Czech Technical University in Prague Faculty of Electrical Engineering Department of Computer Science



MASTER'S THESIS

Inventory Optimization Based on Demand Prediction

Bc. Tomáš Vyskočil

Supervisor: Ing. Ondřej Vaněk, Ph.D.

Study Programme: Open Informatics Field of Study: Artificial Intelligence May 25, 2018

Acknowledgments

I would like to thank my supervisor Ing. Ondřej Vaněk, Ph.D. for the advice he has given me, for his patience, and his sense of humor.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne25.5.2018

Bc. Tomáš Vyskočil



ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení:VyskočilJméno: TomášOsobní číslo: 406439Fakulta/ústav:Fakulta elektrotechnickáZadávající katedra/ústav:Katedra počítačůStudijní program:Otevřená informatikaStudijní obor:Umělá inteligence

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Optimalizace skladových zásob založená na predikci poptávky

Název diplomové práce anglicky:

Inventory Optimization Based on Demand Prediction

Pokyny pro vypracování:

Many companies periodically distributing various goods from their warehouses typically face the problem of restocking their warehouses such that in any time, their goods are available for distribution, however, no unnecessary items are held in the warehouse (i.e., the space taken by the stocked goods is minimal). This well-known problem of inventory optimization can be complemented with a prediction module which predicts the demand for each stocked item. The student should explore these problems and design an approach solving the inventory optimization problem including item prediction. More specifically, the student should:

1. Understand the problem of inventory optimization and demand prediction

2. Study various algorithmic approaches to inventory optimization and demand prediction

3. Develop an algorithm for demand prediction based on historical data provided by thesis supervisor

 Integrate demand prediction with an inventory optimization model such that the model considers costs of unavailability, resupply costs as well as warehousing costs

5. Demonstrate the performance of the inventory optimization model on a set of possibly real-world scenarios

Seznam doporučené literatury:

1] Chen, Xin, and David Simchi-Levi. "Coordinating inventory control and pricing strategies with random demand and fixed ordering cost: The finite horizon case." Operations Research52.6 (2004): 887-896.

[2] Ben-Daya, M., and M. Hariga. "Integrated single vendor single buyer model with stochastic demand and variable lead time." International Journal of Production Economics 92.1 (2004): 75-80.

[3] Schmitt, Amanda J., Lawrence V. Snyder, and Zuo-Jun Max Shen. "Inventory systems with stochastic demand and supply: Properties and approximations." European Journal of Operational Research 206.2 (2010): 313-328.

[4] Khashei, Mehdi, and Mehdi Bijari. "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting." Applied Soft Computing 11.2 (2011): 2664-2675.

[5] Box, George EP, et al. "Time series analysis: forecasting and control." (2015)

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Ondřej Vaněk, Ph.D., centrum umělé inteligence FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: 19.02.2018

Termín odevzdání diplomové práce:

Platnostzadání diplomové práce: 30.09.2019

Ing. Ondřej Vaněk, Ph.D. podpis vedoucí(ho) práce podpis vedouci(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc. podpis děkena(ky)

CVUT-CZ-ZDP-2015.1

Abstract

Efficient inventory management plays an important role for competitiveness of all companies that deliver goods to customers. The main goal of this thesis is to design an inventory optimization model that makes the best possible ordering policies.

In order to reach this goal, we combine methods of time-series prediction together with stochastic optimization concepts. We create a flexible inventory control pipeline, which is capable of generating goods ordering decisions that consider demand uncertainty, goods durability, shortage costs, warehousing costs, and both fixed and per-unit ordering costs. The pipeline is enriched with the progressive hedging decomposition algorithm, which helps to reduce computation times and improves the capability of our models to reduce risks of unexpected demand outcomes.

Keywords: Time series forecasting, exponential Smoothing, ARIMA, inventory control, stochastic optimization, newsvendor model, decomposition

Abstrakt

Efektivní řízení skladových zásob hraje klíčovou roli pro konkurenceschopnost podniků prodávajících zboží. Hlavním cílem této práce je navrhnout model tvořící objednávky zboží se záměrem optimalizovat vývoj skladových zásob.

Za účelem dosažení tohoto cíle kombinujeme metody predikce časových řad s koncepty stochastické optimalizace. Práce popisuje tvorbu modulu, který plánuje objednávky s ohledem na neurčitost poptávky, životnost produktů, možné náklady vyplývající z nedostatku zboží, skladováním zboží a objednáváním zboží. Modul je dále vylepšen pomocí "progressive hedging" algoritmu, který snižuje časovou náročnost modulu, a dále vylepšuje jeho schopnost snižovat rizika spojená s neočekávaným vývojem poptávky.

Klíčová slova: Predikce časových řad, exponenciální vyrovnávání, řízení zásob, stochastická optimalizace, newsvendor model, dekompozice

Contents

1	Intr	oducti	ion	1			
	1.1	Thesis	s goals	2			
2	Technical Background 4						
	2.1	Time s	series forecasting	4			
		2.1.1	Exponential Smoothing	4			
		2.1.2	ARIMA	6			
		2.1.3	Neural networks	8			
	2.2	Invent	cory optimization	9			
		2.2.1	Costs	9			
		2.2.2	Modeling	9			
		2.2.3	Base stock model	10			
		2.2.4	Stochastic programming	10			
		2.2.5	Newsvendor (Inventory) model	11			
3	Rel	ated w	vork overview	16			
0	3.1	Time s	series forecasting	16			
	3.2	Invent	For optimization	16			
	0.2	111,0110		10			
4	Solution approach 19						
	4.1	Motiva	ation \ldots	19			
	4.2	Assum	nptions	19			
	4.3	Prepro	ocessing	20			
	4.4	Demai	nd prediction \ldots	20			
		4.4.1	ARIMA model	20			
		4.4.2	Exponential smoothing model	22			
	4.5	Scenar	rio generation \ldots	23			
	4.6	Invent	cory control	23			
		4.6.1	Base linear program	23			
		4.6.2	Linear program extensions	24			
		4.6.3	Final linear program	26			
		4.6.4	Decomposition	26			
5	Evaluation 29						
0	5.1 General description of our evaluation						
	5.2	Demai	nd prediction	29			
	0	5.2.1	Evaluation metrics	30			
	5.3	Foreca	asting evaluation results	30			
	0.0	5.3.1	SKU1 forecasts	32			
		5.3.2	SKU2 forecasts	34			
		5.3.3	SKU3 forecasts	35			
		5.3.4	SKU4 forecasts	36			

	References					
7	Appendix					
	6.1	Future work	54			
6	Con	nclusion	53			
	5.6	Evaluation conclusion	52			
		5.5.1 Inventory control evaluation conclusion	52			
	5.5	Running time evaluation	51			
		5.4.4 Shortage costs evaluation	50			
		5.4.2 Testing parameters	43 44			
		5.4.1 Models	42			
	5.4	Inventory control	42			
		5.3.7 Forecasting evaluation results summary	42			
		5.3.6 Residual analysis	38			
		5.3.5 SKU5 forecasts	37			

1 Introduction

For many companies, inventory management is one of the most complicated issues in business. Manufacturers, suppliers and retailers need to decide how many goods they need to prevent stock-outs and avoid excess inventory, all this under conditions that can change any time. If they don't meet the customer's demand, the customers can leave for competition. On the other hand, excess inventory can lead to large costs and possible financial loss. This is definition of the inventory control problem, and the solution of this problem should provide a policy in a form that answers question of when and how much goods to order.

Many inventory optimization methods can be used to solve inventory control and other related problems of the companies, depending on the exact conditions in which the company operates. One of the most versatile is the Newsvendor model, which solves the inventory control problem as a trade off between expected leftover and shortage costs, considering stochastic demand.

In this thesis, we propose an extension of the Newsvendor model, which can, based on demand forecast, generate ordering policies for long periods of time. This model can deal with goods durability, fixed and per-unit ordering costs, shortage costs and warehousing costs. All of the costs can be time-dependent. The model also takes the forecast uncertainty into account. Multiple ways of how to tackle the forecast uncertainty exists, but the most promising method is scenario based sampling, which is also used by the model. The finer the sampling, the more precise are the results. Unfortunately, the more samples the model obtains, the higher the computational demands. Therefore we propose a decomposition method for the original model, which solves only an approximation of the original problem, but its computation time grows only linearly with the number of considered demand outcomes. We further evaluate both the original model and the approximating model on real world data, and we discuss the trade off between solving the exact problem with less demand scenarios in consideration, or solving the approximation with much more demand outcomes "in mind".

None of the above introduced models could work without a demand forecast, therefore we also study the problem of time series prediction in this thesis. We develop a demand prediction module, which includes custom extension of exponential smoothing, and the seasonal ARIMA model. We evaluate them both using accuracy measures and visually, and we compare them to another popular forecasting model - the Prophet model. We also examine other properties of the models and their suitability for generation of input to the inventory models.

1.1 Thesis goals

1. Understand the problem of inventory optimization and demand prediction

We provide an overview of demand forecasting in 2.1 and inventory optimization in chapter 2.2, we also provide examples of previous related work in chapter 3. Based our findings we develop demand prediction and inventory modules in chapter 4.

2. Study various algorithmic approaches to inventory optimization and demand prediction.

We study multiple algorithmic approaches to demand prediction in chapter 2.1 and inventory optimization methods in 2.2. In chapter 3 we also provide overview of demand prediction and inventory optimization methods used in previous research and finally in chapter 4 we use our findings to create a custom solution.

3. Develop an algorithm for demand prediction based on historical data provided by thesis supervisor.

In section 4.4, we develop an extension of the exponential smoothing algorithm for demand prediction. We also include the seasonal ARIMA prediction model in our demand forecasting module, as it is a proven method for time-series prediction. We further develop a demand scenario generation module in subsection 4.5, which generates possible outcomes of the demand according to the forecasts. These demand scenarios are later used as an input to our stochastic inventory models.

4. Integrate demand prediction with an inventory optimization model such that the model considers costs of unavailability, resupply cost as well as warehousing costs.

We develop a custom extension of stochastic Newsvendor inventory model in chapter 4.6. This model considers costs of unavailability, both fixed and per unit resupply costs and warehousing costs. All of these costs can be time variable and the model considers goods durability as well. We further develop a custom approximation of this model, which can work with more demand information than the base model.

5. Demonstrate the performance of the inventory optimization model on a set of possibly real-world scenarios.

In chapter 5 we evaluate both the demand forecasting models and the inventory models using real world sales data provided by the supervisor. We first inspect the accuracy and properties of the forecasting models, we also provide a comparison to another popular forecasting approach. In the second half of chapter 5 we evaluate the inventory models using the real world data. We discuss the trade off between an exact solution with less demand information, and its approximation form which can exploit more information.

2 Technical Background

2.1 Time series forecasting

Since the target of this thesis is to create an inventory optimization module, which contains a demand forecasting module, we dedicate this section to the field of time series forecasting. Time series is basically anything that is observed over time sequentially [1], including the demand data that we have at hand. Time-series forecasting consists of methods which try to predict future values of a sequence based on historical observations. It is a field with many applications, including demand prediction [2], but also others such as unemployment rate prediction [3].

Before we can create time series forecasts, we need to perform multiple steps. First, we have to gather information about the time series, perhaps historical records, e.g. data including monthly number of airline passengers (the "Observed" line in figure 1), or other knowledge about the series. Second, we need to perform analysis of the data. This can include search for components like trend - a long term increase or decrease in the series, and seasonality - influence by seasonal factors like week or month or cyclic behavior (long term patterns without fixed period). Examples of both trend and seasonality can be seen in figure 1). Third, we need to select or create a forecasting model, according to the found properties of the time-series. For example we can select exponential smoothing model or recurrent neural network (examples of other forecasting models are introduced later in this section). After training the model, we can finally create forecasts and evaluate the model using measures like mean absolute error, or we can evaluate the model's performance visually. We can also check the properties (such as normality) of residuals of the model. Residuals of a model are difference between a fitted value and the observed value - can be seen in bottom plot of figure 1. We perform evaluation of forecasting methods in section 5.2.

2.1.1 Exponential Smoothing

Exponential smoothing is a set of methods, which were introduced already in 1950s [2] and have been applied to various problems from many areas such as retail or energy industry. The smoothing methods are still commonly used today, as they enable researchers to quickly create simple and relatively well performing models. It produces forecasts based on weighted averages of past observations. The name exponential comes from the way this method weight the past observations, as the weights of older observations decay exponentially, which means that more recent observations are associated with higher weights [1].

The simplest of the exponential smoothing methods is called "simple exponential smoothing". It provides forecast only based on weighted averages of past observations, but it differs from simple weighted average as the



Figure 1: Decomposition of airline passengers time series

past observations have exponentially decaying weights. We can write the equations for simple exponential smoothing as follows:

$$\hat{y}_{t+1|t} = l_t l_t = \alpha y_t + (1 - \alpha) l_{t-1}$$
(1)

Where l_t is the smoothed value of the series as time t, y_t is the observed value at time t, $\hat{y}_{t+1|t}$ is the forecasted value a time t+1. The parameter α may be selected arbitrarily, or by optimization methods. A common approach is by minimizing some sort of error, most frequently sums of squared errors (SSE), which is defined as follows:

$$SSE = \sum_{t=1}^{T} (y_t - \hat{y}_{t|t-1})^2$$
(2)

Although the simple exponential smoothing might provide good forecasts in some cases, it is not suitable for forecasting time series with trend or seasonality. To forecast data with trend, simple exponential smoothing was extended to "double exponential smoothing". The extension consists of adding one more equation to the simple method, which is responsible for smoothing the trend of the series. Double exponential smoothing consists of the following equations:

$$\hat{y}_{t+h|t} = l_t + hb_t$$

$$l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1})$$

$$b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1}$$
(3)

Where β is a trend smoothing coefficient, b_t is the trend estimate at time step t and $\hat{y}_{t+h|t}$ is the h steps ahead forecast.

Even though double exponential smoothing provides a model for wider range of time-series than the simple method, the model does not capture seasonality. Therefore the Holt-Winters method was developed [1], which adds yet another component to the model - a seasonal component, to improve forecasting of series with seasonality. Actually two variants of the Holt-Winters method exist, additive and multiplicative. In this thesis, we use an extension of the additive method, therefore we can show its formulation:

$$\hat{y}_{t+h|t} = l_t + hb_t + s_{t-m+1+(h-1)mod(m)}$$

$$l_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(l_{t-1} + b_{t-1})$$

$$b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1}$$

$$s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m}$$
(4)

Where γ is a smoothing parameter of the the seasonal component, s_t , m is the period of the seasonality (e.g. 7 for weekly seasonality and daily data). As in the case of single and double exponential smoothing, the parameters α , β , γ can be obtained by minimization. We can see that apart from adding the seasonality term, the level component was adjusted by subtracting the seasonal component.

We should remind the reader, that the are many other possible extensions of exponential smoothing methods, but they are based on the general formulations provided here, a comprehensive list of variants of the exponential smoothing can be seen in [1]. We show some extensions in section 4.4 as we decided to use a variant of exponential smoothing for our demand prediction task.

2.1.2 ARIMA

Together with exponential smoothing, ARIMA (Autoregressive Integrated Moving Average) models are the most widely used approaches to time series forecasting [1]. ARIMA models are a class of models for which some d-th difference of series is stationary [4]. A stationary series is a series whose properties such as mean and variance are not time dependent. For example some series that have a trend or seasonality component are not stationary, but a white noise series is stationary [1]. An ARIMA model is actually a combination of two models with - the AR (autoregressive) and MA (moving average) model, with *differencing*. Both AR and MA models can be used to model various types of time-series patterns, but they can't be used for non-stationary time-series.

There are multiple approaches of making a time-series stationary, such as the logarithm transformation, or differencing. Differencing, which is a process of computing differences of consecutive observations, is used by the ARIMA models. A differenced series can be written as follows:

$$y_t \prime = y_t - y_{t-1} \tag{5}$$

Where y_t is the observed value a time t. Sometimes it is needed to difference the series a second time in order to make it (more) stationary. Second order differencing is defined as follows:

$$y_t \prime \prime = y_t \prime - y_{t-1} \prime \tag{6}$$

In some cases, seasonal differencing, which is the difference between the current observation and the observation from last seasonality period, is applied:

$$y_t \prime = y_t - y_{t-m} \prime \tag{7}$$

Where m is the number of seasons. Sometimes it is necessary to do both a seasonal and first order difference to make a series stationary [1]. The order of differencing of ARIMA models is denoted with letter d.

Once we have obtained a stationary series, we can use an autoregressive model of order p - AR(p), which is defined as follows [1]:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \phi_p y_{t-p} + \epsilon_t \tag{8}$$

Where c is a constant and ϵ_t is an error term. The forecasts of such model are linear combinations of past observations.

We continue with the definition of a moving average model of order q - MA(q), which uses past prediction errors e_{t-1}, \ldots, e_{t-q} to create a forecast [1]:

$$y_t = c + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \theta_q e_{t-q} \tag{9}$$

By combining differencing of order d, AR(p) model of order p, MA(q) of order q we obtain an ARIMA(p,d,q) model. However ARIMA models are also capable of forecasting seasonal data [1]. By including additional seasonal autoregressive, differencing and moving average terms in the model, we obtain a seasonal $ARIMA(p,d,q) \times (P,D,Q)_m$ (SARIMA) model. Where:

- P is the order of the seasonal autoregressive part
- D is the order of the seasonal differencing
- Q is the order of the seasonal moving average part

• m is the seasonality period

Further accuracy improvements can be obtained by combining an ARIMA model with a model that considers external factors. In retail industry, such factor could be holidays, price changes or weather. An example of such model is SARIMAX [5]. In general, its forecast can be written as a multi linear regression problem:

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \beta_k x_{1,k} + \omega_t \tag{10}$$

Where $x_{1,t}, \ldots, x_{1,k}$ are external variables and ω_t is a residual series, which can be represented as a seasonal ARIMA model.

2.1.3 Neural networks

Neural networks are a class methods of artificial intelligence that allow to model complex nonlinear relationships between response variables and predictor [1]. Neural networks also allow forecasting with external factors by just adding another input. This all makes them highly suitable for task of time series prediction. We can name multiple applicable neural network architectures that can be used for the time series forecasting task. From very simple networks that have no hidden layers and only perform linear regression on the data, to feed forward networks, which can use lagged values of the time series to perform auto-regression [1]. Also architectures that contain Long short-term memory (LSTM) units can be used for this task [6] as they have the ability to remember and exploit long term dependencies in data.

2.2 Inventory optimization

In this section we focus on inventory optimization, which is a field of great interest of manufacturers, suppliers, retailers and others. Inventory optimization aims to create optimal goods (or e.g. materials, components) ordering policies to meet goals such as maximal availability or minimal cost. To meet such goals, inventory optimization tries to answer questions like *what* to order, *when* the order should be placed and *how much* goods should be ordered.

2.2.1 Costs

In order to be able answer the *what, when, how much* types of questions, researchers use a wide range of algorithms and models. What many of these models (and also the approach used in this thesis) have in common, is that for the determination of the inventory policy they work must with at least a subset of the following costs[7]:

- **Procurement costs.** Procurement costs are fixed or variable costs related to acquisition of the goods. These may include per-order fees, per-unit purchase cost, transportation or handling costs.
- Shortage costs. Shortage costs *are incurred* when customer demand is not met. This can mean lost sales costs, back order costs or even customer dissatisfaction costs.
- **Inventory holding costs.** Inventory holding costs are the costs of storing goods for some period of time. This may include warehousing costs or *opportunity cost* the costs for not investing companies capital in more profitable way.
- **Obsolescence costs.** Obsolescence costs are the costs of lost value of stocked items. This can mean deteriorating food or flower inventories, clothing items going out of fashion etc.

2.2.2 Modeling

As already stated, researches use many types of models to solve inventory optimization problems. In this thesis, we don't focus on deterministic models such as *Economic ordering Quantity* [8], where demands, prices etc are known, even though they have a broad range of applications. Instead, we mainly focus on the class of *stochastic* models. In the following lines we introduce two such models, the Base Stock [9] model and the Newsvendor model [10].

2.2.3 Base stock model

The first stochastic inventory model we will introduce is the Base Stock model, which is a well known model that can provide ordering policy for infinite number of periods. However, the model is in its base form is very limited, as it provides only the ordering amount based on random demand data and expected service level (the probability of not stocking out). It also assumes that products can be analyzed individually and replenishments have known lead time (the time required between making an order and receiving the goods). In addition, only fixed ordering costs are considered by the model.

The steps of the model can be described as follows [9]:

• First we need to determine the review period (the time between successive revisions of the inventory) denoted T:

$$T = \sqrt{\frac{2K}{h\mu}} \tag{11}$$

Where K is fixed ordering cost, h is daily holding cost(the cost of keeping one unit of product in inventory) and μ is mean daily demand.

• If we assume the demand during review period T is from normal distribution with mean μ and variance σ^2 , the ordering amount S for given service level α according to base stock model is:

$$S_T = \mu + z(\alpha)\mu\tag{12}$$

Where $z(\alpha)$ is the value of the inverse cumulative distribution function of standard normal distribution for α .

The model could be theoretically adjusted, so that it considers more inventory aspects like goods durability - by modifying the review period. However, it is not suitable at all for optimizing orders according to goods ordering (or other) costs changes. Therefore in the following sections we introduce the Newsvendor model, which has a very flexible linear programming formulation.

2.2.4 Stochastic programming

Before we introduce another stochastic inventory model - the Newsvendor model, we should also briefly provide introduction of the area of Stochastic programming, whose concepts are used by the Newsvendor model [10].

Stochastic programming is an approach for modeling and solving optimization problems that involve uncertain parameters [11]. Even though there are many other ways how to solve problems involving uncertainty, stochastic programming has proven its usefulness in many areas such as telecommunications, economics, energetics or medicine [10] [12] [13] [14]. A common application of stochastic models is in a setting when a decision must be made [10], for example in problem of investment portfolio optimization. If we knew exactly the future value of the possible investments, we could create a deterministic model that would give us the optimal portfolio. However in real world scenario we usually don't have such precise information, and we have to work (for example) with probability distribution estimates created from data that have been collected over time (in case of portfolio optimization problem that would be the investment future value probability distribution).

In the following lines we introduce some of the basic terms and methods related to stochastic programming. We will illustrate them using the stochastic Newsvendor model, which itself serves as a base for solution of the inventory optimization problem introduced by this thesis.

2.2.5 Newsvendor (Inventory) model

The basic newsvendor model is another stochastic model used to determine optimal inventory levels. Its goal is maximizing expected profit by finding an optimal trade-off between risk of over-stocking and the risk of understocking. In the basic setting, the model assumes zero ordering lead time, no ordering and inventory amount limits and allows inventory shortages. Also the basic setting only provides ordering policy for one ordering period and for one product. However, as the model solution also has linear programming formulation, it can be easily extended. In contrast with the Base Stock model, the model accounts for cost of goods unavailability [9].

The output of the model is the ordering quantity x, which should satisfy the demand d. The unit ordering cost is c. If the demand is greater than the ordered amount of units, a per unit backorder penalty b is incurred. On the other hand, if the demand is smaller than x, a per unit holding cost is incurred. The objective is to minimize the total cost G(x,d), which is defined as follows [11]:

$$G(x,d) = cx + b[d-x]_{+} + [x-d]_{+}$$
(13)

Where $[a]_+$ denotes the maximum of a and 0.

However in the common real world case, the retailer does not know the demand d, as e.g. he needs to determine the optimal inventory level before a sale season with uncertain demand. To cope with this, we can view the demand D (denoted by D to highlight that we mean the random variable instead of its particular realization d) as random variable [11]. If we assume that the distribution of D is known, because we can for example estimate it from past demand observations, we can define the following optimization problem, which aims to minimize total cost expected value [11].

$$\min_{x \ge 0} \mathbb{E}[G(x, D)] \tag{14}$$

The goal of the above mentioned problem is to minimize the total cost on average, its justification comes from the Law of Large Numbers [11]. In case of the basic Newsvendor problem, the expected value formulation has a closed form solution. However, this is not always the case in stochastic programming problems. One way to tackle this is using **scenarios**. We can illustrate the scenario approach using the Newsvendor problem as well [10].

Let's now suppose that the demand random variable D has a finitely supported distribution, which means it takes values $d_1, ..., d_k$, called *scenarios*, with respective probabilities $p_1,..., p_k$. In such case we can model the stochastic program as deterministic optimization problem and write the expected cost as the following sum:

$$\mathbb{E}[G(x,D)] = \sum_{k=1}^{K} p_k G(x,d_k)$$
(15)

And to obtain the ordering quantity x, that minimizes the expected cost, we can use to solution of the following linear program:

$$\min_{\substack{x,t_1...t_k\\k=1}} \sum_{k=1}^{K} p_k t_k$$
s.t. $(c-b)x - t_k \leq -bd_k, \ k = 1, ..., K$
 $(c+h)x - t_k \leq hd_k, \ k = 1, ..., K$
 $x \geq 0$

$$(16)$$

Using the concept of scenarios we obtained the "extensive" form of the problem, which is a tractable approximation of the expected value problem. The scenarios can be obtained in very diverse ways, they can be samples from a known discrete probability distribution, or they can be result of discretization or simulation, the source can of scenarios can also be some limited sample information, and others. This kind of model is also called the a two stage stochastic programming model, as in the first stage we make some decision (orders) that has some cost, and in the second stage we deal with the consequences of the decision and we "pay" the costs related to the consequences.

So far we have introduced only a relatively limited version of the Newsvendor problem (and stochastic programming problems in general), where only one decision is being made at one point in time. However it is quite common that we want to solve a problem where decisions (e.g. order amount) will be made sequentially at certain periods of time, and each decision should



Figure 2: Scenario tree example

be made using the knowledge of outcomes of previous decisions (e.g. leftover inventory). This type of problems are called **Multi-stage stochastic** problems.

In the multistage Newsvendor problem, we suppose that the retailer has a finite planning horizon of T periods, and the demand is modeled as a random process D_t indexed by the time t = 1, ..., T [10] At each stage t = 1, ..., T, the retailer observes the current inventory level y_t , and replenishes the inventory to level x_t by ordering $x_t - y_t$ units. Then the starting inventory level at time t+1 y_{t+1} is obtained from equation $y_{t+1} = x_t - d_t$, where d_t is a particular demand realization at time t. The objective is to minimize the expected total cost over the planning horizon, which we can write as follows [10]:

$$\min_{\substack{x_t \ge y_t}} \sum_{t=1}^T \mathbb{E}\{c_t(x_t - y_t) + b_t[D_t - x_t]_+ + h_t[x_t - D_t]_+\}$$

s.t. $y_{t+1} = x_t - D_t, \ t = 1, ..., T - 1$

As in the simple (single stage) model, we can again employ the concept of scenarios, or more precisely, **scenario trees**. If we consider the random process $\xi_1, ..., \xi_T$ has a finite number (K) of realizations $(\xi^1, ..., \xi^k)$, we can depict the possible sequences in a form of a scenario tree (see example in figure 2). In general, nodes at each level t of the tree represent all possible values of ξ^t . The tree has only one node at level t = 1, which branches in nodes at level t = 2, the branching continues up to level t = T. However the branching factor does not have to be same for all nodes, as the reader can see in the example. Numbers along arcs between nodes represent conditional probabilities of moving to next node. Each scenario is represented as a path from the root at level t = 1 to some node at the last level t = T. The probabilities written along the path segments. As an example we can take scenario $w^s = (\xi_1^s, \xi_2^s, \xi_3^s) = (1, 3, 5)$, which has the probability $p^s = 0.25 \cdot 1$.

Finally, we can write the following linear program, which enables us to find optimal ordering amounts under each scenario and time stage t $x_{k,t}$,

where k is the number of scenarios and k is the number of stages.

$$\min_{x_{k,t}} \sum_{k=1}^{K} p_k \sum_{t=1}^{T} z_{k,t}$$
s.t. $z_{k,t} \ge (c_t - b_t) x_{k,t} + b_t d_{k,t} - c_t y_{k,t}, \ k = 1, \dots, K, \ t = 1, \dots, T$ (17)
 $z_{k,t} \ge (c_t + h_t) x_{k,t} - h_t d_{k,t} - c_t y_{k,t}, \ k = 1, \dots, K, \ t = 1, \dots, T$
 $x_{k,t} \ge 0$
 $y_{k,t+1} \ge x_{k,t} - d_{k,t}$

Where c_t , h_t , b_t are the per unit ordering, holding and shortage costs at time stage t, $d_{k,t}$ is the demand at time stage t under scenario k, p_k is probability of scenario k, $x_{k,t}$ is the inventory level at time stage t under scenario k, $y_{k,t}$ is leftover inventory from stage t-1 under scenario k, $z_{k,t}$ is expected cost at time stage t under scenario k.

However, such model does not make sense, because it allows different decisions for scenarios with common history, and also allows the decision to be adjusted according to future stochastic realizations. Therefore *non-anticipativity constraints* need to be added to the linear program, ensuring that each decision $x_{k,t}$ depends only on information known up to stage t under scenario k. The non-anticipativity constraints look as follows [10]:

$$x_t^k = x_t^l, \ \forall k, l \text{ for which } \xi_{[t]}^k = \xi_{[t]}^l, \ t = 1, \dots, T$$
 (18)

Where $\xi_{[t]}^k$ denotes the scenario history up to time t. For example if $\xi^k = (2,3,4)$, then its subsequence until t = 2 is $\xi_2^k = (2,3)$.

So far, we have shown how to create a multi stage stochastic linear program. However, with increasing number of stages or scenarios, such linear program can easily become untractable in some way. It can happen that the model does not fit in computer's RAM or it is very computationally demanding. To overcome this, researches can use simple techniques such as limiting the number of scenarios, for example by decreasing the branching factor from some later stage, or by grouping the stages [11], e.g. in the case of Newsvendor, weekend days can be grouped with Fridays if orders can't be made during weekend. Researchers can also employ more sophisticated approaches, such as:

• Sample average approximation

This method uses a Monte Carlo simulation to obtain a problem of a manageable size. It approximates the original stochastic expectation function by generating iid samples ξ^1 , ..., ξ^N of random vector ξ , and for this sample it solves the deterministic equivalent problem (the sample is used as a set of scenarios). By sampling and solving the

deterministic problem, it obtains a set of candidate solutions, which are later compared by computing the optimality gap [10].

- Benders decomposition Benders decomposition [15] is one of the decomposition approaches often employed in context of stochastic programming. In general, it decomposes linear programs in a relaxed master problem and subproblems by fixing some of the variables in the linear program. The master and subproblems are being solved iteratively. By solving the subproblems, we obtain feasibility and optimality constraints for the master problem. In multi stage setting, the master problem is always represented by decision at some node of the scenario tree, the subproblems are represented by decisions at its children nodes.
- **Progressive hedging algorithm** Progressive hedging algorithm (PHA) is another decomposition method often used in context of stochastic programming. It was introduced by Rockafellar et al. [16].In contrast with Benders decomposition, it decomposes the original problem by whole scenarios, not by each decision in the scenario tree. We will further introduce this decomposition method in following sections.

3 Related work overview

In this chapter we offer a brief overview of related previous work. The chapter is divided in two sections, one covers previous work related to demand and general time series forecasting, the second provides a summary of previous work related to inventory optimization.

3.1 Time series forecasting

The field of time series forecasting was strongly influenced by book of Box and Jenkins [4], where they offer a guide to ARIMA model identification, estimation and verification [17]. Popularity of their approach most likely arises from the versatility of the guide along with the ability of ARIMA models to fit various types of data. Hyndman et al. [17] provide a list with examples of research papers dedicated to time series forecasting, 9 of them used an ARIMA model, for instance to predict truck sales (in [18]) or telecommunications traffic ([19]). More recently, researchers use ARIMA models as a benchmark model, e.g. in Zhang et al.[20], or they use hybrid approaches of modeling, which combine ARIMA models with other approaches, such as Neural Networks [21].

Another commonly used forecasting method is exponential smoothing and its extensions. Exponential smoothing, was first introduced under the name in exponential smoothing [2], where the author used the method for demand forecasting. This exponentially weighted average method was later updated by Holt [22], who added a trend component to the model. Later Winter [23], in a research paper dedicated to sales forecasting, updated the model with a seasonality component, and thus created the Holt-Winters forecasting method. The model was further extended by multiple authors, for example with components helping in prediction of time series with multiple seasonalites [24].

In general, researchers apply many other approaches than just extended exponential smoothing models or ARIMA based models. With the massive popularity of deep neural networks in recent years, it is no surprise that researchers have also applied its concepts to time series forecasting. For instance in [25] the authors apply a deep model to taxi demand prediction, or in [26] they develop a model, which they benchmark on tide height prediction.

3.2 Inventory optimization

To our own knowledge, there are no previous works on the topic of multiperiod Newsvendor with time-variable costs and deteriorating items, but in general, there are many works solving various other extensions of the Newsvendor problem. However, we should mention that not all research work focuses on the case with stochastic demand, even though some studies on the topic of inventory problem with stochastic demand date back to 1958 [27]. Other researchers also focus on models which ignore some of the relevant costs.

A lot of research work focuses only on the inventory optimization part of the problem introduced in this thesis, however some works try to solve both the demand forecasting problem and the inventory problem as in this work. For instance Levina et al. [28] propose an online algorithm for the Newsvendor problem, which can be used in case no prior demand distribution information is known. Also Oroojlooyjadid et al. [29] provide an end to end solution for the newsvendor problem. They propose a deep neural network based approach, which, using previous demand data and observed features, directly outputs future ordering policy. Ning et al. [30] also use neural networks to predict warehouse customers' demands based on which they create ordering policies. Sachs et al. [31] focus on Newsvendor problem adjusted for the problem of censored demand. Censored demand occurs in case the demand is greater than the available inventory, as in such case we don't know the exact demand, which can further affect future demand predictions.

After reading the previous paragraphs the reader might think that most solutions of inventory optimization problems revolve around application of mixed integer programming or neural networks. However the range of techniques is wider. As an example we can name Daniel et al.[32], which combine a simulation and genetic algorithm approach to optimize a supply chain inventory costs.

We can also name a few authors, that solved a more closely related problem to the optimization part of this thesis, we can start with Kim et al. [33]. Kim et al. focus solely on the inventory problem and provide solution to multi period Newsvendor problem with transhipments. They provide an MILP formulation of the problem, which they try to solve more efficiently with the Progressive hedging decomposition method. They work also includes a table with comparison of works of other authors related to topic of (extended) Newsvendor problem. We can also mention Levi et al., as [34] they solve the regular multi period Newsvendor problem by the Simple Average Approximation approach. They also provide formal proofs of how many samples are required to guarantee that the sampling approach solution is close to optimal solution with arbitrary precision.

Finally, we can state that (to our knowledge) none of the previous works proposes a model of retailer that would be suitable for purposes of this work, as our approach provides ordering policies for multiple steps ahead, considers demand uncertainty, fixed ordering costs, goods durability, disallows back orders (back-order means that part of customer demand from one period can be satisfied in next periods). Also to our knowledge, none of the works suggests to use progressive hedging decomposition approach in order to significantly increase the number of demand scenarios considered by the model.

4 Solution approach

4.1 Motivation

Inventory control plays an important role for competitiveness of any company that provides goods for its customers. An example of such company could be an online store, which both delivers and distributes groceries. Goods shortages cause lower profits and they can easily lead to customer dissatisfaction. On the other hand excess inventory may force the store to sell goods at lower prices, or even worse it can lead to inventory write offs. Higher inventory levels also increase warehousing costs.

A module that would be able to forecast demand with high accuracy, combined with a precise inventory model, that accounts for possible errors of the forecasting module and considers all inventory related costs and goods durability, would be of great use for such companies. In this thesis we aim to create such module, with the focus on precise modeling of an online grocery store.

4.2 Assumptions

During the preparation of the solution approach used in this thesis, we had to make a few assumptions as it helped us to overcome technical difficulties, e.g. in case we did not know some parameters. The list of the most important assumptions with brief justification follows:

- Censored demand: Our models don't take censored demand into account. Censored demand occurs when demand is greater than the available stock. In such case, we only know that the demand was greater than the stock level, but we do not know the exact demand. Some research on this topic has been done by other authors e.g. [31].
- Product class prediction/ordering: During the initial research phase, we have found it very difficult to predict single product class demand. There are many possible reasons. One reason could be variance in product availability, or product sale discontinuation, e.g. because of new product. Therefore we focus on prediction and generation of inventory policies of *product classes*, instead of single products. This means, for example, that we predict the demand for all bananas, instead of single brand of bananas.
- Product class policy: Our models currently provide only product class ordering policies instead of per-product policies or whole multi-item policies.
- Expired products: We assume that expired products are removed immediately after expiration and older products are sold before new products.

• Zero order lead time (the time between making an order and receiving the order): During the preparation of this thesis we did not have access to length of order lead time so we do not consider it in our models.

4.3 Preprocessing

During initial research phase we have studied the influence of preprocessing methods on the accuracy of the demand forecasts. First we tried remove outliers from the historical data by smoothing methods, later we also tested the Hampel filter [35], which measures deviation of time series samples from median of their surrounding samples, and removes those samples that differ significantly from the median. However, the best results were obtained by simply replacing values that coincided with non-working holidays, as these samples were unnecessary outliers in the series, which can potentially harm the fitting process of the forecasting models.

4.4 Demand prediction

Even though the inventory model presented in section 4.6 can account for possible errors of the forecasting module that precedes it, the accuracy of the demand forecasts still plays a crucial role for its performance. Therefore we use the ARIMA and exponential smoothing methods which have many times proven its ability to provide accurate forecasts.

4.4.1 ARIMA model

The first model we decided to include is the seasonal ARIMA $(p, d, q) \times (P, D, Q)_m$ as it was many times successfully used for forecasting time-series from various areas. Because our inventory pipeline is expected to work automatically, we had to create an automated method of the model order selection. Our approach is based on a grid search approach, which tests various model order configurations. The final model order is selected by minimizing the Bayesian information criterion (BIC) [1]:

$$BIC = AIC + (log(T) - 2)(p + q + k + 1)$$
(19)

Where AIC is the Akaike information criterion, which is defined as follows:

$$-2log(L) + 2(p+q+k+1)$$
(20)

and L is the value of maximum likelihood function of the model, k is the number of parameters of the model, p is the autoregressive order and q is the moving average order of the model, T is number of observed samples.

To narrow down the number tested model configurations, we investigated some basic properties of the series. An example of the provided time series data can be seen in figure 3. The top plot shows scaled demand data, it



Figure 3: An example of the provided time-series data, together with correlation plots

is clear that the series is not stationary - it has increasing trend and likely time-changing variance (heteroscedasticity). Also the autocorrelation plots suggests the series is not stationary. Therefore at least one of regular or seasonal differencing terms have to be included. The seasonality period of this daily demand data series is likely to be 7 days, which we can also support by applying a seasonal differencing with m = 7. After differencing, we can evaluate the Augmented Dickey-Fuller test of stationarity, The test has the hypothesis that the series is not stationary. Clearly, the seasonal differencing has made the series stationary, as the test returned an almost zero p-value and we have to reject the null hypothesis.

Finally, we have decided to limit all parameters p,d,q to values 0,1 or 2, the P,D,Q values to 0 or 1, and m to 7. We prefer lower order values to prevent over-fitting.

4.4.2 Exponential smoothing model

The second model we decided to include in our prediction module is the exponential smoothing model, as it is easily interpretable and the parameter selection is straightforward and fast. In the exploratory analysis section, we have shown that the provided data contain signs of weekly seasonality and has increasing trend. Therefore it might seem that the Holt-Winters method as introduced in section 2.1.1 might be suitable. However, instead of using the regular Holt-Winters method, we initially decided to use multiple seasonality extension of the method similar to the one used in [24] so that the model could account for weekly, monthly and yearly seasonalities. After we have performed some preliminary benchmarks, we found out that the model is based on "double seasonal additive exponential smoothing method", and is formulated as follows:

$$\hat{y}_{t+h|t} = l_t + hb_t^{(h\phi)} + w_{t-m_1+h} + r_{t-m_2+h} \\
l_t = \alpha(y_t - w_{t-m_1} - r_{t-m_2}) + (1 - \alpha)(l_{t-1} + b_{t-1}) \\
b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1} \\
w_t = \gamma_w(y_t - l_{t-1} - r_{t-m_2}) + (1 - \gamma_w)w_{t-m_1} \\
r_t = \gamma r(y_t - l_{t-1} - w_{t-m_1}) + (1 - \gamma_r)r_{t-m_2}$$
(21)

Where $\hat{y}_{t+h|t}$ is the h steps ahead forecast from time step t, l_t denotes the level component at time step t, b_t is the trend estimate at time step t, w_t is the weekly seasonality component at time step t, r the yearly seasonal component, α is the level smoothing coefficient, β is a trend smoothing coefficient, γ_w , γ_r are smoothing parameter of the weekly and seasonality components. The parameters m_1 , m_2 denote the periodicity of the seasonal components, so $m_1 = 7$ is the periodicity of the weekly component, $m_2 =$ 365 of the yearly component. Another parameter is ϕ , which is a trend damping parameter. We decided to include it, so that it prevents longer term forecast to be strongly affected by the last trend estimate, which often happened in the preliminary phase of research.

The initial values of l_0 is selected as the first value in the time-series training data, b_0 is selected as the difference between the first two values in the series, $w_0 r_0$ are set to the mean difference between the first value and the values in the first period of the seasonality component.

For the coefficient selection, we use L-BFGS [36] method, which is a optimization algorithm for parameter selection. During the training, for every parameter set, the SSE error is computed for forecasting in last three periods from the end of the training series, each period has the same length as the future forecasting period, each time all data before the period is used to tune the model components.

4.5 Scenario generation

So far, we have described the used demand forecasting algorithms. In this section we describe our scenario generation procedure for our stochastic inventory models.

Once we have fitted any model described in section 4.4, we obtain the following:

- The forecasts x_1, \ldots, x_T where x_i is a forecast for future timestep i and T is the forecasting horizon
- The model residuals $\{e_1, \ldots, e_L\}$. Where each residual value is obtained from the difference $e_j = y_j \hat{y}_j$, which is the difference between the actual values at time step j denoted y_j and the fitted values \hat{y}_j , and L is the number of time steps in training data.

From the residuals, we compute their standard deviation estimate σ , and we sample "errors" from $\mathcal{N}(0, \sigma^2)$ which are added to the forecasts.

4.6 Inventory control

In previous sections, we have shown the proposed demand predictors and instructions on how to generate set of demand scenarios from these predictions. In this section we describe our inventory model, whose input is the set of scenarios and the output is the ordering policy for next periods.

We will formulate our solution using stochastic programming as we expect our models to account for demand uncertainty. More specifically, the solution will be based on the Multi period Newsvendor model, as the formulation is very flexible and allows us to further extend it to create more precise model for purposes of our task.

In section 2.2 we have provided an introduction to multi stage stochastic models, which are built on the assumption that we can make the decision for each time stage right before the time stage. However, our solution is based on **multi period** modeling approach, which differs from multi stage models by its assumption that all decisions are made already before the first stage.

4.6.1 Base linear program

As we have already stated, our solution is based on the stochastic Newsvendor model. The uncertainty of the solution is represented by a set scenarios and their respective probabilities and the set of decisions is obtained by solving a linear program. In section 2.2 we have shown the linear program for the base multi stage Newsvendor model. Considering the differences between the assumptions of the multi stage and multi period models, we can state the linear program for the multi-period Newsvendor as follows:

$$\min_{x_t} \sum_{k=1}^{K} p_k \sum_{t=1}^{T} z_{k,t}$$

s.t. $z_{k,t} \ge (c_t - b_t)x_t + b_t d_t - by_{k,t}, \ k = 1, \dots, K, \ t = 1, \dots, T$
 $z_{k,t} \ge (c_t + h_t)x_t - h_t d_t + hy_{k,t}, \ k = 1, \dots, K, \ t = 1, \dots, T$
 $x_t \ge 0, \ t = 1, \dots, T$
 $y_{k,t+1} = x_t + y_{k,t} - d_{k,t}, \ t = 1, \dots, T$ (22)

Where c_t , h_t , b_t are the per unit ordering, holding and shortage costs at time stage t, $d_{k,t}$ is the demand at time stage t under scenario k, p_k is probability of scenario k, x_t is ordered amount for time stage t, $y_{k,t}$ is expected leftover inventory from stage t-1 under scenario k, $z_{k,t}$ is expected cost at time stage t under scenario k.

We will use this mixed integer linear program as a base for our final solution.

4.6.2 Linear program extensions

Even though the proposed linear program 22 allows us to obtain the ordering amounts by optimizing some sort of expected cost, it does not fully capture the real world retailer's case, as it, for instance, does not account for goods deterioration. For this reason, we have to add the following necessary extensions to the base linear program:

• Disallow back orders: Because we are modeling am online grocery store, we have to remove the possibility of back orders (back order is a case of unsatisfied demand at some period satisfied in later period) in from the model, as back orders are not supported by the shop. Therefore, we have to update the constraint $y_{k,t+1} = x_t - d_{k,t}$ and replace it with $y_{k,t+1} = max(0, x_t - d_{k,t})$. Such constraints are commonly added to linear programs using the *big M method*, which is also recommended by the solver used by this study, Gurobi 8.0.0 [37], if the M constant is as small as possible. Therefore, we will replace the original constraint with the following:

$$y_{k,t+1} \geq x_t + y_{k,t} - d_{k,t}, \ t = 1, \dots, T$$

$$y_{k,t+1} \geq 0, \ t = 1, \dots, T$$

$$y_{k,t+1} \leq x_t + y_{k,t} - d_{k,t} + M(1-b), \ t = 1, \dots, T$$

$$y_{k,t+1} \leq Mb, \ t = 1, \dots, T$$
(23)

Where b is an additional binary variable and M is constant, either equal to maximum inventory level (if available), or to multiple to daily maximum demand in all scenarios. • **Per order fixed costs:** The base linear program only includes per unit ordering costs, although it is quite common that suppliers charge per order costs as well. Therefore, we add the following constraint:

$$x_t \leq w_t M, \ t = 1, \dots, T \tag{24}$$

Where w_t is binary variable indicating that order will be placed at time stage t. We also have to update the objective to include the per order costs:

$$\min_{x_t} \sum_{k=1}^{K} p_k \sum_{t=1}^{T} z_{k,t} + \sum_{t=1}^{T} f_t w_t$$
(25)

Where f_t is the per order cost at time stage t.

• SKU expiration date: Because our target is to generate ordering policies of a grocery online store for multiple weeks ahead, our model is also expected to include item expiration. For simplicity, we assume that the store sells goods in the order they were received, the expired goods are removed from inventory right after expiration, and the expiration date always occurs after fixed interval of **u** time stages after they are received in stock. We introduce another set of variables $l_{k,\tau,t}$, which denotes the left over inventory from time stage τ at time stage t under scenario k. Finally we update the constraints for $y_{k,t}$ and add constraints for $l_{k,\tau,t}$:

$$y_{k,t+1} \geq x_t + y_{k,t} - d_{k,t} - l_{k,t-u,t}, \ t = 1, \dots, T - 1$$

$$y_{k,t+1} \geq 0, \ t = 1, \dots, T - 1$$

$$y_{k,t+1} \leq x_t + y_{k,t} - d_{k,t} - l_{k,t-u,t} + M(1-b), \ t = 1, \dots, T - 1$$

$$y_{k,t+1} \leq Mb, \ t = 1, \dots, T - 1$$

$$y_{k,t+1} = \sum_{\tau=t-u}^{t} l_{k,\tau,t}, \ t = 1, \dots, T - 1$$

$$l_{k,t,t} \leq x_{k,t}, \ t = 1, \dots, T - 1$$

$$l_{k,\tau,t+1} \leq l_{k,\tau,t}, \ t = 1, \dots, T - 1, \ \tau = 1, \dots, T - 1$$

(26)

4.6.3 Final linear program

After adding all the extensions mentioned above, we obtained this final mixed integer linear program:

$$\min_{x_t} \sum_{k=1}^{K} p_k \sum_{t=1}^{T} z_{k,t} + \sum_{t=1}^{T} f_t w_t$$
s.t. $z_{k,t} \ge (c_t - b_t) x_t + b_t d_t - by_{k,t}, \ k = 1, \dots, K, \ t = 1, \dots, T$
 $z_{k,t} \ge (c_t + h_t) x_t - h_t d_t + hy_{k,t}, \ k = 1, \dots, K, \ t = 1, \dots, T$
 $x_t \ge 0, \ t = 1, \dots, T$
 $y_{k,t+1} \ge x_t + y_{k,t} - d_{k,t} - l_{k,t-u,t}, \ t = 1, \dots, T - 1$
 $y_{k,t+1} \ge 0, \ t = 1, \dots, T - 1$
 $y_{k,t+1} \le Mb, \ t = 1, \dots, T - 1$
 $y_{k,t+1} \le Mb, \ t = 1, \dots, T - 1$
 $y_{k,t+1} = \sum_{\tau=t-u}^{t} l_{k,\tau,t}, \ t = 1, \dots, T - 1$
 $l_{k,t,t} \le x_{k,t}, \ t = 1, \dots, T - 1$
 $l_{k,\tau,t+1} \le l_{k,\tau,t}, \ t = 1, \dots, T - 1, \ \tau = 1, \dots, T - 1$

4.6.4 Decomposition

It can be easily seen that our final final mixed integer linear program is quite complex. For illustration, a testing problem with three week planning horizon (21 time stages) and 100 scenarios has over 15000 variables. On the testing machine with Intel Core i7 7700HQ and 24 Gigabytes of RAM, the Gurobi optimizer found the optimal solution after more than 15 minutes, you can see results of an experiment with the running time rate of growth in figure 4. The task of the model in the experiment was to iteratively create inventory policy, while considering 5,10,20,40,60 and finally 80 scenarios. The running time for 80 scenarios that can be seen in the graph is not the actual running time. As the inventory policy generation took more than the 2 hour time limit, it was stopped before generating an optimal solution. The experiment therefore did not reject our hypothesis that the order of growth is exponential in number of scenarios. Considering that companies would likely need to create ordering policies for thousands of products, we decided to implement one of decomposition approaches introduced in section 2.2.5, the Progressive Hedging Algorithm. This decomposition methods improves the running time of the model, and allows us to use the model with more demand scenarios, which can further improve its performance, as can be seen in section 5.



Figure 4: Experiment with extensive form running time.

Progressive hedging algorithm's idea is straightforward, it decomposes the extensive form of a multi stage problem by scenarios in a set of smaller problems. The scenario subproblems are modified with a penalization term, which measures the deviation of the scenario solution from the scenario invariant solution. Even though convergence to globally optimal solution in the mixed integer programming case is not guaranteed, researchers have shown that the algorithm can find high quality solutions [38]. Another difficulty arises with selection of penalty parameters and converge acceleration techniques, however researchers such as. Watson et al. [39] provide ways around these problems.

To clarify, PHA is commonly used only for multi stage problems and the penalization terms are used to force non-anticipativity constraints. Even though our problem is a multi period model and does not contain any non-anticipativity constraints, we will use the penalty terms to enforce that the ordering policies $(x_{k,1}, \ldots, x_{k,T})$ generated by solving the subproblems converge to the scenario invariant policy $\hat{x}_1, \ldots, \hat{x}_T$. To include the penalty term, we update the objective function of the subproblem of each scenario s to the following:

$$\min_{x_{s,t}} \quad \sum_{t=1}^{T} z_t + \sum_{t=1}^{T} f_t w_t + \sum_{t=1}^{T} \lambda_{s,t} (x_{s,t} - \widehat{x}_t) + \frac{1}{2} \sum_{t=1}^{T} \lambda_{s,t} (x_{s,t} - \widehat{x}_t)^2 \quad (28)$$

Where λ is a Lagrange multiplier and ρ is a parameter.

The PHA is defined as follows:

Algorithm 1 Progressive Hedging Method for Multi period Newsvendor problem

1: Initialize variables ρ , γ and θ , set variable k = 0. 2: for each scenario $s \in S$ (set of all scenarios) do solve scenario sub-problem without penalization term 3: 4: end for 5: Set $\hat{x}_t = \sum_{s \in S} p_s x_{s,t}, t = 1, \dots, T$ 6: Set $\lambda_{s,t} = \rho_s(x_{s,t} - \hat{x}_t), s = 1, \dots, |S|, t = 1, \dots, T$ 7: for each scenario $s \in S$ do solve scenario sub-problem with penalization term 8: 9: end for 10: Update $\hat{x}_t \leftarrow \sum_{s \in S} p_s x_s, t = 1, \dots, T$ 11: Update $\lambda_{s,t} \leftarrow \lambda_{s,t} + \rho_s(x_{s,t} - \hat{x}_t), s = 1, \dots, |S|, t = 1, \dots, T$ 12: Update $\rho \leftarrow \gamma \rho$ 13: $\delta_t = \frac{\sqrt{\sum_{s \in S} (x_{s,t} - \hat{x}_t)}}{\hat{x}_t}, t = 1, \dots, T$ 14: for $t = 1, \dots, T$ do if $\delta_t \geq \theta$ then return to step 7 15: end if 16:17: end for 18: Terminate.

The algorithm definition raises a few questions, namely the following:

- The initial value of ρ , γ and θ : We have empirically selected the initial value of $\rho = 1$, $\gamma = 1.4$ and $\theta = 0.0001$. The value of γ ensures that the value of ρ almost doubles on every second iteration.
- Is the termination test on lines 15 17 sufficient? Our simulations have shown that additional tests need to be added. In [39] the authors propose PHA algorithm cycling behavior detection. At each iteration of the algorithm, they save the x_i components of the scenario invariant solution $\widehat{x_1}, \ldots, \widehat{x_T}$ using hashing. Once cyclic behavior of any of the decision variables $\widehat{x_i}t$ from $\widehat{x_1}, \ldots, \widehat{x_T}$ is detected between iterations, its value is fixed to $max(\widehat{x_{i,s}}), s \in S$.

We also check the following quantity Δ at each iteration $k \geq 1$:

$$\Delta = \frac{\sum_{t=1}^{T} |\widehat{x}_{k,t} - \widehat{x}_{k-1,t}|}{T}$$
(29)

Where $\hat{x}_{k,t}$ denotes the scenario invariant solution at iteration k. The value of Δ indicates the mean solution variable change between iterations. If $\Delta \leq \frac{1}{T}$ (means that on average only one variable changed by 1) for 10 consecutive iterations, we terminate the algorithm.

5 Evaluation

In this section we evaluate both topics of this thesis. For the evaluation purposes, we use the data provided by the supervisor. The evaluation consists of two tasks. First part is dedicated to evaluation of proposed demand forecasting methods, the second contains evaluation of inventory models, primarily with focus on the applicability of an extensive form inventory model and its decomposed form.

5.1 General description of our evaluation

First, we take all the proposed time series prediction models and we benchmark them on 5 selected product classes. Each algorithm has to provide forecasts for 3 periods of length of 28 days. The forecasts are then evaluated using standard accuracy metrics and also visual evaluation is performed. Second, we pick the best performing models, and compare their properties, primarily the residuals. Afterwards, we pick the model with the best properties, and we use its forecasts and residuals to generate demand scenarios for the inventory models. The inventory models are then benchmarked, every model will generate ordering policies for each of the 3 periods and each product class. Finally, the inventory models are evaluated according to costs associated with their ordering policies.

5.2 Demand prediction

As already mentioned, the evaluation consists of two parts. In this part we evaluate the demand prediction models on a set of 5 product classes (SKU) across 3 consecutive periods of length of 28 days. The periods are the following:

- From August 1st 2017 to August 28th 2017
- From September 1st 2017 to August 28th 2017
- From October 1st 2017 to August 28th 2017

Each time a model is tested for given SKU and period, it obtains all historical demand data available before start of the period.

In addition to the models introduced in section 4.4, we also include the Prophet[40] model for comparison of accuracies. The Prophet model is a forecasting model developed and used by the company behind the largest social network in the world. The model uses a decomposable time series model of the following form:

$$y_t = g(t) + s(t) + h(t) + \epsilon_t \tag{30}$$
Where y_t represents a fitted value at time step t, g(t) is a logistic or linear trend component, s_t is a seasonality component modeled by Fourier series, h_t represents possible effect of holidays and ϵ_t is an error term which is assumed to be normally distributed. For finding the model's parameters a L-BFGS [36] algorithm is used.

5.2.1 Evaluation metrics

We use mainly two performance measures of forecasting accuracy, root mean squared error and mean absolute percentage error. In general, a forecast error e at is defined as follows:

$$e_i = y_i - \hat{y}_i \tag{31}$$

Where y_i is the i-th observation of the time series, and \hat{y}_i is the forecast of the i-th observation. Root mean squared error (RMSE) is probably the most commonly used scale-dependent measure, it is defined as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (e_i)^2}$$
 (32)

The fact that RMSE is scale dependent makes it unsuitable for comparing forecasting performance on different time-series. However, for purposes of anonymization, we actually compute RMSE from scaled values:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\frac{y_i - \hat{y}_i}{m})^2}$$
(33)

Where m denotes the maximum observed value in the time-series. This adjustment allows for better comparison across different series. A more common scale-independent forecast error measure is the mean absolute percentage error (MAPE), which describes the error as a percentage value:

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{x_i}$$
(34)

5.3 Forecasting evaluation results

In this part of thesis we discuss the results of evaluation of the proposed forecasting models. Before a general summary, we first discuss the results by each product class in separate subsections. In each subsection, we provide a summary commentary. In addition to the commentary, each subsection contains a figure showing the product class' demand forecasts, where the forecasts of each model is shown separately, namely the figures 5, 6, 7, 8 and 9. In the appendix section 7 we also provide figures (15, 16, 17, 18 and

19) containing the forecasts for each product class, but with the forecasts provided by the models drawn in one plot. Nonetheless, each of the above mentioned plots also contains part of the time series history preceding the forecasting periods, and the start of each forecasting period is denoted by a dashed line.

We also provide tables with scores of each model in the respective product classes' sections, in addition to these per-product-class results tables, the appendix section 7 contains table 11, which contains all scores of all models for every product class.



Figure 5: Results of first product class - SKU1 forecasting evaluation

	Exp. Sn	noothing	ARI	MA	Prophet		
	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	
Period 1	5.918	0.042	8.536	0.062	10.907	0.077	
Period 2	9.459	0.079	10.176	0.083	17.506	0.139	
Period 3	9.327 0.087		10.917	0.095	14.426	0.129	

Table 1: Results of SKU1 forecasting

5.3.1 SKU1 forecasts

We first discuss the results of forecasting of the demand of the first product class (SKU1). The forecasts can be seen in figures 5 and 15 and the RMSE and MAPE scores are in table 1.

It is clear from both the figure and the table that the best performing models were the ARIMA and exponential smoothing models, which both provide quite similar forecasts, however the smoothing method has slightly better scores. Both models are well adapted for the weekly seasonality in the first period forecast, a little less in the other periods, maybe because they are over-fitted to the seasonality pattern - which has slightly changed in later testing periods. The weekly seasonality adjustment can be seen from the selected order of the (S)ARIMA model - $(0, 1, 2)x(0, 1, 1)_7$ in first period and $(1, 0, 2)x(0, 1, 1)_7$ in second and third. Also the smoothing model has selected high weekly seasonality smoothing parameter - γ_w was higher than 0.4 in all periods. Both models have underestimated the slope of the trend in second and third period, namely the smoothing model in second period, most likely because it selected quite low trend parameter $\beta = 0.06$. The prophet model's forecasts seem to be adjusted to the seasonality as well, but in somewhat damped way. In the second period, the Prophet model forecasted a decreasing trend, whereas it was actually increasing, which resulted in the worst score.



Figure 6: Results of second product class - SKU2 forecasting evaluation

	Exp. Sm	oothing	ARI	MA	Prophet		
	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	
Period 1	6.287 7	0.032	10.964	0.064	10.455	0.068	
Period 2	20.644	0.156	20.671	0.156	31.996	0.229	
Period 3	14.169	0.097	23.078	0.130	13.094	0.093	

Table 2: Results of SKU2 forecasting

5.3.2 SKU2 forecasts

Product class 2 (SKU2) consists of similar type of products as SKU1, which can be also seen if we compare figures 5 and 6. Both the historical data and forecasts show similar patterns. Again, a $(1, 0, 2)x(0, 1, 1)_7$ SARIMA model (this time for each period) was selected. The exponential smoothing method also used similar parameters $\alpha \sim 0.01$, $\beta \sim 0.9$, $\phi \sim 0.8$, $\gamma_r \sim 0.04$ and $\gamma_w \sim$ 0.4 to those selected for SKU1. Unfortunately, the models underestimated the trend slope in second period, and the decrease in level in third period as well. This resulted in overall worse scores than the models obtained for forecasting SKU1.



Figure 7: Results of third product class - SKU3 forecasting evaluation

	Exp. Sm	oothing	ARI	MA	Prophet		
	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	
Period 1	10.072	0.076	7.227	0.060	6.954	0.052	
Period 2	10.744	0.112	13.106	0.133	12.630	0.126	
Period 3	8.532	0.094	8.662	0.091	8.129	0.087	

Table 3: Results of SKU3 forecasting

5.3.3 SKU3 forecasts

Figure 7 contains the forecasts for product class 3 (SKU3), which is a class of ham products, the table 3 contains all scores. An ARIMA model of order $(1,1,2)x(1,1,1)_7$ was selected for all periods, which performed well in first and third testing period. The exponential smoothing method's parameters were selected always a little differently for each period. For first period it was $\alpha \sim 0.01$, $\beta \sim 0.95$, $\phi \sim 0.95$, $\gamma_r \sim 0.04$ and $\gamma_w \sim 0.34$, for second $\alpha \sim 0.11$, $\beta \sim 0.49$, $\phi \sim 0.92$, $\gamma_r \sim 0.04$ and $\gamma_w \sim 0.42$, and the third $\alpha \sim 0.17$, $\beta \sim 0.74$, $\phi \sim 0.92$, $\gamma_r \sim 0.04$ and $\gamma_w \sim 0.26$, so the trend and weekly seasonality parameters changed the most, but it always resulted in good scores. On average the Prophet model provided the most accurate forecasts by a slight margin. Forecasting of SKU4



Figure 8: Results of fourth product class - SKU4 forecasting evaluation

	Exp. Sn	noothing	ARI	MA	Prophet		
	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	
Period 1	19.258	0.164	10.685	0.108	9.692	0.106	
Period 2	11.662	0.122	11.891	0.125	10.561	0.107	
Period 3	7.311	0.081	11.276	0.118	7.139	0.070	

Table 4: Results of SKU4 forecasting

5.3.4 SKU4 forecasts

Product class 4 (SKU4) is a class of a hygiene product. From the historical data it seems that the weekly seasonality pattern is not as significant as in the demand time-series of previous products. Figure 8 contains the plot of forecasts for SKU4 against actual historical data, and the table 4 contains the scores of all models. For SKU4, an ARIMA model of order $(1,0,2)x(0,1,1)_7$ was selected for all periods which seems to follow the slightly increasing trend and the level as well. However the Prophet model's forecasts were more accurate. The exponential smoothing method provides similar results to both other methods in second and third testing period, but the selected parameters $\alpha \sim 0.01$, $\beta \sim 0.86$, $\phi \sim 0.8$, $\gamma_w \sim 0.25$ and $\gamma_r \sim 0.04$ for first period led to MAPE of almost 20%.

	Exp. Sn	noothing	ARI	MA	Prophet		
	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	
Period 1	9.438	0.067	7.677	0.051	10.245	0.066	
Period 2	15.991	0.088	16.248	0.087	15.372	0.089	
Period 3	17.712	0.127	8.320	0.061	9.686	0.071	

Table 5: Results of SKU5 forecasting

5.3.5 SKU5 forecasts

We will finish our per-product evaluation with product class 5 (SKU5). Figure 9 contains the plot of forecasts for SKU5 against actual historical data, and the table 5 contains the scores of all models. SKU5 is a class consisting of popular drink, which might be the reason why the historical data show a repeating weekly seasonality pattern. In the first testing period, all models seem to fit the seasonality quite well. However in the second testing period the difference between the minima and maxima of the series grew, and there is a slight drop of level at start of September (possibly because the summer holiday ends at that time). None of the models did forecast that. Possibly some product classes would require models adjusted for these "holiday" effects. For SKU5, an ARIMA models of orders $(1,1,2)x(1,1,1)_7$, $(1,1,1)x(1,1,1)_7$, $(1,1,2)x(0,1,1)_7$ were selected for each period respectively. The ARIMA model $(1, 1, 1)x(1, 1, 1)_7$ selected for the second period had the highest MAPE of them, maybe it would have been better if the model selection process (BIC criterion) selected same model for all periods. The exponential smoothing method provides similar results to both ARIMA in the first two periods. But the $\alpha \sim 0.32$, $\beta \sim 0.08$, $\phi \sim 0.95$, $\gamma_w \sim 0.54$ and $\gamma_r \sim 0.04$ parameters selected for third period resulted in MAPE of 17.7%. The parameters show signs of over-fitting as the level component is quite high and the trend component is low. Therefore future work might focus on improvement of the "training" process for both methods. For the holiday effects, a SARIMAX (Seasonal Arima with External Factors 2.1) model with additional "holiday" input might provide better results. Before we finish the commentary, we can state Prophet model achieved very high MAPE scores in every period.

Forecasting of SKU5 1.0 Actual value 0.8 ARIMA 0.6 0.4 1.0 Actual value 0.8 Exp. Smoothing 0.6 0.4 1.0 Actual value 0.8 Prophet



Date

Sep

Oct

Aug

Νον

5.3.6 Residual analysis

Jul 2017

Units sold

Units sold

0.0 0.0 0.4

We will now look at the residuals of the proposed forecasting methods, as we want to generate demand scenarios for the inventory models using the distribution estimated from the sample mean and sample variance of the residuals. Also the properties of the residuals also provide us with an insight in quality of the model. As stated in [1] a well fitted model's residuals have the following properties:

- The residuals are uncorrelated if the residuals are correlated, there is some information in the residuals which could be used in the forecasts but were not
- The residuals have zero mean. If the mean is not zero then then the model is biased (but it can be fixed by shifting the forecasts).

For the assessment of possible autocorrelations of the residuals, we can use the autocorrelation plot. An example of autocorrelation plot can be seen in figure 10, which was created from residuals of ARIMA and Exponential smoothing models selected for forecasting the demand in first testing period for SKU1. In the plot we can see that that the residuals of the exponential



Figure 10: Autocorrelation plots of residuals of ARIMA and Exponential smoothing models selected for forecasting of SKU1 in first testing period

smoothing model are significantly correlated, whereas the ARIMA model's residuals exhibit almost no correlation.

Another option of autocorrelation detection is the Ljung-Box test, which may be defined as follows:

- H0(null hypothesis): The data are independently distributed.
- H1(alternative hypothesis): The data are not independently distributed, they exhibit correlation.

The test statistic can be computed as follows:

$$Q^* = T(T+2) \sum_{k=1}^{h} (T-k)^{-1} r_k^2$$
(35)

Where T is the number of observations, h is the maximum considered lag ([1] suggests h = 2m for seasonal data, where m is the seasonality period), r_k is the autocorrelation at lag k, which is defined as follows:

	SKU1	SKU2	SKU3	SKU4	SKU5
ARIMA	0.15	0.18	0.32	0.13	0.28
Smoothing	~ 0				

Table 6: Ljung-Box test p-values for lag = 14 of residuals of models used for first period forecasting

$$r_{k} = \frac{\sum_{t=1}^{n-k} (x_{t} - \overline{x})(x_{t+k} - \overline{x})}{\sum_{t=1}^{n} (x_{t} - \overline{x})^{2}}$$
(36)

Table 6 contains the p-values of the Ljung-Box test statistic for first 14 lags computed from residuals of models selected for forecasting of demand in first testing period for all SKUs. According to the table, we can reject the hypothesis that the residuals of the exponential smoothing models are independent. A relatively high p-values of the ARIMA model would suggest that we can't reject the hypothesis that its residuals are independent.

	SKU1	SKU2	SKU3	SKU4	SKU5
ARIMA	0.84	0.81	0.85	0.84	0.89
Smoothing	0.90	0.87	0.91	0.85	0.94

Table 7: Shapiro-Wilk test p-values of residuals of models used for first period forecasting

Another important property of the residuals is normality, because our scenario generation approach assumes that the residuals are from normal distribution. For this we will use the Shapiro-Wilk test, which has the null hypothesis that the tested sample comes from normal distribution. Table 7 contains Shapiro-Wilk test p-values of residuals of models used for first period forecasting. We can see high p-values in the rows of each model, which means the we cannot reject the hypothesis that the residuals are from normal distribution.

5.3.7 Forecasting evaluation results summary

We have evaluated the proposed models both visually and using accuracy scores, and we also evaluated the models' residuals using statistical tests. Both our proposed methods performed better than that provided by the Prophet library. The Exponential Smoothing method has the best average accuracy scores, but some of its forecasts had signs of over-fitting. Also the residual auto-correlation evaluation showed the model (or rather the parameter selection) could be further improved. The ARIMA models' average MAPE is slightly higher, but its mean RMSE (computed from scaled series) is exactly same. However, the evaluation has shown that its residuals are likely not correlated and can be from normal distribution. Because of its good "on average" performance in all evaluation parts, we have chosen the ARIMA model as the model for generating demand scenarios for the inventory models.

Future research might focus on improving the model selection methods used for selecting the ARIMA models, namely we could theoretically use higher order models. In case of Exponential Smoothing the autocorrelation plots have shown it clearly didn't use some of the information in the historical data.

5.4 Inventory control

The next part of our evaluation is dedicated to the inventory control models. In this subsection, we only consider forecasts provided by the best forecasting model according to the evaluation in section 5.2 - the ARIMA model, and use it for scenario generation.

During the demand evaluation, the ARIMA model generated forecasts of demand for 5 SKUs across 3 consecutive periods of length of T = 28days. Each of these forecasts, and the fitted models residuals, are used for demand scenario generation. According to generated scenarios, the models proposed in 5.4.1 will output ordering amounts x_i, \ldots, x_T for each SKU and each period. The performance of the models is later compared according to costs, which would theoretically be incurred if the inventory policies provided by the models were applied in real world.

5.4.1 Models

In section 4.6 we have designed our extended formulation of the stochastic multi period Newsvendor inventory model. Due to the complexity of its mixed integer linear programming formulation and the computation complexity (especially when used with problem with more stage or scenarios) associated with it, we have suggested that a decomposition approach should be used - the Progressive Hedging Algorithm. Progressive Hedging Algorithm is commonly used for problems that can't be solved in its extensive form in reasonable time, or those that don't fit in RAM. This is not exactly the case of our formulation. As you can see in table 10, some of the less complex testing problems were solved in a couple of minutes. On the other hand, for some of the problems, the selected Gurobi solver did not provide an optimal solution for the extensive form with only 25 scenarios in under 20 minutes. Considering that an inventory of a store may contain thousands of products, we decided to include the following models in the evaluation, with "acceptable" computation time in mind:

- Extensive form model with 25 scenarios this model solves the exact formulation problem, but has rather limited amount of information about the possible outcomes of the random demand
- Progressive hedging model with 250 scenarios this models solves the decomposed problem, but due to the increased number of scenarios it can hedge against more possible outcomes of the random demand.
- Base stock model with daily restocking this model is very simplistic, its only objective is to prevent shortages

5.4.2 Testing parameters

We will now briefly list the parameters used by the above mentioned models and in the evaluation.

First we can repeat that the planning horizon T of the models is T=28 days. And the goods durability estimates are the following:

- SKU1 4 days as it represents is a fruit product
- SKU2 6 days as it represents a different fruit product
- SKU3 8 days as it represents packaged ham product
- SKU4 "unlimited' durability, as it is a hygiene product which most likely lasts longer than the planning horizon
- SKU5 "unlimited' durability, as it is a packaged drink which most likely lasts longer than the planning horizon

Second we list cost types. As we don't know the exact shortage, warehousing and other costs associated with each SKU, we decided to the use the following estimates:

- Shortage cost b_t : We use a value extracted from the historical data equal to the average daily selling price of the product class
- Ordering costs (per unit) c_t : we use the values $c_t = 0.65b_t$

- Fixed ordering costs (per order) f_t : we omitted this cost in evaluation
- Holding (warehousing) costs (per unit) h_t : we use the values $h_t = \frac{b_t}{50}$

It is clear that the values are very rough estimates and without expert's knowledge our total cost estimates are rather illustrative.

Now that we have the prices, we can show how we compute the estimated total costs. If we assume that the inventory model required the orders x_1, \ldots, x_T and the actual demands were d_1, \ldots, d_T , then we can calculate the costs as follows:

• The total shortage cost - S:

$$B = \sum_{i=1}^{T} b_t max(0, d_i - x_i)$$
(37)

• The total ordering cost - C:

$$C = \sum_{i=1}^{T} c_i x_i \tag{38}$$

• The total holding cost - H:

$$H = \sum_{i=1}^{T} h_t max(0, x_i - d_i)$$
(39)

• The overall cost V:

$$V = B + C + H \tag{40}$$

5.4.3 Total Costs evaluation

With computation time, the estimated costs is the most important measure when comparing the models mentioned above. The evaluation of the total costs associated withe the orders generated by the models can be seen in table 8, where PHA stands for Progressive Hedging Algorithm, EF for Extensive form. Estimated costs with a star mean that the algorithm did not finish in the time limit of 20 minutes and the result is only the best solution obtained before the limit.

It can be seen that in most cases, both the Newsvendor based models perform better than the simple Base Stock model, sometimes by up to 33% as in case of testing period 1 for SKU2. This is also reflected in the total sum of costs. PHA model has the lowest sum of estimated costs, EF model is behind by roughly 0.5%, and the Base Stock model has the largest sum of costs - 12% higher than PHA.

			E	Estimated costs				
		MAPE	PHA	EF	Base Stock			
	Period 1	8.54	$6.6\cdot 10^5$	$6.7 \cdot 10^{5}$	$7.9\cdot 10^5$			
SKU1	Period 2	10.17	$7.7\cdot 10^5$	$7.9 \cdot 10^{5}$	$8.6 \cdot 10^{5}$			
	Period 3	10.91	$8.5\cdot 10^5$	$8.7 \cdot 10^{5} *$	$9.0 \cdot 10^{5}$			
	Period 1	10.96	$5.7\cdot 10^5$	$5.8 \cdot 10^{5} *$	$7.3\cdot 10^5$			
SKU2	Period 2	20.67	$9.2 \cdot 10^{5}$	$9.5 \cdot 10^{5} *$	$8.8\cdot 10^5$			
	Period 3	23.08	$1.2\cdot 10^6$	$1.2 \cdot 10^{6} *$	$1.7 \cdot 10^{6}$			
	Period 1	7.23	$1.2\cdot 10^6$	$1.2 \cdot 10^{6} *$	$1.4 \cdot 10^{6}$			
SKU3	Period 2	13.10	$1.4\cdot 10^6$	$1.5 \cdot 10^{6} *$	$1.4 \cdot 10^{6}$			
	Period 3	8.66	$1.5\cdot 10^6$	$1.5 \cdot 10^{6} *$	$1.6 \cdot 10^{6}$			
	Period 1	10.68	$9.9\cdot 10^5$	$9.9 \cdot 10^5$	$1.0\cdot 10^6$			
SKU4	Period 2	11.89	$1.2\cdot 10^6$	$1.2 \cdot 10^{6}$	$1.2 \cdot 10^{6}$			
	Period 3	11.28	$1.4\cdot 10^6$	$1.4 \cdot 10^{6}$	$1.4 \cdot 10^{6}$			
	Period 1	7.68	$1.7\cdot 10^{6}$	$1.7 \cdot 10^{6}$	$2.4 \cdot 10^{6}$			
SKU5	Period 2	16.25	$1.8 \cdot 10^{6}$	$1.8\cdot 10^6$	$2.4 \cdot 10^{6}$			
	Period 3	8.32	$1.6\cdot 10^6$	$1.7\cdot 10^6$	$2.2 \cdot 10^{6}$			
		Sum	$1.79\cdot 10^7$	$1.80 \cdot 10^{7}$	$2.08 \cdot 10^7$			

Table 8: Estimated total costs evaluation

As the total sum of costs suggest, in most cases the Progressive Hedging model provided the best ordering policy. In only one test case was the EF model's policy better - test period 3 of SKU5. In one test case the Best Stock model had the best result - test case 2 of SKU2, possibly because the forecast had poor accuracy. From both the total results and the sub-results it seems that the extra information that the PHA model obtained in the additional scenarios was beneficial enough to outweigh the fact that its just an approximation algorithm of the EF.

Lets now take a closer look at some of the more interesting test cases.



Figure 11: Inventory model simulation visualization for test period 2 of SKU1. The top 3 plots contain inventory level simulations for each model, with periods with shortage highlighted by red rectangles, the bottom plot contains the forecast of ARIMA model with prediction interval in light blue.

We start with test case 2 of SKU1. The policies generated by the PHA model are the best in most cases, however this test case is one of those in which PHA was best by the highest margin. The visualization of the inventory levels can be seen in figure 11. We can see that at the end of the testing period, the actual demand is slightly higher than the forecasted one, which resulted in multiple periods with shortages of both EF and PHA model. Nonetheless, PHA's policy has significantly less periods with shortages - likely because its solution accounts for more possible demand outcomes. The base stock model has the largest estimated cost due to big amounts of excess inventory.



Figure 12: Inventory model simulation visualization for test period 1 of SKU5. The top 3 plots contain inventory level simulations for each model, with periods with shortage highlighted by red rectangles, the bottom plot contains the forecast of ARIMA model with prediction interval in light blue.

We continue with test case 1 for SKU5, as the estimated total cost difference between the Newsvendor based models and the base stock model was highest in this case. PHA model's policy estimated cost was lower by 33% than that of the base stock model. An inventory simulation of this test case can be seen in figure 12. The large difference between the model's is caused by the accuracy of the forecast (MAPE was 7.7%). As the base stock model orders the amount equal to the top of the demand prediction interval, it has large amounts of excess inventory in this case. The EF model's policy caused some shortages at the end of the testing period - possibly because the forecasted demand was lower. PHA's policy seems to cover this better, which means it achieved the best result.



Figure 13: Inventory model simulation visualization for test period 2 of SKU2. The top 3 plots contain inventory level simulations for each model, with periods with shortage highlighted by red rectangles, the bottom plot contains the forecast of ARIMA model with prediction interval in light blue.

The next inventory simulation we show in full detail is again for SKU2, this time for test period 2 (figure 13. In this case the base stock model's policy was better than PHA by almost 3%. For this case, the forecast expected smaller demand (MAPE was almost 21%). However the top of the prediction interval, which is used as the order amount by the base stock model, aligns with the actual demand quite well for the most part. But during some days in the test period, the demand was so much higher than the expected one that even the base stock model's policy had shortages. Still the shortage costs of base stock model were much lower than that of other models.

The PHA's more scenarios and its ability to finish processing in the 20 minute time limit (the EF model did not finish in the time limit), have probably helped it to achieve significantly better result than EF, but its estimated costs were still significantly higher then those of base stock model.



Figure 14: Inventory model simulation visualization for test period 2 of SKU5. The top 3 plots contain inventory level simulations for each model, with periods with shortage highlighted by red rectangles, the bottom plot contains the forecast of ARIMA model with prediction interval in light blue.

The only test case, in which the EF's policy was better than that of PHA was in case of test case 2 of of SKU5 (visualization in figure 14). However, their policies and estimated costs are almost identical anyway. Both models achieved zero shortage costs, but the PHA model had more excess inventory - its inventory line in the subplot is clearly a little higher. The base stock model had even larger amounts of excess inventory, which resulted in much higher estimated costs.

			Estimated shortage costs					
		MAPE	PHA	EF	Base Stock			
	Period 1	8.54	$2.0 \cdot 10^4$	$7.9 \cdot 10^4$	0			
SKU1	Period 2	10.17	$4.6 \cdot 10^{4}$	$1.2 \cdot 10^{5}$	0			
	Period 3	10.91	$9.5\cdot 10^4$	$1.5 \cdot 10^5 *$	0			
	Period 1	10.96	$3.1 \cdot 10^{4}$	$8.6 \cdot 10^4 *$	0			
SKU2	Period 2	20.67	$2.4 \cdot 10^{5}$	$3.0 \cdot 10^{5} *$	$2.4 \cdot 10^4$			
	Period 3	23.08	$7.8 \cdot 10^{3}$	$10.0 \cdot 10^3 *$	0			
	Period 1	7.23	$7.3 \cdot 10^4$	$1.0 \cdot 10^{5} *$	0			
SKU3	Period 2	13.10	$2.5 \cdot 10^5$	$3.0 \cdot 10^{5} *$	0			
	Period 3	8.66	$1.6 \cdot 10^{5}$	$2.1 \cdot 10^{5} *$	0			
	Period 1	10.68	$1.6 \cdot 10^{5}$	$1.8\cdot 10^5$	0			
SKU4	Period 2	11.89	$2.1 \cdot 10^5$	$2.3 \cdot 10^5$	0			
	Period 3	11.28	$2.7 \cdot 10^5$	$2.9 \cdot 10^5$	$1.1 \cdot 10^{4}$			
	Period 1	7.68	$6.0 \cdot 10^{3}$	$5.4 \cdot 10^{4}$	0			
SKU5	Period 2	16.25	0	0	0			
	Period 3	8.32	$6.3\cdot 10^4$	$1.3 \cdot 10^5$	0			
		Sum	$1.6 \cdot 10^{6}$	$2.2 \cdot 10^{6}$	$3.5 \cdot 10^4$			

Table 9: Estimated shortage costs evaluation

5.4.4 Shortage costs evaluation

We have already provided commentary on the overall performance of the models including shortage costs. Nonetheless the value of shortage costs is not just monetary, as it can cause customer dissatisfaction, so we provide another look at the models, this time with focus on shortage costs.

Table 9 contains the estimated shortage costs obtained from the inventory simulations. It can be seen that the Base Stock model has the smallest shortage costs overall , which is as expected, because low shortage of goods is the only objective of the model. If we look at the columns with results of the PHA and EF models, we can see that the PHA model achieved smaller shortage costs. Therefore we can state, that it has the potential to cause smaller dissatisfaction than the EF model.

		Running time [s]					
		PHA	EF	Base Stock			
SKU1	Period 1	256	335	0			
	Period 2	287	283	0			
	Period 3	210	1290	0			
	Period 1	332	1200	0			
SKU2	Period 2	258	1201	0			
	Period 3	284	1200	0			
	Period 1	652	1200	0			
SKU3	Period 2	602	1201	0			
	Period 3	669	1201	0			
	Period 1	240	978	0			
SKU4	Period 2	506	493	0			
	Period 3	500	900	0			
	Period 1	161	534	0			
SKU5	Period 2	165	386	0			
	Period 3	168	614	0			

Table 10: Model running time evaluation

5.5 Running time evaluation

Until now, we have focused on the estimated costs of the generated policies. However another important aspect is the running time of the models, as an inventory might contain thousand of products. Note that we have set a time limit of 1200 seconds for the inventory policy generation, which stopped the Extensive form model before completely finishing in 7 cases. We also won't discuss the running time of the Base Stock model, as it provided the results in an instant in all cases.

As we can see in table 10, the PHA model was significantly faster than EF in 13 of 15 cases. Its longest test case running time was 669 seconds, however as the algorithm can easily be parallelized, this could be dramatically improved. Even if the running times of EF model were on average significantly longer, it did not provide better results than PHA, except of one test case. The worse time effectiveness of EF meant it had to use smaller count of demand scenarios, and in 7 test cases the EF did not even finish in the time limit anyway.

If we consider that a single powerful machine could theoretically run the EF algorithm for 8 or 16 products at once, an ordering policy for inventory of 1000 products could be generated in roughly 10 hours. With the PHA algorithm, it could finish in 5 hours or even less. Or we could feed even higher count of scenarios in the PHA model, which could further improve its results.

5.5.1 Inventory control evaluation conclusion

In this subsection we have evaluated three inventory models, first model was the extensive form of our proposed extension of Newsvendor model, which worked with 25 demand realization scenarios in each test case, the second one was the a decomposed version of the previous model, which used the Progressive Hedging Algorithm and 250 demand scenarios. The last model was the simple Base Stock model.

Across 3 testing periods for 5 SKUs, forming 15 possibly real world test cases, the best on average model was the PHA base model, as it generated inventory policies with the smallest cost. Its extensive form provided worse results, which were obtained in significantly longer running times.

The Base Stock model was the worst performing one, as it does not consider price changes and also due to the relatively accurate forecasts, its safety inventory level was not needed in most cases.

5.6 Evaluation conclusion

In this section, we have evaluated both the demand forecasting models and the inventory control models. For the purposes of evaluation, we have extracted 15 possibly real world test cases from the data provided by the supervisor.

Both the exponential smoothing and ARIMA forecasting models showed reasonable forecasting performance given the random nature of the testing data. However, during our in depth analysis we have found out that the "fitting" procedure of both models could be further improved. Possible future work could therefore focus on the fitting phase, or on forecasting with external factors.

In the part dedicated to inventory control, we compared the performance of two version of our extended Newsvendor inventory model. This model has the advantage of large variability due to its mixed integer linear programming formulation, but solving the program requires large amounts of computation time. Therefore we have compared the extensive formulation of the model and its approximation form based on Progressive Hedging Algorithm. The extensive form of the model had to use less demand information in order to be able to finish in reasonable computing time. The approximation form had the advantage of more demand information.

The PHA model performed significantly better on average than the extensive form, and it required significantly less computation time. Both models performed better than the the third model - the simplistic Base Stock model. In future work, we could focus on development of some combination of these models. The decomposition formulation could be used to presolve parts of the problem, until a smaller problem is obtained, which could be later solved in the extensive form.

6 Conclusion

The task of this thesis was to study the problems of demand prediction and inventory optimization, and to employ this knowledge to create an automated inventory control model. After a thorough analysis we have presented a set of time series forecasting and inventory control approaches, which together form an inventory control pipeline capable of generating possibly optimal goods ordering policies.

The demand prediction part of the thesis contains a survey of time series forecasting problematic and available methods. The prediction part of the inventory control pipeline was then implemented to include an extended exponential smoothing forecasting method and a seasonal ARIMA model. Both methods have proven its worth in many previous research works and results of our evaluation confirm they are suitable for our task as well.

The inventory control part of the thesis consists of an introduction to the inventory control problematic, and contains a review of important concepts of stochastic programming, which are highly useful for inventory models that assume demand uncertainty. We then proceed by creating an extension of the stochastic multi-stage Newsvendor inventory model, that can generate ordering policies for many time periods ahead. This model considers fixed and per unit ordering costs, shortage costs, holding costs and goods durability. All costs can be time-dependent and the model can be possibly extended even further, as it has a flexible mixed integer linear program formulation. We then suggested to solve the problem using the progressive hedging algorithm to improve the model's efficiency and significantly enhance its ability to hedge its inventory policies against more demand outcomes.

We have performed a two stage, in depth evaluation of the proposed methods, using real world data. In the first stage, which was dedicated to evaluation of the forecasting methods, the seasonal ARIMA model proved its superiority and robustness. Its forecasts were always either the most accurate, or very close to the best forecast, and its performance was the most stable. Its residuals showed that it fitted the provided data well, as its errors showed almost no correlation, and normality. The second stage evaluated the proposed inventory models by estimating costs of their inventory policies and compared the models from the customer satisfaction perspective. We also examined the trade of between solving an exact form of the inventory model with limited amount of demand scenario samples, and solving its approximation with significantly more demand scenarios. The results showed that it is sufficient to use reasonable number of scenarios to achieve efficient inventory control policy, and it is preferable to use the approximation approach. The approximation form provided the best inventory policies due to the significantly larger amount of demand scenarios it could handle, and its running times were much lower than those of the exact form.

6.1 Future work

We can name multiple examples of possible improvements and topics for possible future research.

The demand prediction module's accuracy could be improved by including a forecasting method that uses external factors in addition to the historical time series data, such as SARIMAX.

It would be interesting to focus on the scenario generation procedure as well. For example we could test other methods of the forecast prediction interval estimation, such as bootstrapping the residuals.

The inventory module could be improved by combining the decomposition approach and the extensive form solution approach. The decomposition model could be used to generate a subset of the inventory decisions, until a problem of manageable size for the extensive form is obtained. The extensive form could then generate optimal decisions for the remaining problem.

7 Appendix

				1				SKU		I							
Me		KU 5			KU 4			KU 3			KU 2			KU 1			
UE		Period			Period		Period		Period								
	3	2	1	ω	2	1	ω	2	1	ω	2	1	ω	2	1		
11.768	17.712	15.991	9.438	7.311	11.662	19.258	8.532	10.744	10.072	14.169	20.644	6.287	9.327	9.459	5.918	MAPE	Smoc
0.095	0.127	0.088	0.067	0.081	0.122	0.164	0.094	0.112	0.076	0.097	0.156	0.032	0.087	0.079	0.042	RMSE	othing
11.962	8.320	16.248	7.677	11.276	11.891	10.685	8.662	13.106	7.227	23.078	20.671	10.964	10.917	10.176	8.536	MAPE	ARI
0.095	0.061	0.087	0.051	0.118	0.125	0.108	0.091	0.133	0.060	0.130	0.156	0.064	0.095	0.083	0.062	RMSE	MA
12.586	9.686	15.372	10.245	7.139	10.561	9.692	8.129	12.630	6.954	13.094	31.996	10.455	14.426	17.506	10.907	MAPE	Prop
0.101	0.071	0.089	0.066	0.070	0.107	0.106	0.087	0.126	0.052	0.093	0.229	0.068	0.129	0.139	0.077	RMSE	ohet

Table 11: Demand forecasting evaluation results



Figure 15: Results of first product class forecasting evaluation



Figure 16: Results of second product class forecasting evaluation



Figure 17: Results of third product class forecasting evaluation



Figure 18: Results of fourth product class forecasting evaluation



Figure 19: Results of fifth product class forecasting evaluation

References

- [1] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2014.
- [2] Robert G Brown. Exponential smoothing for predicting demand. In Operations Research, volume 5, pages 145–145. INST OPERATIONS RESEARCH MANAGEMENT SCIENCES 901 ELKRIDGE LAND-ING RD, STE 400, LINTHICUM HTS, MD 21090-2909, 1957.
- [3] Nikolaos Askitas and Klaus F Zimmermann. Google econometrics and unemployment forecasting. Applied Economics Quarterly, 55(2):107– 120, 2009.
- [4] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control.* John Wiley & Sons, 2015.
- [5] Nari Sivanandam Arunraj, Diane Ahrens, and Michael Fernandes. Application of sarimax model to forecast daily sales in food retail industry. *International Journal of Operations Research and Information Systems* (IJORIS), 7(2):1–21, 2016.
- [6] Nikolay Laptev, Jason Yosinski, Li Erran Li, and Slawek Smyl. Timeseries extreme event forecasting with neural networks at uber. In *In*ternational Conference on Machine Learning, 2017.
- [7] Gianpaolo Ghiani, Gilbert Laporte, and Roberto Musmanno. Introduction to logistics systems planning and control. John Wiley & Sons, 2004.
- [8] Leroy B Schwarz. The economic order-quantity (eoq) model. In *Building Intuition*, pages 135–154. Springer, 2008.
- [9] Annie Chen. Inventory management, part 2. available at https://ocw.mit.edu/courses/sloan-school-of-management/ 15-772j-d-lab-supply-chains-fall-2014/lecture-notes/ MIT15_772JF14_Lec14.pdf, 2014. Accessed: 2018-12-05.
- [10] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. Lectures on stochastic programming: modeling and theory. SIAM, 2009.
- [11] Alexander Shapiro and Andy Philpott. A tutorial on stochastic programming. Manuscript available at www2.isye.gatech.edu/ ashapiro/publications.html, 2007. Accessed: 2018—12—05.
- [12] Stein-Erik Fleten and Trine Krogh Kristoffersen. Short-term hydropower production planning by stochastic programming. *Computers* & Operations Research, 35(8):2656-2671, 2008.

- [13] Mehdi Lamiri, Frédéric Grimaud, and Xiaolan Xie. Optimization methods for a stochastic surgery planning problem. *International Journal* of Production Economics, 120(2):400–410, 2009.
- [14] Andreas Eisenblätter and Jonas Schweiger. Multistage stochastic programming in strategic telecommunication network planning. *Computational Management Science*, 9(3):303–321, 2012.
- [15] Jacques F Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, 4(1):238–252, 1962.
- [16] R Tyrrell Rockafellar and Roger J-B Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of operations* research, 16(1):119–147, 1991.
- [17] Jan G De Gooijer and Rob J Hyndman. 25 years of time series forecasting. International journal of forecasting, 22(3):443–473, 2006.
- [18] RMJ Heuts and JHJM Bronckers. Forecasting the dutch heavy truck market: A multivariate approach. *International Journal of Forecasting*, 4(1):57–79, 1988.
- [19] Allan P Layton, Lorraine V Defris, and Ben Zehnwirth. An international comparison of economic leading indicators of telecommunications traffic. *International Journal of Forecasting*, 2(4):413–425, 1986.
- [20] Guoqiang Zhang, B Eddy Patuwo, and Michael Y Hu. Forecasting with artificial neural networks:: The state of the art. *International journal of forecasting*, 14(1):35–62, 1998.
- [21] G Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.
- [22] Charles C Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International journal of forecasting*, 20(1):5–10, 2004.
- [23] Peter R Winters. Forecasting sales by exponentially weighted moving averages. *Management science*, 6(3):324–342, 1960.
- [24] Ashraf A Shahin. Using multiple seasonal holt-winters exponential smoothing to predict cloud resource provisioning. *arXiv preprint* arXiv:1701.03296, 2017.
- [25] Lingxue Zhu and Nikolay Laptev. Deep and confident prediction for time series at uber. In *Data Mining Workshops (ICDMW)*, 2017 IEEE International Conference on, pages 103–110. IEEE, 2017.

- [26] Bernardo Pérez Orozco, Gabriele Abbati, and Stephen Roberts. Mordred: Memory-based ordinal regression deep neural networks for time series forecasting. arXiv preprint arXiv:1803.09704, 2018.
- [27] Herbert Scarf. A min max solution of an inventory problem. Studies in the mathematical theory of inventory and production, 1958.
- [28] Tatsiana Levina, Yuri Levin, Jeff McGill, Mikhail Nediak, and Vladimir Vovk. Weak aggregating algorithm for the distribution-free perishable inventory problem. Operations Research Letters, 38(6):516–521, 2010.
- [29] Afshin Oroojlooyjadid, Lawrence Snyder, and Martin Takáč. Applying deep learning to the newsvendor problem. arXiv preprint arXiv:1607.02177, 2016.
- [30] Andrew Ning, Henry CW Lau, Yi Zhao, and TT Wong. Fulfillment of retailer demand by using the mdl-optimal neural network prediction and decision policy. *IEEE Transactions on Industrial Informatics*, 5(4):495–506, 2009.
- [31] Anna-Lena Sachs. The data-driven newsvendor with censored demand observations. In *Retail Analytics*, pages 35–56. Springer, 2015.
- [32] J Sudhir Ryan Daniel and Chandrasekharan Rajendran. A simulationbased genetic algorithm for inventory optimization in a serial supply chain. *International Transactions in Operational Research*, 12(1):101– 127, 2005.
- [33] Gitae Kim, Kan Wu, and Edward Huang. Optimal inventory control in a multi-period newsvendor problem with non-stationary demand. *Advanced Engineering Informatics*, 29(1):139–145, 2015.
- [34] Retsef Levi, Robin O Roundy, and David B Shmoys. Provably nearoptimal sampling-based policies for stochastic inventory control models. *Mathematics of Operations Research*, 32(4):821–839, 2007.
- [35] Ronald K Pearson, Yrjö Neuvo, Jaakko Astola, and Moncef Gabbouj. Generalized hampel filters. EURASIP Journal on Advances in Signal Processing, 2016(1):87, 2016.
- [36] Roberto Battiti and Francesco Masulli. Bfgs optimization for faster and automated supervised learning. In *International neural network* conference, pages 757–760. Springer, 1990.
- [37] Inc. Gurobi Optimization. Gurobi modeling 2. http://www.gurobi. com/pdfs/user-events/2017-frankfurt/Modeling-2.pdf, 2016. Accessed: 2018-12-05.

- [38] Dinakar Gade, Gabriel Hackebeil, Sarah M Ryan, Jean-Paul Watson, Roger J-B Wets, and David L Woodruff. Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs. *Mathematical Programming*, 157(1):47–67, 2016.
- [39] Jean-Paul Watson, David L Woodruff, and David R Strip. Progressive hedging innovations for a class of stochastic resource allocation problems. 2008.
- [40] Sean J Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 2017.
List of Figures

1	Decomposition of airline passengers time series	5
2	Scenario tree example	13
3	An example of the provided time-series data	21
4	Experiment with extensive form running time	27
5	Results of first product class - SKU1 forecasting evaluation .	32
6	Results of second product class - SKU2 forecasting evaluation	34
$\overline{7}$	Results of third product class - SKU3 forecasting evaluation .	35
8	Results of fourth product class - SKU4 forecasting evaluation	36
9	Results of fifth product class - SKU5 forecasting evaluation .	38
10	Autocorrelation plots of residuals of ARIMA and Exponential	
	smoothing models selected for forecasting of SKU1 in first	
	testing period	39
11	Inventory model simulation visualization for test period 2 of	
	SKU1	46
12	Inventory model simulation visualization for test period 1 of	
	SKU5	47
13	Inventory model simulation visualization for test period 2 of	
	SKU2	48
14	Inventory model simulation visualization for test period 2 of	
	SKU5	49
15	Results of first product class forecasting evaluation	57
16	Results of second product class forecasting evaluation	58
17	Results of third product class forecasting evaluation	59
18	Results of fourth product class forecasting evaluation	60
19	Results of fifth product class forecasting evaluation	61

List of Tables

1	Results of SKU1 forecasting	32
2	Results of SKU2 forecasting	34
3	Results of SKU3 forecasting	35
4	Results of SKU4 forecasting	36
5	Results of SKU5 forecasting	37
6	Ljung-Box test p-values for $lag = 14$ of residuals of models	
	used for first period forecasting	40
7	Shapiro-Wilk test p-values of residuals of models used for first	
	period forecasting	41
8	Estimated total costs evaluation	45
9	Estimated shortage costs evaluation	50
10	Model running time evaluation	51
11	Demand forecasting evaluation results	56