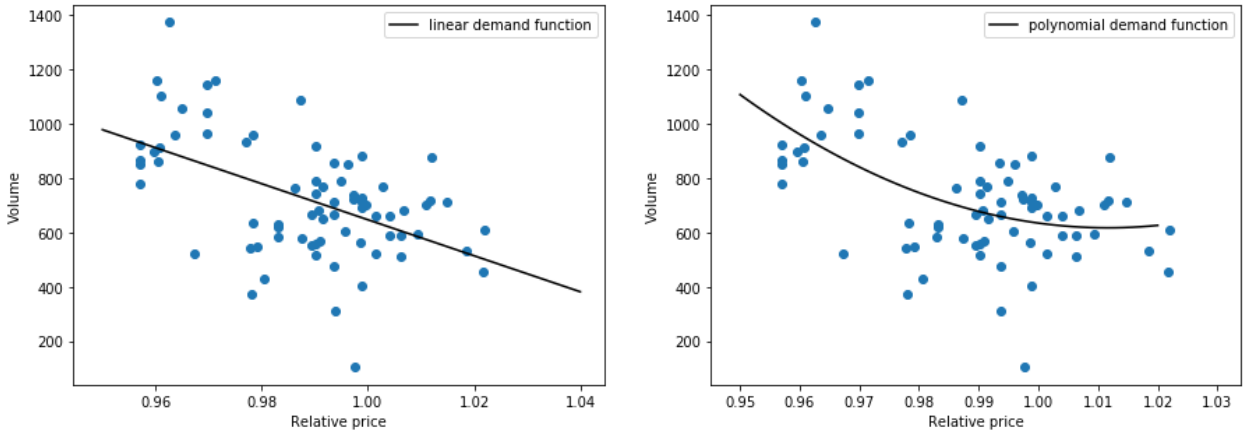

Figure 9. Demand curves for the 131 hour of a week: linear (left) and polynomial (right).

For example, observations in the 131 hour of a week (11 hours of Saturday) are displayed in Figure, 9 also plotted linear and polynomial demand curves. These curves were obtained using the ordinary least squares method at the observed demand points. Polynomial demand function formulated as a 2-degree polynomial regression. The construction of such demand functions describing implies that demand is constant. But as can be seen from the data (Fig. 8), in the context of each hour in a week, demand curves may differ from each other. Therefore, we assumed that at each moment in time there is a demand curve for petroleum products at a price, and also that the demand curves at different points in time may differ from each other.

In principle, for each hour in a week, it is possible to build a separate demand function by price. The disadvantage of this method is taking into account the observations of only one specific hour, in order to correct this deficiency, it is proposed to use surface interpolation methods. In this case, the demand function will depend on the price and also on time, and it will be not a curve but a surface on the visualization.

In this study, RBF interpolation was used, which, in contrast to linear interpolation, is resistant to outliers. Using interpolation from a cloud of points, we got the demand function which can be visualized as a surface (Fig. 10).

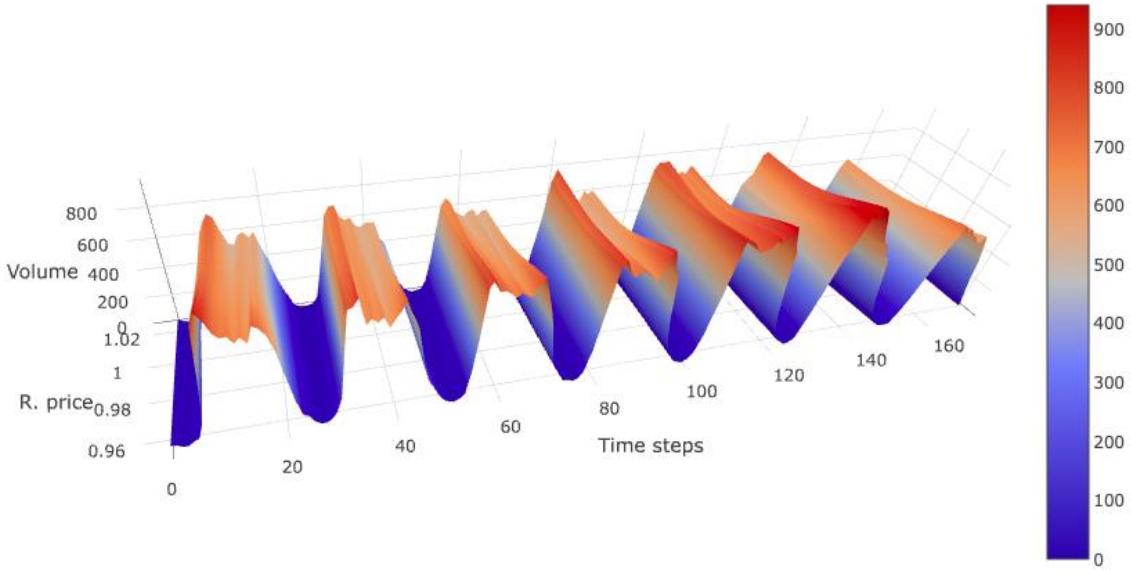

Figure 10. Demand surface RBF-interpolation.

Thus, the demand surface, which is a function of sales by time and price, can be constructed from consecutive demand curves, which is a function of sales by price. It is also important to take into account that the surface of demand has seasonality; in the context of this work, hourly and weekly seasonality is considered.

The proposed demand surface reconstruction algorithm:

1. Selection of similar time series;
2. Elimination of trend and seasonality, the period of which is more than a week;
3. "Cutting" of time series by weeks;
4. Placement of points on the surface with a length of one week (Fig. 9);
5. Interpolation of the surface of demand (Fig. 10);
6. Add trend and seasonality removed on step 2.

The pattern of behavior of the demand for petroleum products during the week and depending on the price is the result of the algorithm. Interpolation can also be performed on a certain proportion of randomly selected points in order to obtain slightly different surfaces, but with a single pattern of behavior.

3.4. OpenAI Gym

OpenAI Gym [3] was used as a framework for an agent interaction with the simulated environment. OpenAI Gym is a tool for creating environments for RL tasks. For example, in the standard build, there are Atari 2600 games [26], which nowadays have become a classical RL problem. OpenAI Gym is a state-of-the-art common interface in RL environment development. It is reasonably not written from scratch but uses ready-made agent models developed for this interface for the environment that matching all OpenAI Gym criteria.

Conclusions on Chapter 3

During Chapter 3 of the thesis, two different approaches to the training and testing of agents were considered: on a historical and on synthetic data. The pros and cons of both approaches were examined, the choice of the simulation environment was reasonable. Demand was represented as a function of price and time. Reconstructing demand surface was represented as interpolation historical demand values placed in a weekly period of time. This simulated environment was wrapped into OpenAI Gym framework for easy testing.

CHAPTER 4. EXPERIMENTS WITH THE SIMULATED ENVIRONMENT

After reconstructing the demand surface for a given period for modeling the agent's behavior, it is necessary to set the cost price of the product at each time point. For the experiment conducted in our work a 16-week-long demand surface was generated, the first 15 of which were used to train the agent, and the last one for a test.

During the learning process, the agent on each step selects a price for the next period. As a result of the chosen action, the agent receives the demand for the product for the previous step. As a reward function, it is proposed to use the profit earned by the agent for the past period according to the formula (13):

$$r = (price - cost) \cdot volume, \tag{13}$$

where $price$ is set by the agent, *cost* is the cost of the product, *volume* is the amount of product sold over the past period by the given *price*. Since we know the demand surface, the surface of rewards can be calculated (Fig. 11).

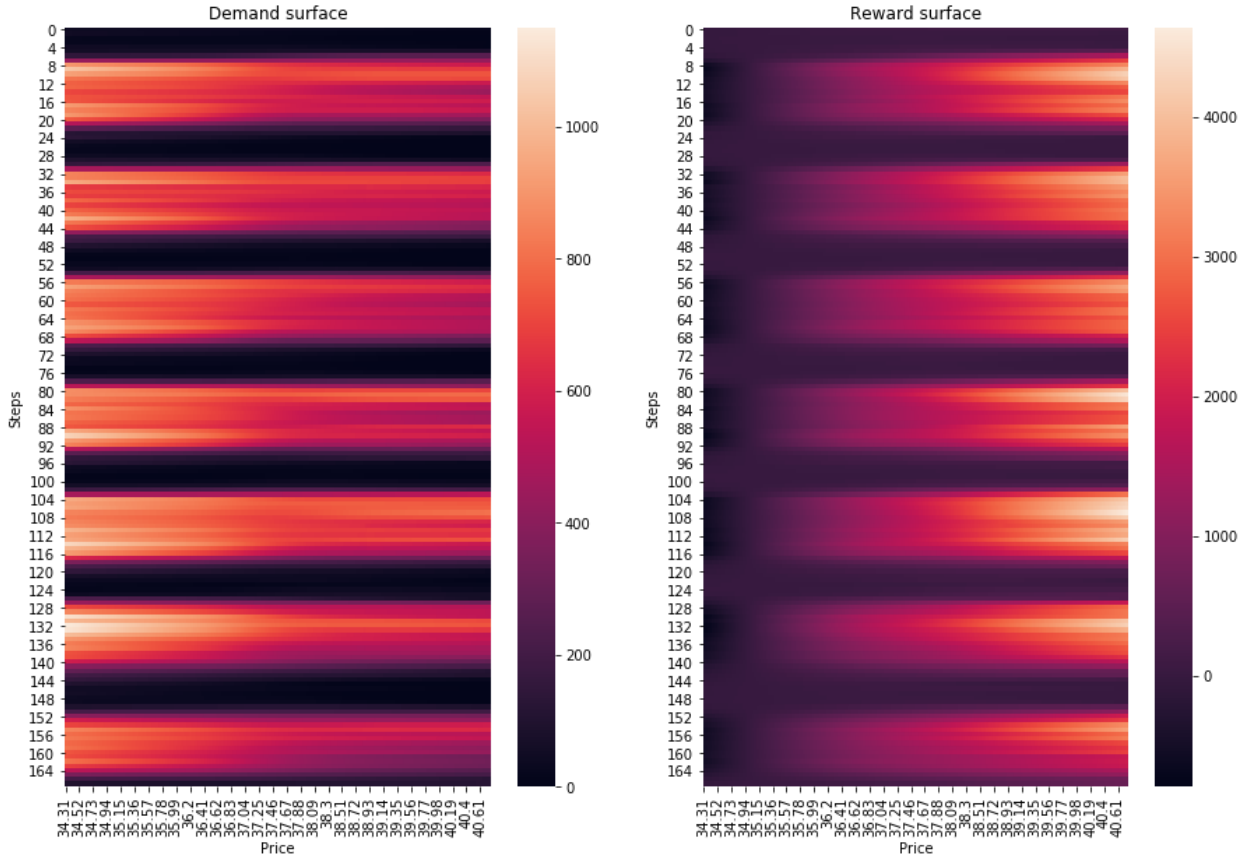

Figure 11. Heat maps of demand surface (left) and rewards (right)

At figure 11, the horizontal axis shows product prices (rubles), the vertical axis shows the time period (hour), the color shows the volume of demand for price and time.

The agent was trained at the first 15 weeks of the generated surface. After that its performance was evaluated on the last (test) week.

In our case, the agent reached the performance of 94.38% of the optimal profit value, i.e. the maximum value that can be obtained on this demand surface. The path of the agent on the surface is close to an optimal path (Fig. 12). Besides that, the agent learns seasonal patterns, such as a nightly decrease in demand, which corresponds to an intuition. This leads to the conclusion that the agent remembers the pattern of beneficial behavior (in terms of maximizing the reward function).

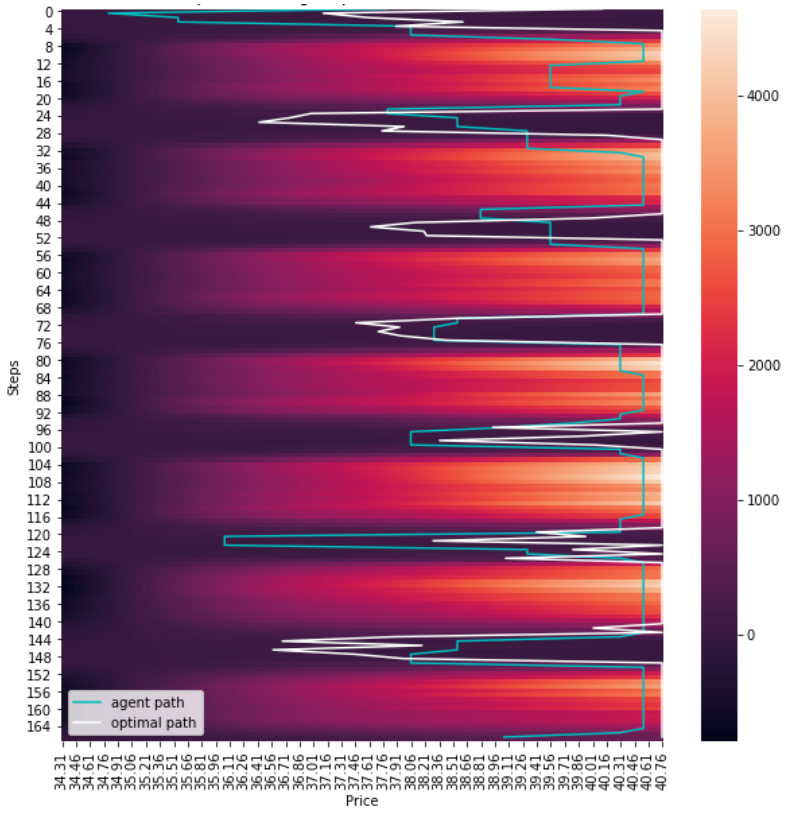

Figure 12. The price setting trajectory by the trained agent on the test surface and the optimal trajectory.

When experimenting with the training multiple agent models assembling the prices offered by them, it was possible to achieve 95.58% of the optimal profit value. It can be noticed that the average path of the agents, in this case, is smoother, and, what is more important, closer to the optimal one (Fig. 13).

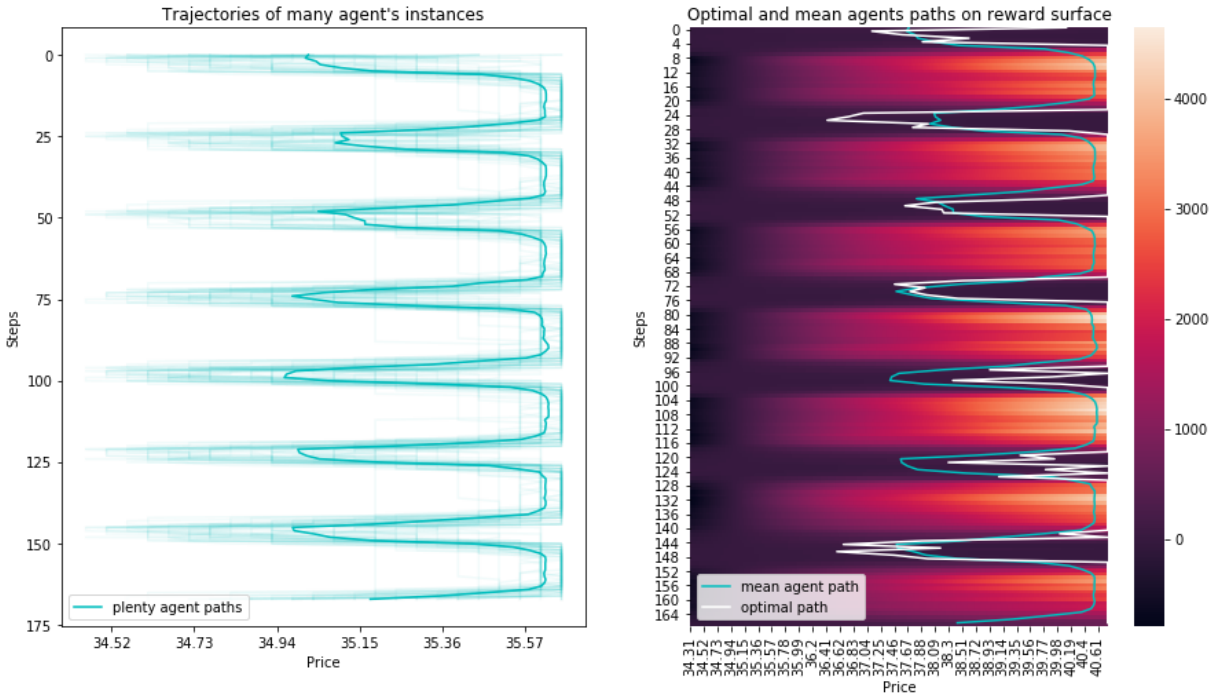

Figure 13. The price setting trajectory by fifty trained agents on the test surface and the optimal trajectory.

CONCLUSIONS ON CHAPTER 4

As a result of Chapter 4, the agent model was trained and tested within a constructed simulation environment. DQN agent model achieves 94-95% of the maximum possible total reward. DQN model learns that at night it is more beneficial to decrease the price to maximize profits.

CONCLUSIONS

The objective of this study achieved – there is a developed RL-based model for dynamic pricing at the gas station. All tasks of the thesis were solved: the reference review was described, an environment for modeling the demand was built, a methodology demand surface reconstruction was proposed, DQN agent model was implemented.

This model was trained and validated within a simulated environment. The DQN designed agent reaches 94-95% of the maximum possible total reward. As part of the MDP, an agent can be taught near-optimal pricing strategies, making decisions based only on information related to the decision time (day of the week and hour).

In the future, we are planning to expand the state space, using the prices of competitors, the gas station's geolocation, weather, and a different context. In this study, we considered the demand for only one specific product. The store has a range of products, the demand for which may also depend on the prices of substitutes from this range. In further research, we will take this into account. Also, we are planning to transfer from discrete to continuous action space by using DDPG architecture, because the price naturally looks like a continuous value. With so many elements in the state vector, some difficulties in constructing the demand surface arise. In this case, it is proposed to look towards learning from historical data, recent studies show that such training can be quite effective [14, 23].

In the near future, it is planned to launch a pilot project of dynamic price control at real gas stations. In this case, it is possible to conduct A/B testing and empirically confirm the outperforming of dynamic pricing over directive one.

REFERENCES

1. Baker W. L. et al. Getting prices right on the web //The McKinsey Quarterly. – 2001. – С. 54-54.
2. Bellman R. A Markovian decision process //Journal of Mathematics and Mechanics. – 1957. – С. 679-684.
3. Brockman G. et al. Openai gym //arXiv preprint arXiv:1606.01540. – 2016.
4. Brooks C. H. et al. Automated strategy searches in an electronic goods market: Learning and complex price schedules. – 1999.
5. Case Story: OK Benzin [Electronic resource] // A2I Systems – 2018.– Access mode: https://www.a2isystems.com/wp-content/uploads/2018/11/PriceCast-Fuel-Case-Story-15.pdf
6. Chen L., Mislove A., Wilson C. An empirical analysis of algorithmic pricing on amazon marketplace //Proceedings of the 25th international conference on world wide web. – International World Wide Web Conferences Steering Committee, 2016. – С. 1339-1349.
7. Chen M. K., Sheldon M. Dynamic Pricing in a Labor Market: Surge Pricing and Flexible Work on the Uber Platform //Ec. – 2016. – С. 455.
8. Chinthalapati V. L. R., Yadati N., Karumanchi R. Learning dynamic prices in multi-seller electronic retail markets with price-sensitive customers, stochastic demands, and inventory replenishments //IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). – 2006. – Т. 36. – №. 1. – С. 92-106.
9. Cybenko G. Approximation by superpositions of a sigmoidal function //Mathematics of control, signals, and systems. – 1989. – Т. 2. – №. 4. – С. 303-314.
10. Dasgupta P., Das R. Dynamic pricing with limited competitor information in a multi-agent economy //International Conference on Cooperative Information Systems. – Springer, Berlin, Heidelberg, 2000. – С. 299-310.
11. Dimicco J. M., Maes P., Greenwald A. Learning curve: A simulation-based approach to dynamic pricing //Electronic Commerce Research. – 2003. – Т. 3. – №. 3-4. – С. 245-276.
12. Do Chung B. et al. Demand learning and dynamic pricing under competition in a state-space framework //IEEE Transactions on Engineering Management. – 2012. – Т. 59. – №. 2. – С. 240-249.
13. Ehrenthal J. C. F., Honhon D., Van Woensel T. Demand seasonality in retail inventory management //European Journal of Operational Research. – 2014. – Т. 238. – №. 2. – С. 527-539.

14. Hester T. et al. Deep q-learning from demonstrations //Thirty-Second AAAI Conference on Artificial Intelligence. – 2018.
15. Howard R. A. Dynamic programming and markov processes. – 1960.
16. Jintian W., Lei Z. Application of reinforcement learning in dynamic pricing algorithms //2009 IEEE International Conference on Automation and Logistics. – IEEE, 2009. – С. 419-423.
17. Kephart J. O., Hanson J. E., Greenwald A. R. Dynamic pricing by software agents //Computer Networks. – 2000. – Т. 32. – №. 6. – С. 731-752.
18. Kutschinski E., Uthmann T., Polani D. Learning competitive pricing strategies by multi-agent reinforcement learning //Journal of Economic Dynamics and Control. – 2003. – Т. 27. – №. 11-12. – С. 2207-2218.
19. Kwon C. et al. Non-cooperative competition among revenue maximizing service providers with demand learning //European Journal of Operational Research. – 2009. – Т. 197. – №. 3. – С. 981-996.
20. Levina T. et al. Dynamic pricing with online learning and strategic consumers: an application of the aggregating algorithm //Operations Research. – 2009. – Т. 57. – №. 2. – С. 327-341.
21. Linder M. et al. Price cycles in the German retail gasoline market-Competition or collusion //Economics Bulletin. – 2018. – Т. 38. – №. 1. – С. 593-602.
22. Liu G., Wang H. An online sequential feed-forward network model for demand curve prediction //JOURNAL OF INFORMATION &COMPUTATIONAL SCIENCE. – 2013. – Т. 10. – №. 10. – С. 3063-3069.
23. Liu J. et al. Dynamic Pricing on E-commerce Platform with Deep Reinforcement Learning. – 2018.
24. Lu R., Hong S. H., Zhang X. A dynamic pricing demand response algorithm for smart grid: reinforcement learning approach //Applied Energy. – 2018. – Т. 220. – С. 220-230.
25. Maestre R. et al. Reinforcement Learning for Fair Dynamic Pricing //Proceedings of SAI Intelligent Systems Conference. – Springer, Cham, 2018. – С. 120-135.
26. Mnih V. et al. Human-level control through deep reinforcement learning //Nature. – 2015. – Т. 518. – №. 7540. – С. 529.
27. Mullen P. B. et al. Particle swarm optimization in dynamic pricing //2006 IEEE International Conference on Evolutionary Computation. – IEEE, 2006. – С. 1232-1239.
28. Pricecast fuel [Electronic resource] // A2I Systems – 2018.– Access mode: https://www.a2isystems.com/wp-content/uploads/2018/11/PriceCast-Fuel-Product-Folder-15.compressed.pdf
29. Ramezani S., Bosman P. A. N., La Poutré H. Adaptive strategies for dynamic pricing agents //Proceedings of the 2011 IEEE/WIC/ACM International

Conferences on Web Intelligence and Intelligent Agent Technology-Volume 02. – IEEE Computer Society, 2011. – С. 323-328.

30. Sutton R. S., Barto A. G. Reinforcement learning: An introduction. – MIT Press, 2018.
31. Watkins C. J. C. H., Dayan P. Q-learning //Machine learning. – 1992. – Т. 8. – №. 3-4. – С. 279-292.
32. Заморозка цен на бензин: грозит ли России дефицит и кто виноват [Electronic resource] // BBC News русская служба – 2018. – Access mode: https://www.bbc.com/russian/news-46221156
33. Липсиц И. В. Ценообразование. – М. : Экономистъ, 2004.
34. Николенко С. И., Кадурин А. А., Архангельская Е. О. Глубокое обучение. – " Издательский дом"" Питер""", 2017.