Wageningen University - Department of Social Sciences

Operations Research and Logistics

MSc Thesis Report

Inventory optimization of a multi-echelon supply chain with perishable products with a fixed lifetime

Abstract

As we progress up the supply chain from the retailer to the producer, there is an increase in the magnification of order fluctuations. This bullwhip effect leads to instability in the supply chain. Getting proper coordination between supply chain partners can be obtained by using a multi-echelon inventory management approach. Using multi-echelon inventory optimization, the inventory levels throughout the supply chain are decreased. In this research, a single-product perishable two-echelon inventory optimization problem is formulated as a mixed integer linear programming (MILP) model. The model is solved with a branch and bound (B&B) algorithm using Gurobi in Python. Time-restricted experiments are presented in this research with a focus on evaluating the performance of the algorithm by considering optimality gaps and time limits. The findings highlighted the importance of setting an appropriate optimality gap / time limit to balance the solution quality and computational time. Thereby, some additional experiments are conducted aimed to investigate the behaviour of the model under different parameters and demand patterns. The experiments provided valuable insights into the behaviour of the model under different conditions. The results demonstrated the importance of considering demand levels, fixed order costs, the trade-off between lost sales costs and outdate costs, and fixed lifetime when optimizing supply chains. These findings contribute to a better understanding of the model's performance and can guide decision-making in supply chain management practices.

Keywords: multi-echelon, inventory optimization, perishability, fixed lifetime, MILP

July 2023

Student	Jelle Loosveld
Registration number	1018122
MSc program	Management, Economics & Consumer Studies
Specialisation	Business Studies
Supervisor(s)	dr. KGJ. (Karin) Pauls-Worm
Examiner/2nd supervisor	dr. P. (Peter) Kirst
Thesis code	ORL-80436
Word count	11.999

WAGENINGEN UNIVERSITY & RESEARCH

1. Introduction

A retailer sells an average of ten steaks per day. Due to an increase in temperature, the sales rise to an average of 20 steaks per day. To meet the new demand of the customers, the retailer makes an order of 25 steaks per day at the distribution centre (DC) to decrease the chance for a stockout. These 5 extra products are called the safety stock. Safety stock is the additional stock a company holds to reduce the risk of having a stockout caused by uncertainties in supply and demand. While the DC gets an order of 25 products from the retailer, the DC has noticed the increase in order volume and subsequently increases their forecast demand as well. Because of the high service level requirements for the DC, the DC places an order of 32 steaks at the producer. The producer also notices this increase in volume and starts to produce more to be able to deliver the quantities ordered from the DC. The producer puts a date on the product which represents the end date of the product. Such a product is seen as a perishable product with a fixed lifetime. The order of 32 steaks from the DC results in the production of 40 steaks at the producer. Following, the average sales of 20 steaks per day at the retailer causes the producer to produce 40 steaks to the DC, generating a total safety stock in the supply chain of 20. All the parties in the supply chain try to protect themselves from the changing demand which can lead to an accumulation of products at the producer and throughout the whole supply chain.

This problem is called the bullwhip effect. The bullwhip effect can be described as the amplification in the fluctuation of order, as we move upwards in a supply chain from the retailer to the producer, according to Chopra (2019). The bullwhip effect leads to instability in the supply chain. A small change in the forecasted demand by a retailer can grow out to large changes in the orders received by producers. Multiple causes can affect the bullwhip, following: demand forecast, order batching, inventory policy, lack of transparency, number of echelons, and more. Simon (1952) and Forrester (1958) were the first ones that studied the bullwhip effect. Forrester has studied the bullwhip effect through simulation and put the term ''demand amplification'' to it. The bullwhip effect has an influence on the supply chain in multiple ways. But the most common problem is excessive inventory throughout the whole supply chain (Sun & Ren, 2005). This is majorly caused by a lack of coordination between supply chain partners, which is also visible in the steak supply chain example. This leads to improper, inconsistent, and insufficient information that results in order variability. Thus, proper coordination between supply chain partners can mitigate the bullwhip effect (Bhattacharya & Bandyopadhyay, 2010).

Getting proper coordination between supply chain partners can be obtained by using a multi-echelon inventory management approach. Multi-echelon inventory optimization is a method that combines the partners in the supply chain to help the companies optimize the inventory levels throughout the whole chain. From the research of Ekanayake et al. (2016), it is concluded that when using a multi-echelon system, inventory levels are lower, while maintaining higher fill rates for the whole supply chain, in comparison to single-echelon inventory optimization. When using perishable products in the supply chain, proper coordination between supply chain partners is even more necessary. Having perishable items, inventory management results in a trade-off of shortages and lost sales against wastage. Having an excessive inventory of perishable products could mean that products are decaying and need to be disposed of. Having too little inventory causes products to be out of stock, resulting in lost sales, and a lower service level.

This paper consists of the following sections: In section 2 a systemic literature review is applied to find existing research. In section 3, the problem description and model formulation are described. In section 4, the results of the model with different experiments are presented. Section 5 presents a discussion. The paper ends with section 6, a conclusion, and recommendations on future research are given.

2. Literature review

In this section, a systematic literature review is conducted to explore aspects of the multi-echelon supply chain. The purpose of this literature review is to gain a understanding of the subject by examining research and identifying key findings.

To ensure a thorough investigation, specific keywords and search criteria are employed during the literature search. The full explanation is presented in Appendix A.

2.1. Multi-echelon supply chain

Considering a multi-echelon inventory model, literature shows that many echelons are present in inventory systems. From de Kok et al. (2018), where an extensive literature research of multi-echelon inventory management under uncertain demand until the end of 2016 is reviewed, it is found that the majority of papers uses two-echelon systems. Some consider n-echelon systems and only a few uses a three-echelon system. The majority of papers on multi-echelon inventory models is a two-echelon system, this is explained by the increase of complexity when extending the model from a two to an n-echelon system, except for papers with serial structures (de Kok et al., 2018; Chaudhary et al., 2018; Gholami-Zanjani et al., 2021).

The first papers researching multi-echelon inventory systems all had a serial structure (Simpson, 1958; Clark & Scarf, 1960). In practice, serial systems do not exist and so research on divergent and general structures is needed (de Kok et al., 2018). Hill et al., (2007), Andersson & Melchiors, (2001), Kanchanasuntorn & Techanitisawad, (2006), all use a one warehouse, multiple retailer, inventory system.

Within supply chains, customers can react differently to stockouts. Two options are backordering of demand and lost sales. Backordering of demand is popular in papers because of tractability and optimal ordering policies are only known under the assumption of backordering (de Kok et al., 2018). Lost sales models are popular within perishability research, because it is more representative to model stockouts as lost sales if the retailer is in a competitive market and consumers can easily choose a substitute (Andersson & Melchiors, 2001). Research with lost sales included in the model is done by Kouki et al. (2015), Hill et al. (2007), Tsai et al. (2022)

Including perishability into a multi-echelon supply chain increases the complexity of the model (Suryawanshi & Hsein, 2010; Kanchanasuntorn & Techanitisawad, 2006; Kouki et al., 2015). A majority of the research on perishable inventory management focuses on how much inventory is needed to balance supply and demand (Herbon, 2017). Perishable inventory management is conducted by multiple researchers. Kouki et al. (2014) considers an order up to level (T,S) policy for perishable items with random lifetime. Kouki et al. (2015) consider a continuous review (r, Q) policy, where unfulfilled demand is lost in a perishable inventory system with stochastic demand, constant lifetime and constant lead time, but both in a single-echelon environment. Research on perishability in multi-echelon supply chains is done by Ali et al., (2021) and Jaigirdar et al., (2022). For perishable supply chains the FIFO issuing policy is used in almost all models (Lee, 2006; Chen & Sapra, 2013).

Some articles which include a multi-echelon supply chain are: the article of Mitra (2009), where a deterministic as well as a stochastic model is developed for a two-echelon inventory system with returns. In this research the objective was to determine the values of the policy variables which minimizes the total costs. Also, the paper of Cohen & Lee (1988), where a series of linked, approximate sub models and a heuristic optimization procedure are introduced to solve an integrated production-distribution system. The paper presents a model framework designed for forecasting a firm's performance in relation to: the cost of the supply chain, the level of service that is provided to its customers, and the flexibility of the system. The article of Noordhoek et al., (2018), a simulation-optimization approach to optimize (s,S) inventory policy is developed. They compared the performance of a Nested Bisection Search (NBS) and a novel Scatter Search (SS) metaheuristic using 1280 instances from literature and derived managerial implications from a real-life case.

2.2. Order policy

There are different types of order policies but two policies are widely used, namely periodic and continuous review systems. Most of the research in inventory management assumes inventory levels are reviewed periodically, which means that the inventory on hand is known only at discrete points in time (Chaudhary et al., 2018). In the research of Kanchanasuntorn & Techanitisawad (2006), a (R, S_i) policy is used for the retailers in the chain with identical review interval R and different inventory levels S_i. For the warehouse a (R, s, S) policy is applied. Duong et al. (2015) applies a (R, S_{ji}), in a single vendor, multi-retailer model, where S is the maximum inventory level of product j^{th} at i^{th} vendor, and retailers. In the research of Shang and Zhou (2010), a (r, nQ, T), also known as a (R, s, nQ), policy is applied for a serial inventory system, where the objective is to find the policy parameters so that the average total cost per period is minimized. Broekmeulen & Van Donselaar (2009) came up with a modified (R, s, nQ)-policy for perishable products, called the EWA-policy. This policy takes into account the full age distribution of the inventory when determining order quantities. Using a numerical experiment for a lost sales system, Broekmeulen and van Donselaar show that the policy decreases the costs over all experiments and performs better than a traditional (R, s, nQ)-policy, mostly when dealt with shelf life less than 10 days. Van Donselaar & Broekmeulen (2011) uses the modified (R, s, nQ)-replenishment policy in a lost sales environment. The paper approximates the customer service level for a perishable item in a stochastic lost sales environment.

The alternative policy is a continuous review policy, within this policy the inventory level is tracked continuously. If inventory levels reach the threshold, a particular amount of products is ordered with a base-stock policy, mostly a (s, S) or (r, Q) policy. In the article of Cohen et al (1990), the authors present IBM's Optimizer as a powerful multi-echelon inventory system designed for managing service logistics. It outlines the system's features, benefits, and real-world applications, highlighting its ability to optimize inventory levels and enhance supply chain performance using a (s, S) review policy (De Kok et al., 2018). Federgruen and Zheng (1992) present an algorithm for efficiently calculating optimal (r, Q) policies in continuous review stochastic inventory systems, where demand and lead times are uncertain and follow probabilistic distributions. The objective is to find the (r, Q) policy that minimizes the total expected cost, which includes holding costs, ordering costs, and stockout costs.

2.3. Solving method

Multi-echelon inventory problems are solved over the years by multiple methods. Mathematical programming is one of the techniques and is commonly used in the literature of perishable inventory models. Mixed integer linear programming is used by Claassen et al. (2016) and Sel et al. (2015). Pinkus (1971) addresses the challenge of designing inventory systems that involve multiple products and multiple echelons. The objective is to determine the optimal inventory levels and allocation policies that minimize the total system cost while satisfying customer demand. The article introduces a branch-and-bound algorithm, a commonly used optimization technique, to solve this complex inventory design problem. The article of Lourenção et al. (2017) introduces a mixed-integer nonlinear model for optimizing multiproduct inventory systems. The authors propose a combination of interior point and branch-and-bound methods to solve the complex optimization problem.

Mathematical programming can also be used together with other techniques, van der Heijden et al. (1999), Andersson & Marklund (2000), Forsberg (1997) consider a two-echelon inventory system where mathematical programming and simulation are used together.

In the article of Cohen & Moon (1991), an integrated plant loading model with economies of scale and scope is solved using a solution algorithm. A mixed integer linear programming model is formulated and decomposed into an integer master problem and a linear problem and solved with the Benders decomposition

However, inventory problems including multi-echelon and perishability are complex. Using a mathematical model has limitations in solving these problems (Xu et al., 2019; Köchel & Nieländer, 2005). Various scholars have used simulation-based optimization to solve the problem (Xu et al., 2019; Attar et al., 2016). But the computational time for optimization models can become large. This large computational time is identified by the study of Lagodimos et al. (2012) as the main cause for the lack of analytical results of periodic review polices and the impractical use in real life environments due to the high run-times. Metaheuristic approaches are appropriate methods for solving multi-echelon perishability problems (Kanchanasuntorn & Techanitisawad, 2006; Lagodimos et al., 2012; Christou et al., 2020). These methods can reduce the computational time and thereby obtain near-optimal results. Among existing meta-heuristics, Tabu Search (TS), Genetic Algorithms (GA), Simulated Annealing (SA) and Evolution Strategies (ES) have received great attention in the field of optimization problems. Daniel and Rajendran (2005) applied a simulation-based genetic algorithm for inventory optimization in a serial supply chain, where the base stock parameters are optimized. Radhakrishnan et al. (2009) use a genetic algorithm-based inventory optimization analysis to minimize the total supply chain costs to determine the inventory level held at different stages in the supply chain. Jackson et al. (2018), describes an eventual combination of discrete-event simulation and genetic algorithm to define the optimal inventory policy in stochastic multi-product inventory systems. Alrefaei & Diabat (2009) present a simulated annealing algorithm for solving multi-objective simulation optimization problems. La Fata and Passannanti (2017) use a simulated annealing-based algorithm combined with a Monte Carlo simulation module as a resolution approach for the joint optimization of production/inventory polices. Datta and Regis (2016), proposes a surrogate-assisted evolution strategy (ES) that can be used for constrained multi-objective optimization of expensive black-box objective functions subject to expensive black-box inequality constraints.

The existing literature on multi-echelon inventory management focuses on several aspects. Firstly, the majority of papers assume backordering of demand, which is tractable and allows for optimal ordering policies. However, the consideration of lost sales models is relatively limited. Secondly, the inclusion of perishability in multi-echelon supply chains is recognized as a complex problem, and so limited research on perishability in multi-echelon supply chains is conducted. Finally, while multiple solving methods such as mathematical programming, simulation-based optimization, and metaheuristic approaches have been employed. There are limitations in solving complex inventory problems, particularly those involving multi-echelon systems and perishability. Resulting in the need for efficient and practical solving methods, with reduced computational time and near-optimal results.

3. Problem description and model formulation

Consider a single-product perishable two-echelon inventory optimization problem, where the total costs are minimized for a finite time horizon of T periods. Periods can be days, weeks or months, but in this study periods are measured in days. The network consists of one warehouse j and multiple retailers I. The perishable products are characterized by a finite shelf life of M periods, which denotes the maximum duration for which they remain consumable. After the expiration of M periods, the products become unsuitable for consumption and are removed from inventory. The index denoting the age of the item is $m = 1, 2, \dots, M$. Given the retailer's preference for receiving products that have a considerable usable lifespan, the warehouses are constrained to dispatch items with an age not exceeding $M - v$. Where v is the minimum amount of periods over when the items are delivered at the retailer. Variables It_{jmt} and It_{imt} denote the inventory level of the item with age m at the end of period t in the warehouse and retailers respectively. Variables $It_{j,M-v,t}$ and It_{iMt} denote the inventory level at the end of period t with the maximum age level M and is thus considered waste. Items are delivered from an external supplier, who can always deliver, to the warehouse. Inventory is reviewed at the end of each period, wherein a replenishment decision is made if the inventory level falls below the predetermined reorder level s. Specifically, the replenishment order is placed at the end of period t and is scheduled for delivery at the start of the subsequent period $t + 1$, with a zero lead time assumption for both the warehouse j and retailers I. Items are delivered at the beginning of the period t at the warehouse and have age $m = 1$ at the end of period t. Because of simplicity, a big penalty is awarded to having lost sales. In this way, the probability of a sufficient service level is increased.

The retailers face a Poisson demand distribution. Demand is known for each time period t for the finite time horizon T, meaning demand is deterministic and non-stationary. No backlogging is allowed at the retailers and warehouse. Customers that face stockouts at the retailers will become lost sales. Stockouts that occur at the warehouse are also considered lost sales. All products are issued using a FIFO-policy at the warehouse and the retailers. Inventory is controlled by a periodic (R, s, Q) policy. Every R periods the inventory position is monitored, in this case R is set to 1. When the inventory level is lower than reorder level s, an order is placed of Q items.

Units held in stock in both the warehouse and retailers will incur holding costs per unit of product per unit of time. For every unit bought, purchase costs per unit are incurred. For every order, fixed order costs are incurred for both the warehouse and retailers. The review costs for periodically reviewing the inventory are assumed to be neglectable. When units are not used by demand during their lifetime and are disposed of, an outdate cost is incurred, and when units are not in stock during demand, the cost of a lost sale is charged. All costs remain constant during the time horizon. Throughout this paper, we use the notations presented below.

The objective of this study is to determine the variable Q of the periodic (R, s, Q) policy that minimizes the total costs.

Parameters:

$i = \text{ index denoting retailers}, i = 1, 2, \dots, I$

$I = \text{maximum number of retailers}$

$j = \text{index denoting warehouse}, j = 1$

$t = \text{index denoting period}, t = 1, 2, \dots, T$

$T = \text{maximum number of periods of the finite time horizon}$

$m = \text{index denoting the age of the item}, m = 1, 2, \dots, M$

$M = \text{maximum age of the product (maximum shelf life)}$

$d_{it} = \text{stochastic demand per retailer } i, \text{during period } t, \text{poission distributed}$

$c_i = \text{purchase cost per unit for retailer } i$

$c_j = \text{purchase cost per unit for warehouse } j$

$w_i = \text{outdate cost per unit for retailer } i$

$w_j = \text{outdate cost per unit for warehouse } j$

$p_i = lost\ sales\ cost\ per\ unit\ for\ retailer\ i$
$p_j = lost\ sales\ cost\ per\ unit\ for\ warehouse\ j$
$h_i = holding\ cost\ per\ unit\ held\ for\ retailer\ i$
$h_j = holding\ cost\ per\ unit\ held\ for\ warehouse\ j$
$k_i = fixed\ ordering\ cost\ per\ order\ for\ retailer\ i$
$k_j = fixed\ ordering\ cost\ per\ order\ for\ warehouse\ j$
$v = minimum\ number\ of\ periods\ left\ when\ the\ items\ are\ delivered\ at\ the\ retailer$
$s_j = reorder\ level\ at\ warehouse\ j$
$s_i = reorder\ level\ at\ retailer\ i$

Variables:
$Q_{jt} = order\ quantity\ at\ warehouse\ j\ in\ period\ t$
$Q_{it} = order\ quantity\ at\ retailer\ i\ in\ period\ t$
$It_{jmt} = inventory\ level\ of\ age\ m\ at\ warehouse\ j\ at\ the\ end\ of\ period\ t$
$X_{jt} = numbers\ of\ items\ short\ at\ warehouse\ j\ in\ period\ t$
$It_{imt} = inventory\ level\ of\ age\ m\ at\ retailer\ i\ at\ the\ end\ of\ period\ t$
$X_{it} = numbers\ of\ items\ short\ at\ retailer\ i\ in\ period\ t$
$TIout_{mt} = total\ products\ of\ age\ m\ going\ out\ by\ warehouse\ j\ in\ period\ t$
$Iout_{imt} = products\ going\ out\ of\ age\ m\ by\ warehouse\ j\ in\ period\ t\ to\ retailer\ i$
$A_{imt} = the\ shortage\ of\ inventory\ at\ retailer\ i\ of\ age\ m\ in\ period\ t$
$Y_{jt} \begin{cases} 1, & if\ an\ order\ is\ placed\ by\ warehouse\ j\ in\ period\ t \\ 0, & otherwise \end{cases}$
$Y_{it} \begin{cases} 1, & if\ an\ order\ is\ placed\ by\ retailer\ i\ in\ period\ t \\ 0, & otherwise \end{cases}$

The model is formulated as a mixed-integer linear programming model, where the total costs (TC) are minimized. Costs arise by ordering products, fixed order costs and unit purchase costs, outdating of products, keeping inventory and by not satisfying demand. The objective function is therefore denoted as:

$$min\ TC = \sum_{t=1}^{T}\Big(\sum_{i=1}^{I}(c_i * Q_{it}) + \sum_{j=1}^{J}(c_j * Q_{jt}) + \sum_{i=1}^{I}(w_i * It_{iMt}) + \sum_{j=1}^{J}(w_j * It_{j,M-v,t}) + \sum_{j=1}^{J}(k_j * Y_{jt}) + \sum_{i=1}^{I}(k_i * Y_{it}) + \sum_{j=1}^{J}(h_j * \sum_{m=1}^{M-v-1} It_{jmt}) + \sum_{i=1}^{I}(h_i * \sum_{m=1}^{M-1} It_{imt}) + \sum_{i=1}^{I}(p_i * X_{it}) + \sum_{j=1}^{J}(p_j * X_{jt})\Big)$$

The model is subject to many constraints. Here, the constraints are expressed with some explanations.

<u>Inventory:</u>
Equations (1)-(2) show that the inventory level of all products with different ages m at the end of period t is the inventory level of the end of the period $t-1$ plus the order quantity minus the demand in period t. Units of age M can not be used in the next period for the retailer and of age $M-v$ for the warehouse. If the demand in period t exceeds the inventory of period t, a part of the demand is lost and there is a shortage of X_{it}, X_{jt}

$$\sum_{m=1}^{M} It_{imt} - X_{it} = \sum_{m=1}^{M-1} It_{im,t-1} + Q_{i,t-1} - d_{it} \quad i = 1,\dots,I;\ t = 1,\dots,T \tag{1}$$
$$\sum_{m=1}^{M-v} It_{jmt} - X_{jt} = \sum_{m=1}^{M-v-1} It_{jm,t-1} + Q_{j,t-1} - \sum_{i=1}^{I} Q_{it} \quad j = 1,\dots,J;\ t = 1,\dots,T \tag{2}$$

<u>FIFO:</u>
Equations (3)-(7) make sure that the inventory shipped with age $m-1$ from the warehouse is delivered at the retailer with age m.

$\sum_{i=1}^{I} Q_{it} + X_{jt} = \sum_{m=1}^{M-v} TIout_{mt} \quad t = 1,\ldots,T$ (3)
$TIout_{M-v,t} = It_{j,M-v-1,t-1} - It_{j,M-v,t} \quad t = 1,\ldots,T$ (4)
$TIout_{mt} = A_{jmt} - It_{jmt} \quad j = 1,\ldots,J;\ m = 1,\ldots,M-v-1;\ t = 1,\ldots,T$ (5)
$TIout_{mt} = \sum_{i=1}^{I} Iout_{imt} \quad m = 1,\ldots,M-v;\ t = 1,\ldots,T$ (6)
$Q_{it} = \sum_{m=1}^{M-v} Iout_{imt} \quad i = 1,\ldots,I;\ t = 1,\ldots,T$ (7)

These equations (8)-(10) keep track of the age-distribution of the items under a FIFO-issuing policy and are first used in the article of Pauls-Worm et al. (2014). Auxiliary variable A_{imt} is added to denote the shortage of inventory of age m with $m = 1,\ldots,M-1$ in period t to fulfil the demand of period t. If the value of variable A_{imt} is positive, fresher inventory is used to fulfil the demand.

$It_{i,M-1,t-1} - d_{it} = It_{iMt} - A_{i,M-1,t} \quad t = 1,\ldots,T;\ i = 1,\ldots,I$ (8)
$It_{im,t-1} + Iout_{i,m+1,t} - A_{i,m+1,t} = It_{i,m+1,t} - A_{imt} \quad t = 1,\ldots,T;\ m = 1,\ldots,M-2;\ i = 1,\ldots,I$ (9)
$Iin_{i1t} - A_{i1t} = It_{i1t} - X_{it} \quad t = 1,\ldots,T;\ i = 1,\ldots,I$ (10)

Because only one variable on the right-hand-side of equations (8)-(10) can have a positive value the following constraints are made (11)-(14). Binary variables $B(A_{imt})$ and $B(X_{it})$ are added and M_{big} is a sufficiently large number, in this case $\sum_{i=1}^{I}\sum_{t=1}^{T} d_{it}$.

$M_{big} * B(A_{imt}) \geq A_{imt} \quad t = 1,\ldots,T;\ m = 1,\ldots,M-1;\ i = 1,\ldots,I$ (11)
$M_{big} * \left(1 - B(A_{imt})\right) \geq It_{i,m+1,t} \quad t = 1,\ldots,T;\ m = 1,\ldots,M-1;\ i = 1,\ldots,I$ (12)
$M_{big} * B(X_{it}) \geq X_{it} \quad t = 1,\ldots,T;\ i = 1,\ldots,I$ (13)
$M_{big} * \left(1 - B(X_{it})\right) \geq It_{i1t} \quad t = 1,\ldots,T;\ i = 1,\ldots,I$ (14)

The same constraints (15)-(21) are used for the warehouse:

$It_{j,M-v-1,t-1} - \sum_{i=1}^{I} Q_{it} = It_{j,M-v,t} - A_{j,M-v-1,t} \quad t = 1,\ldots,T;\ j = 1,\ldots,J$ (15)
$It_{jm,t-1} - A_{j,m+1,t} = It_{j,m+1,t} - A_{jmt} \quad t = 1,\ldots,T;\ m = 1,\ldots,M-v-2;\ j = 1,\ldots,J$ (16)
$Q_{jt} - A_{j1t} = It_{j1t} - X_{jt} \quad t = 1,\ldots,T;\ j = 1,\ldots,J$ (17)
$M_{big} * B\left(A_{jmt}\right) \geq A_{jmt} \quad t = 1,\ldots,T;\ m = 1,\ldots,M-v-1;\ j = 1,\ldots,J$ (18)
$M_{big} * \left(1 - B\left(A_{jmt}\right)\right) \geq It_{j,m+1,t} \quad t = 1,\ldots,T;\ m = 1,\ldots,M-v-1;\ j = 1,\ldots,J$ (19)
$M_{big} * B\left(X_{jt}\right) \geq X_{jt} \quad t = 1,\ldots,T;\ j = 1,\ldots,J$ (20)
$M_{big} * \left(1 - B\left(X_{jt}\right)\right) \geq It_{j1t} \quad t = 1,\ldots,T;\ j = 1,\ldots,J$ (21)

Order:
Equations (22)-(27) keep track of the order timing. When the sum of the inventory level of ages $m = 1,..,M-1$ is lower than the re-order point S, an order Y is placed. Products with age M age left out of these equations because inventory is checked at the end of the period and thus are waste in the next period.

$\sum_{m=1}^{M-1} It_{imt} \leq S_i + M_{big} * (1 - Y_{it}) \quad i = 1,\ldots,I; t = 1,\ldots,T$ (22)
$\sum_{m=1}^{M-1} It_{imt} > S_i - M_{big} * Y_{it} \quad i = 1,\ldots,I; t = 1,..,T$ (23)
$\sum_{m=1}^{M-v-1} It_{jmt} \leq S_j + M_{big} * \left(1 - Y_{jt}\right) \quad j = 1,\ldots,J; t = 1,\ldots,T$ (24)
$\sum_{m=1}^{M-v-1} It_{jmt} > S_j - M_{big} * Y_{jt} \quad j = 1,\ldots,J; t = 1,\ldots,T$ (25)
$Q_{it} \leq M_{big} * Y_{it} \quad i = 1,\ldots,I;\ t = 1,\ldots,T$ (26)

$$Q_{jt} \leq M_{big} * Y_{jt} \quad j = 1, \ldots, J;\ t = 1, \ldots, T \tag{27}$$

Equations (28)-(38) are definition constraints.

$$Y_{it}, Y_{jt} \in \{0,1\} \quad i = 1, \ldots, I; j = 1, \ldots, J;\ t = 1, \ldots, T \tag{28}$$
$$B(A_{imt}) \in \{0,1\} \quad i = 1, \ldots, I; m = 1, \ldots, M-1; t = 1, \ldots, T \tag{29}$$
$$B(A_{jmt}) \in \{0,1\} \quad i = 1, \ldots, I; m = 1, \ldots, M-v-1; t = 1, \ldots, T \tag{30}$$
$$B(X_{it}), B(X_{jt}) \in \{0,1\} \quad t = 1, \ldots, T;\ i = 1, \ldots, I; j = 1, \ldots, J \tag{31}$$
$$A_{imt} \geq 0 \quad i = 1, \ldots, I;\ m = 1, \ldots, M-1, t = 1, \ldots, T \tag{32}$$
$$A_{jmt} \geq 0 \quad j = 1, \ldots, J; m = 1, \ldots, M-v-1, t = 1, \ldots, T \tag{33}$$
$$X_{it}, X_{jt} \geq 0 \quad i = 1, \ldots, I; j = 1, \ldots, J; t = 1, \ldots, T \tag{34}$$
$$TIout_{mt}, Iout_{imt} \geq 0, integers \quad i = 1, \ldots, I;\ m = 1, \ldots, M-v;\ t = 1, \ldots, T \tag{35}$$
$$It_{imt}, \geq 0, integers \quad i = 1, \ldots, I; m = 1, \ldots, M;\ t = 1, \ldots, T \tag{36}$$
$$It_{jmt} \geq 0, integers \quad j = 1, \ldots, J;\ m = 1, \ldots, M-v;\ t = 1, \ldots, T \tag{37}$$
$$Q_j, Q_i \geq 0, integers \quad i = 1, \ldots, I; j = 1, \ldots, J \tag{38}$$

4. Results with Brand & Bound algorithm

The Branch and Bound (B&B) algorithm, which is based on Linear Programming (LP), is the most commonly used method for solving Mixed Integer Linear Programming (MILP) problems. This approach involves exploring the solution space through LP-based relaxations of the MILP problem. The implementation of the B&B algorithm is structured as a tree search, where the LP relaxation serves as the root node, and new nodes are created by branching on an existing node when the optimal solution of the relaxation is fractional (Kostikas & Fragakis, 2004). In this research, Python 3.8 is used to solve the MILP model with optimization solver Gurobi v10.0.1 on a AMD Ryzen 7 3700U 8GB RAM with Radeon Vega Mobile Gfx 2.30 GHz. Section 4.1 presents time-restricted experiments that pertain to the optimality gaps and time limits. Section 4.2 describes the design of the sensitivity experiments that relate to demand, fixed order costs, lost sales costs, shelf life, and reorder point and presents the outcomes for the experiments.

4.1. Time-restricted experiments

This section presents the time-restricted experiments of the model, which incorporate time limits and optimality gaps due to the model's complexity and long computation times. Parameter T, the maximum numbers of period in the finite time horizon is increased during this analysis. The other parameters are kept constant during the increase of T in the model.

The chosen parameters that are kept constant are as follows: the number of retailers I is set to 5. Warehouse $j = 1$ is kept constant, because with multiple warehouses more constraints are necessary to divide the flow of products from different warehouses to the retailers. Maximum shelf life $M = 4$. A minimum leftover of $v = 2$ periods for the retailer. Purchase costs for the warehouse and retailers are $c_j = 2$ and $c_i = 3$ respectively. Holding costs are set to $h_j = 0.25$ and $h_i = 0.5$. Outdate costs are set to both the warehouse and retailers on $w_j, w_i = 10$. Fixed order costs are set to $k_j = 20$ and $k_i = 20$. In this model, a required service level is not specified. For maintaining sufficient inventory, the lost sales costs are used as a penalty for stockouts. The costs are therefore set for both the warehouse and retailers to $p_j, p_i = 20$. Since lead time is assumed to be 0, reorder points s_j and s_i only contain the safety stock. Demand is sampled over finite time horizon T with a Poisson distribution with a mean of 10. In the given scenario with Poisson distributed demand, the safety stock can be calculated using the Poisson distribution function in Excel. The formula used is $Poisson.dist(15, 10, 1) \geq 0.95$, where the parameters represent the inventory level (15), demand rate (10), and cumulative (1). In this scenario, with an initial inventory level of 15, a safety stock of 5 is determined. To augment the level of assurance even more, the safety stock is increased to 6 units (a 20% increase). Consequently, the reorder point, denoted as s_i, is set to 6. For the warehouse, $Poisson.dist(62, 50, 1) \geq 0.95$, In this scenario, with an initial inventory level of 62, a safety stock of 12 is determined. To augment the level of assurance even more in the same way as the retailers, the safety stock is increased to 14.4 units, rounding of to 15, also about 20% increase. Consequently, the reorder point, denoted as s_j , is set to 15. Values of the parameters for the base case are presented in table 1. In this research, different random seeds are used to evaluate the outcomes of the model. Different random seeds are used for several reasons. It allows for exploring the robustness of the model by considering multiple initial conditions, replicating the variability and uncertainty present in real-world scenarios. Additionally, exploring different random seeds enables the exploration of a broader solution space. Different random seeds can lead to different simulation outcomes, which allows for a broader range of possible outcomes. In section 4.1.1., the impact of different optimality gaps on the results of the model are shown. Table 2 is an overview of the experiments for the optimality gaps. In section 4.1.2, the impact of different time limits on the results of the model are presented. In table 3, an overview is displayed with the different experiments for time limits.

Table 1: Constant parameter values for time-restricted experiments

I	J	M	v	d_{it}	c_j	c_i	h_j	h_i	w_j	w_i	k_j	k_i	p_j	p_i	s_i	s_j
5	1	4	2	10	2	3	0.25	0.5	10	10	20	20	20	20	6	15

Table 2: Overview of experiments optimality gap

$\boldsymbol{T = 200}$	**Optimality gaps**
	20%
	10%
	5%
$\boldsymbol{T = 500}$	
	10%
	5%

Table 3: Overview of experiments time limits

$\boldsymbol{T = 200}$	**Time limit**
	5 minutes
	15 minutes
	1 hour
	2 hours
	6 hours

4.1.1. Optimality gaps

For the first experiment, T is set to 200. Due to the significant computational time required to obtain an optimal solution, the use of optimality gaps is often necessary to derive a solution in a timely manner. The optimality gap is the difference between the objective value of the current best solution found by the solver and the lower bound at the objective value of an optimal solution. It is shown as a percentage and indicates how close the solver is to finding an optimal solution. A small optimality gap indicates that the solver has made significant progress towards finding an optimal solution, while a large optimality gap indicates that there is still significant room for improvement (Gurobi Optimization, LLC, 2023). The following KPI's are used: fill-rate (FR), cycle service-level (CSL), waste (W), total costs (TC), and computational time (CT).

Looking at the results in table 4, it is visible that the fill rates for all the retailers are below 93%, meaning that the retailers were not able to meet at least 7% of the customer demand. Additionally, the cycle service levels for all the retailers are low, with some being as low as 72.5%. Interestingly, the waste for all the retailers is zero, indicating that the algorithm was able to avoid overstocking and waste. However, this may have come at the cost of lower fill rates and cycle service levels. In terms of total costs, the result obtained was 80,720 euros, which is a relatively high cost. Finally, the computational time was only 46 seconds, which shows that the algorithm was able to find a solution relatively quickly, but at the cost of sacrificing solution quality.

The retailers in table 5 with an optimality gap of 10% have a higher average fill rate and cycle service level compared to the retailers in table 4 with an optimality gap of 20%. The waste is higher compared to the results in table 4, with an average of 22.8 units across all retailers. This shows that the retailers are generating more waste than the retailers in table 4. The total cost of inventory for all retailers is €73,096.5, which is significantly lower than the one with a optimality gap of 20%. The computational time required to solve this problem was 287 seconds.

The outcomes reported in table 6 indicate that the retailers are achieving high fill rates and service level, with an average fill rate of 98.51% and cycle service level of 95.1%. This suggests that they are able to meet a high percentage of customer demand. The waste appears to be relatively low, with an average of 5.6 units across all retailers. The total cost of inventory for all retailers is €71,992 and the computational time was 341 seconds for an optimality gap of 5%.

In the case of the three conclusions drawn here, the differences in the results obtained comes from the difference in optimality gap. The optimality gap was first set to 20%, second to 10% and third to 5%. This means that the B&B algorithm used to solve this problem was set to terminate when the solution found was within 5% of the optimal solution, and so producing more accurate and high quality solutions, whereas the algorithm was set to terminate when the solution found was within 10% of the optimal solution.

According to the data in figure 1, there is an inverse relationship between optimality gap and computational time. When the optimality gap increases, the computational time decreases. In general, as the optimality gap increases, the fill rate and cycle service level decrease, while the total cost increases, as is displayed in figure 2. This is expected because a larger optimality gap allows for more suboptimal solutions that may have lower fill rates and cycle service levels and have higher costs.

Table 4: KPI's for 20% OG with T=200 with λ = 10

20%	**Retailer 1**	**Retailer 2**	**Retailer 3**	**Retailer 4**	**Retailer 5**	**Average**
FR (%)	90.38	92.03	90.71	88.80	92.15	90.81
CSL (%)	78	77.5	75.5	72.5	81.5	77.4
W (units)	0	0	0	0	0	0
TC (€)	80720					
CT (s)	46					

Table 5: KPI's for 10% OG with T=200 with λ = 10

10%	**Retailer 1**	**Retailer 2**	**Retailer 3**	**Retailer 4**	**Retailer 5**	**Average**
FR (%)	96.35	98.52	96.79	97.95	96.11	97.14
CSL (%)	91	95.5	92.5	94	90	92.4
W (units)	11	11	13	36	43	22.8
TC (€)	73096.5					
CT (s)	287					

Table 6: KPI's for 5% OG with T=200 with λ = 10

5%	**Retailer 1**	**Retailer 2**	**Retailer 3**	**Retailer 4**	**Retailer 5**	**Average**
FR (%)	96.80	98.50	98.84	99.25	98.16	98.51
CSL (%)	93.5	94.5	95.5	98	95	95.1
W (units)	1	2	7	10	8	5.6
TC (€)	71992					
CT (s)	341					

The results for the warehouse have been excluded from the tables in section 4.1. as they are identical in terms of demand and order quantity in every time period t. For example, with an optimality gap of 5% presented in table 7, it is visible that the demand at the warehouse is equal to 50, which is the to the sum of order quantities from the retailers ($\sum_{i=1}^{I} Q_i$). This sum of order quantities from the retailers remains consistent and equal every period, so every period the demand at the warehouse is equal to 50, ensuring predictability for the warehouse. As a result, the warehouse can place the same fixed order quantity for products in each period, as the demand pattern is stable and predictable, meaning that. Therefore, the warehouse orders the same amount of products ($\sum_{i=1}^{I} Q_i$) every period, resulting in a fill rate and cycle service level of 100% and zero waste every period. Table 7 displays the results for the warehouses from the $T = 200$ runs with varying optimality gaps of 5, 10 and 20%.

Table 7: Warehouse outcome for different OG's

	5%	**10%**	**20%**
$\sum_{i=1}^{I} Q_i$	50	48	45
Q_j	50	48	45
FR (%)	100	100	100
CSL (%)	100	100	100
W (units)	0	0	0

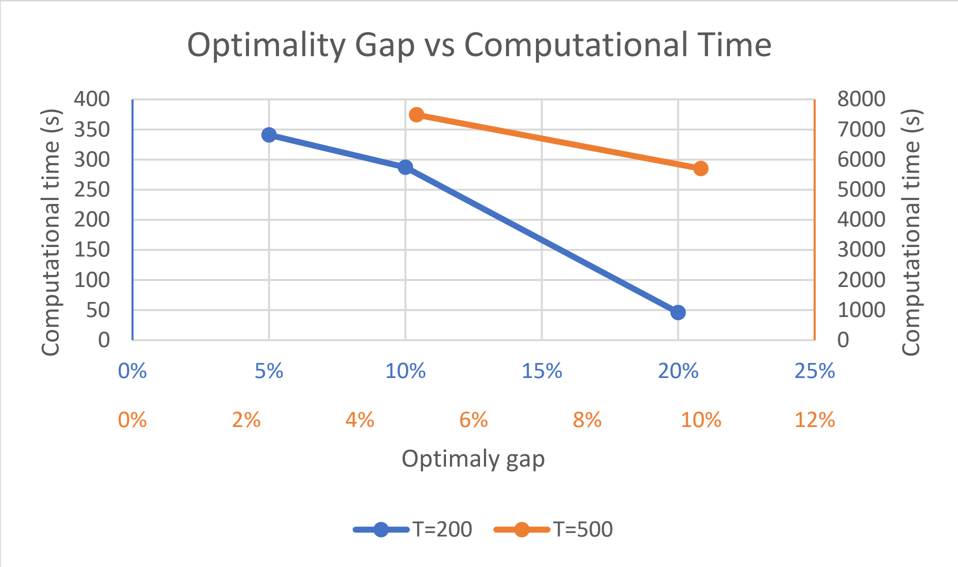

Figure 1: Optimality gap vs computational time for T=200 and T=500

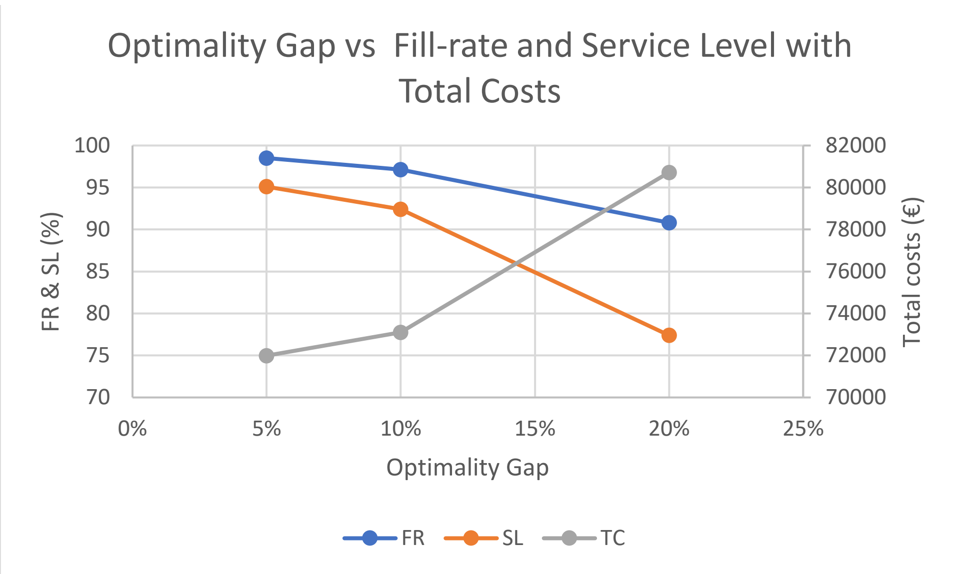

Figure 2: Optimality gap and performance metrics with total costs

In the second experiment, the finite time horizon T is extended to 500 periods, the rest of the parameters are kept constant. Notably, the outcomes from the first experiment with an optimality of 20% was found to be too low and will not be close enough to optimality for usage in real life situations, and therefore is excluded in the current analysis. Only optimization gaps of 10%, and 5% were obtained for the problem. The tables 8-9 below present the findings for experiment 2.

Table 8: KPI's for 10% OG with T=500

10%	**Retailer 1**	**Retailer 2**	**Retailer 3**	**Retailer 4**	**Retailer 5**	**Average**
FR (%)	93.42	94.21	93.83	93.08	93.94	93.70
CSL (%)	83.4	83	84.2	81.2	84	83.16
W (units)	0	0	0	0	0	0
TC (€)	189533.5					
CT (s)	5700					

Table 9: KPI's for 5% OG with T=500

5%	**Retailer 1**	**Retailer 2**	**Retailer 3**	**Retailer 4**	**Retailer 5**	**Average**
FR (%)	99.92	99.56	99.74	99.49	99.49	99.44
CSL (%)	99.6	98.2	99	98.6	98.8	98.44
W (units)	73	64	106	36	102	76.2
TC (€)	176946.5					
CT (s)	7485					

For the 5% optimality gap, the average fill rate is very high, close to 99.5%, and the cycle service level is also very high, close to 99%. The total cost is the lowest among the two optimality gaps. For the 10% optimality gap, the average fill rate is slightly lower, around 93.5%, and the cycle service level is also lower, around 83.5%. Looking at the results, it appears that in the case of 5%, there were some units of waste produced. There were 73 units of waste produced at retailer 1, 64 units of waste produced at retailer 2, 106 units of waste produced at retailer 3, 36 units of waste produced at retailer 4, and 102 units of waste produced at retailer 5.

Experiments with $T = 500$ are impractical due to the excessively long computational times required to obtain good results. Therefore, we will proceed with experiments using $T = 200$.

Table 10 provides the outcomes of five independent runs of the model to evaluate its precision under an optimality gap of 5%. Table 11 shows the performance of the same model under a 10% optimality gap. For the fill rates and cycle service levels, only the highest and lowest values are reported for making the tables more compact. Additionally, the average waste over the five retailers is used. Running the model multiple times with the exact same conditions, expect for generating new demand but still with the same average, can be beneficial for multiple reasons. Firstly, it helps assess the impact of randomness on the results by observing the variability in the solutions obtained from newly generated demand. Furthermore, multiple runs of the model can be used to assess how well the optimization algorithm or solver performs and how efficient it is. Moreover, it establishes benchmarks for future modifications or alternative models, providing a baseline for measuring improvements.

Table 10: Precision evaluation for 5% OG

OG 5%	**TC (€)**	**FR (%)**	**CSL (%)**	**W (units)**	**CT (s)**
Run 1	71992	96.8; 99.2	93.5; 98	5.6	341
Run 2	71477	99.3; 99.9	96; 99	27.6	513
Run 3	71689.5	96.0; 97.1	87; 91	4.4	490
Run 4	71479.5	96.4; 98.7	90; 94	0	570
Run 5	70805.5	98.1; 99.6	94; 99	16.2	825

Table 11: Precision evaluation for 10% OG

OG 10%	**TC (€)**	**FR (%)**	**CSL (%)**	**W (units)**	**CT (s)**
Run 1	73096.5	96.1; 98.5	90; 95.5	22.8	287
Run 2	73567.5	94.4; 96.9	85.5; 90.5	0.4	608
Run 3	73635	95.9; 97.9	88; 92.5	6.4	253
Run 4	74558	93.8; 96.5	87; 91	1	156
Run 5	74230.5	93.57; 95.33	85; 90	0.8	145

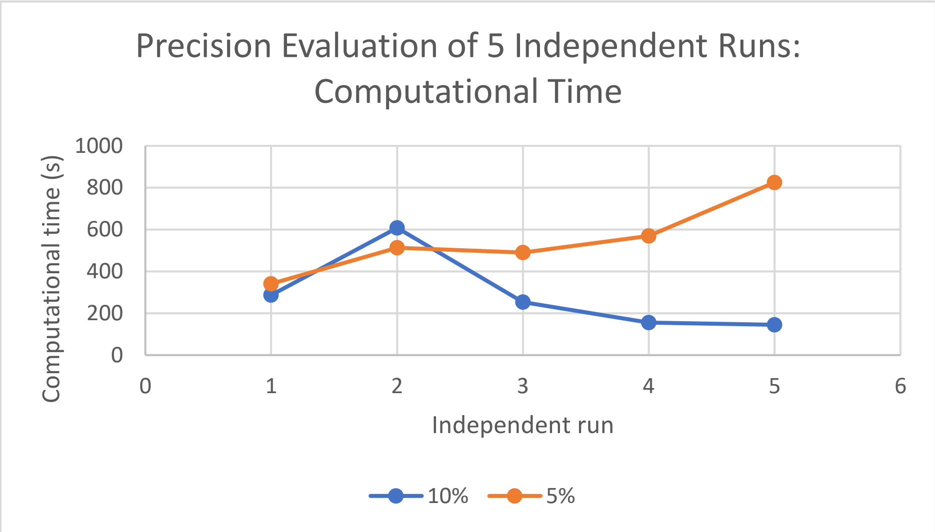

Figure 3: Precision evaluation for 10% and 5% OG

As is visible in tables 10-11 and figure 3, the differences between the five independent runs can be due to the various factors, such as the randomness of the generated demand. Every run of the model the random number generator of the demand produces different sequences of numbers and consequently leads to different solutions. Furthermore, the optimization algorithm may get stuck in local optima, which can lead to different results in different runs.

4.1.2. Time limits

Time limits are used to set a maximum time that the algorithm is allowed to run. Optimization problems can be complex and computationally intensive, and there can be instances where an algorithm may not find the optimal solution within a reasonable amount of time. Therefore, a time limit is set to ensure that the optimization process does not continue indefinitely. By setting a time limit, the algorithm can terminate the search and provide the best solution found within the given time. Time limits are set to 5 and 15 minutes and 1, 2 and 6 hours. The model is run with the same parameters described in section 4.1. with $T = 200$. The following KPI's are used: fill-rate (FR), cycle service-level (CSL), waste (W), total costs (TC), and optimality gap (OG).

Looking at tables 12-16 and figure 4, it appears that as the time limit increases, the results improve. This is reflected in the increasing fill rates and cycle service levels in most cases, and shows that more of the demand from the retailers is being met. The optimality gap also decreases as the time limit increases, showing that the solver is getting closer to the optimal solution. However, it is worth mentioning that even with a 6-hour time limit, there is still an optimality gap of 1.45%, indicating that there is still room for improvement. Overall, these results suggest that increasing the time limit can lead to improved results, but it is important to consider the trade-off between computational time and solution quality. Thereby, the results from the 15-min time limit are surprisingly good, even have a lower total costs than the 6-hour time limit. This can be possible because the optimization process is not linear and the usage of a random seed is in the model. Different starting points can affect the solution even within the same model. In some cases, the solver may be able to find a good solution quickly, and in others, it may require more time to explore other possibilities and find a better solution. The optimality gap represents the percentage difference between the best solution found and the optimal solution, so it is possible that the solver in the 15-min run found a good solution that was closer to the optimal solution than the one found in the 1-hour run. Similarly, the total costs can depend on the specific solution found, so it is possible that the solution found in the 2-hour run resulted in lower total costs than the solution found in the 6-hour run.

Table 12: KPI's for 5 min time limit

5 min	**Retailer 1**	**Retailer 2**	**Retailer 3**	**Retailer 4**	**Retailer 5**	**Average**
FR (%)	90.14	90.62	89.93	91.82	89.36	90.37
CSL (%)	76.5	79	76.5	81	74	77.4
W (units)	0	0	0	0	0	0
TC (€)	79713					
OG (%)	15.23					

Table 13: KPI's for 15 min time limit

15 min	**Retailer 1**	**Retailer 2**	**Retailer 3**	**Retailer 4**	**Retailer 5**	**Average**
FR (%)	99.79	99.62	99.44	99.24	99.32	99.48
CSL (%)	98.5	98.5	97.5	97	98	97.9
W (units)	8	4	10	1	12	7
TC (€)	69480					
OG (%)	2.78					

Table 14: KPI's for 1 hour time limit

1 hour	**Retailer 1**	**Retailer 2**	**Retailer 3**	**Retailer 4**	**Retailer 5**	**Average**
FR (%)	97.31	98.60	97.62	97.67	97.93	97.83
CSL (%)	91	94.5	92.5	93	92	92.8
W (units)	0	0	0	0	0	0
TC (€)	70239.5					
OG (%)	2,80					

Table 15: KPI's for 2 hours' time limit

2 hours	**Retailer 1**	**Retailer 2**	**Retailer 3**	**Retailer 4**	**Retailer 5**	**Average**
FR (%)	99.65	99.65	99.44	99.34	99.21	99.46
CSL (%)	97.5	98.5	97	96.5	97	97.3
W (units)	25	8	13	18	9	14.6
TC (€)	69214.5					
OG (%)	1.93					

Table 16: KPI's for 6 hours' time limit

6 hours	**Retailer 1**	**Retailer 2**	**Retailer 3**	**Retailer 4**	**Retailer 5**	**Average**
FR (%)	99.71	100	100	99.77	99.83	99.86
CSL (%)	98.5	100	100	98.5	98	99
W (units)	46	31	31	42	5	31
TC (€)	70225.5					
OG (%)	1.45					

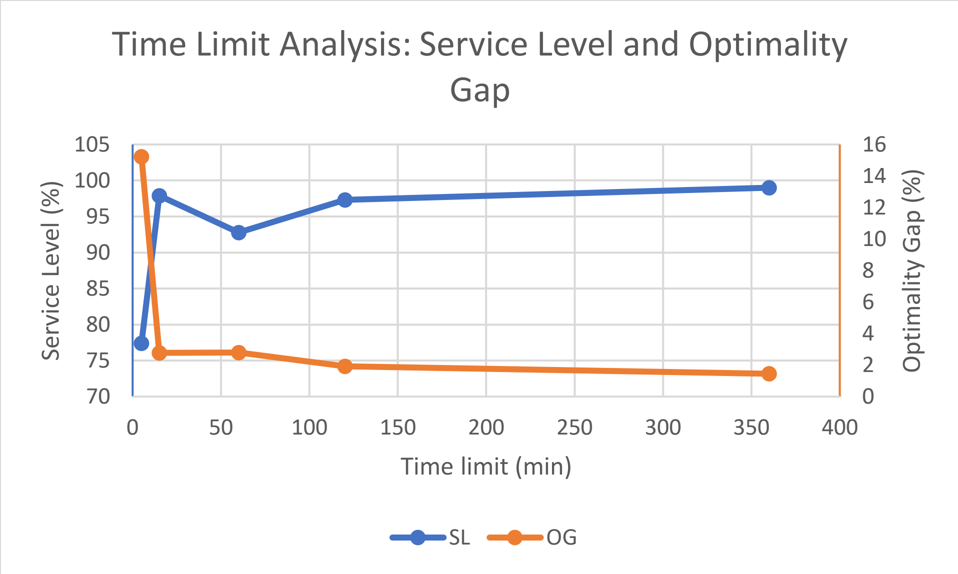

Figure 4: Time limit analysis for service level and optimality gap

4.2. *Model sensitivity experiments*

In this section, the behaviour of the model for different parameters and different demand patterns will be investigated. All experiments are executed with $I = 5$ and $T = 200$ and with a time limit of 15 minutes to avoid excessive computational time per experiment, which could hinder the advancement of this research. The Poisson distributed demand function will vary between two values ($\lambda = 5, \lambda = 50$), but the demand average is identical for every retailer. Also, different demand averages will be implemented for the different retailers within the same model: retailers 1 and 2 have an average demand of 5 ($\lambda = 5$), retailer 3 an average demand of 10 ($\lambda = 10$), retailers 4 and 5 an average demand of 15 ($\lambda = 15$). Different values for fixed order costs will be investigated ($k_i, k_j = 0$, $k_i, k_j = 50; 50$). In addition, one scenario for fixed order costs in relation to demand, low demand ($\lambda = 3$) with high fixed order costs , ($k_i, k_j = 50; 50$). Thereby, multiple experiments will be investigated with different penalties for lost sales costs in relation to the outdate costs. Three scenarios are described:

$w_i, w_j \geq p_i, p_j$	$w_i, w_j = 10, p_i, p_j = 0$	$w_i, w_j = 10, p_i, p_j = 5$	$w_i, w_j = 10, p_i, p_j = 9$
$w_i, w_j = p_i, p_j$	$w_i, w_j = 10, p_i, p_j = 10$		
$w_i, w_j \leq p_i, p_j$	$w_i, w_j = 10, p_i, p_j = 11$	$w_i, w_j = 10, p_i, p_j = 50$	$w_i, w_j = 10, p_i, p_j = 200$

Furthermore, two scenarios will be investigated with different shelf life's $M = 3, v = 1$; $M = 3, v = 2$. Finally, the model is free in calculating the reorder points itself.

Experiments are compared to a base case. This base case is the model with parameters presented in table 17. For comparing accurately with the base case, the model is run five times independently to check for the precision of the model under a time limit of 15 minutes and to compare the results of the experiments more accurately to the base case. The averages over the five runs are presented in table 18 below. The five full tables are in Appendix B.

Table 17: Parameters for base case

I	J	T	M	v	d_{it}	c_i	c_j	w_i	w_j	p_i	p_j	h_i	h_j	k_i	k_j	s_i	s_j
5	1	200	4	2	10	3	2	10	10	20	20	0.5	0.25	20	20	6	15

Table 18: Average over five runs for the base case

	Retailer 1	Retailer 2	Retailer 3	Retailer 4	Retailer 5	Average
FR (%)	99.48	96.95	99.75	98.39	99.74	98.86
CSL (%)	97.9	89.9	99.2	94.6	98.7	96.06
W (units)	7	0	4.4	0.8	20.2	6.48
Q (units)	50	48	50	50	51	49.8
TC (€)	69480	70838	68432	70534.5	70256.5	69908.2
OG (%)	2.78	5.79	2.79	3.32	2.56	3.45

Demand

The experimental results indicate that the model's ability to identify an optimal solution may be influenced by the demand level. This is visible by looking at the optimality gap. In table 19, the optimality gap is 4.47 with a demand of five, while with a demand of 50, the optimality gap is 0.72. A reason could be that larger demand values may lead to more predictable patterns in demand. This means that if there is more demand, it becomes easier to find the best way to set up the supply chain to meet that demand. Thus, the results of the experiments highlight the potential benefits of considering larger demand values when optimizing supply chain configurations in this model. The tables (19-21) below present the results.

Reorder levels are adjusted. For the retailer, $s_i = 4$ and for the warehouse $s_j = 8$.

Table 19: KPI's for a demand of five

$\lambda = 5$	W	Retailer 1	Retailer 2	Retailer 3	Retailer 4	Retailer 5	Average
FR (%)	100	95.87	96.43	94.72	94.19	93.60	94.96
CSL (%)	100	90.5	90.5	90	87	88	89.2
W (units)	0	1	0	1	0	0	0.4
Q (units)	24						
TC (€)	44378						
OG (%)	4.47						

Reorder levels are adjusted for the retailer to $s_i = 12$ and for the warehouse to $s_j = 26$.

Table 20: KPI's for a demand of 50

$\lambda = 50$	W	Retailer 1	Retailer 2	Retailer 3	Retailer 4	Retailer 5	Average
FR (%)	100	99.87	99.97	99.60	99.82	99.90	99.83
CSL (%)	100	98.5	99.5	98	99	99.5	98.9
W (units)	0	0	0	0	0	25	5
Q (units)	250						
TC (€)	278558						
OG (%)	0.72						

Reorder levels are adjusted for retailer 1 and 2 to $s_i = 4$, retailer 3 to $s_i = 6$, retailer 4 and 5 to $s_i = 7$, and for the warehouse to $s_j = 12$.

Table 21: KPI's for various demands

	W	**1** ($\lambda = 5$)	**2** ($\lambda = 5$)	**3** ($\lambda = 10$)	**4** ($\lambda = 15$)	**5** ($\lambda = 15$)	Average
FR (%)	100	99.61	99.80	100	100	100	99.88
CSL (%)	100	99	99	100	100	100	99.6
W (units)	0	52	53	6	0	1	22.4
Q (units)	50						
TC (€)	69021.5						
OG (%)	3.73						

To further examine this phenomenon, an additional experiment is done using ten retailers. The purpose of conducting this additional experiment is to determine whether the model shows improved optimization capabilities when dealing with larger demand values as opposed to smaller ones within the same supply chain. Because I is increased to ten and T is held constant at 200, the complexity of this model has increased. Results from this experiment are presented in table 22.

Table 22: KPI's for 10 retailers with various demands

	W	**1 λ = 3**	**2 λ = 3**	**3 λ = 5**	**4 λ = 5**	**5 λ = 10**	**6 λ = 10**	**7 λ = 20**	**8 λ= 20**	**9 λ = 50**	**10 λ = 50**	**Average**
FR (%)	100	66.76	74.03	87.49	82.21	99.26	97.01	99.83	100	100	99.96	87.66
CSL (%)	100	63	71	84	82	97	93	99	100	100	99.5	86.85
W (units)	0	1	1	0	0	1	0	0	0	0	0	0.3
Q (units)	171											
TC (€)	216095											
OG (%)	4.73											

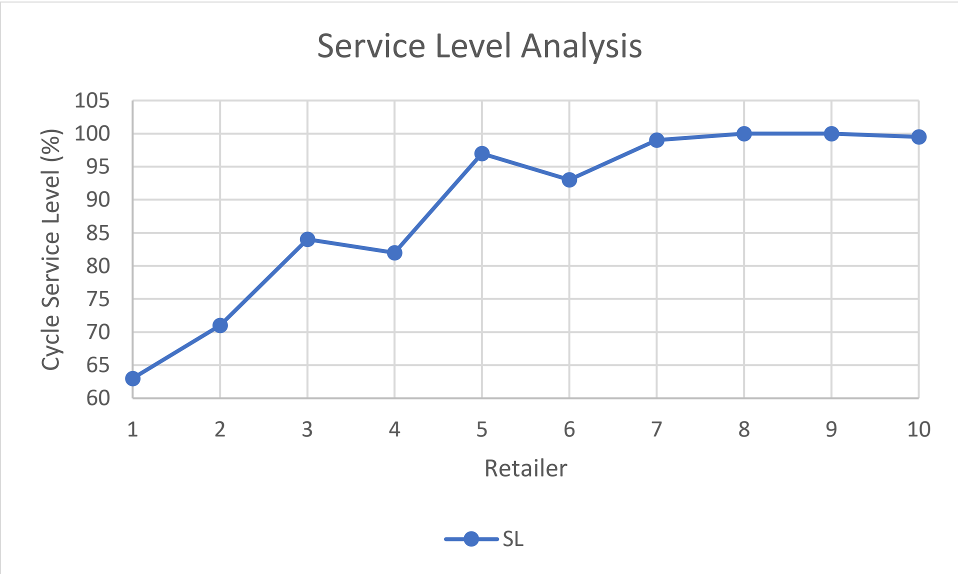

Figure 5: Service level analysis

Table 22 clearly demonstrates that as the demand of the retailer increases, both the fill rate and cycle service level also increase. In figure 5, a visual representation of the cycle service level is presented in relation to the different retailers with various demands. This observation supports the conclusion that the model exhibits improved optimization capabilities when faced with higher demand values. Notably, the model's results for retailers with a demand of ten or higher consistently exceed a 95% service level, with the exception of retailer 6. These outcomes indicate that the model's performance is satisfactory and appropriate for the given demand levels. Additionally, it is observed that the results show a slightly higher optimality gap, which can be attributed to the increased number of retailers from 5 to 10, while maintaining a constant time limit of 15 minutes. It is logical to expect this outcome as the optimization complexity grows with a larger number of retailers within the given time limit. Moreover, there is no significant difference between the demand values of 20 and 50. However, to draw more definitive conclusions regarding this matter, further experiments would be necessary.

Fixed order costs

Setting the fixed order costs to zero results in a lower optimality gap for both the warehouse and retailers. This is because having no costs related to ordering, cancels out the variables Y_{jt} and Y_{it}. Meaning that the model has less variables and so can more easily optimize the supply chain configuration to meet the demand while minimizing the total cost, resulting in lower optimality gap values. In table 23 the results for fixed order costs of zero are presented.

Table 23: KPI's for fixed order costs of zero

$k_i, k_j = 0$	**W**	**Retailer 1**	**Retailer 2**	**Retailer 3**	**Retailer 4**	**Retailer 5**	**Average**
FR (%)	100	99.78	99.69	98.43	99.50	99.44	99.37
CSL (%)	100	98	99.5	96.5	99.5	98.5	98.4
W (units)	0	19	28	17	23	0	17.4
Q (units)	50						
TC (€)	54283						
OG (%)	0.02						

In contrast to when the fixed order costs are non-zero, the model has to balance the trade-off between ordering too frequently, incurring high ordering costs, and ordering too infrequently, risking stockouts and incurring high holding costs. This added complexity can make it harder for the model to identify the optimal supply chain configuration, resulting in higher optimality gap values. Table 24 presents the results for a fixed order costs of 50.

Table 24: KPI's for fixed order costs of 50

$k_i, k_j = 50$	**W**	**Retailer 1**	**Retailer 2**	**Retailer 3**	**Retailer 4**	**Retailer 5**	**Average**
FR (%)	100	95.62	97.24	96.73	96.60	96.97	96.63
CSL (%)	100	87	90	89.5	90.5	90.5	89.5
W (units)	0	0	2	0	2	0	0.8
Q (units)	48						
TC (€)	89620						
OG (%)	6.56						

Combining a low demand ($\lambda = 3$) with high fixed order costs ($k_i, k_j = 50$) as is presented in table 25, causes that the model struggles to identify an optimal solution that minimizes costs. This can be seen in the relatively high optimality gap of 7.58%, indicating that the solution found by the model is not close to an optimal solution. In figure 6, a graph is presented where the optimality gap is shown in relation to the fixed order costs.

Table 25: KPI's for a demand of three and fixed order costs of 50

$\lambda = 3$, $k_i, k_j = 50$	**W**	**Retailer 1** $\lambda = 3$	**Retailer 2** $\lambda = 3$	**Retailer 3** $\lambda = 3$	**Retailer 4** $\lambda = 3$	**Retailer 5** $\lambda = 3$	**Average**
FR (%)	100	94.96	94.39	96.60	95.06	94.76	95.15
CSL (%)	100	90.5	92	92.5	92.5	91	91.7
W (units)	0	26	28	14	23	26	23.4
Q (units)	15						
TC (€)	51091.5						
OG (%)	7.58						

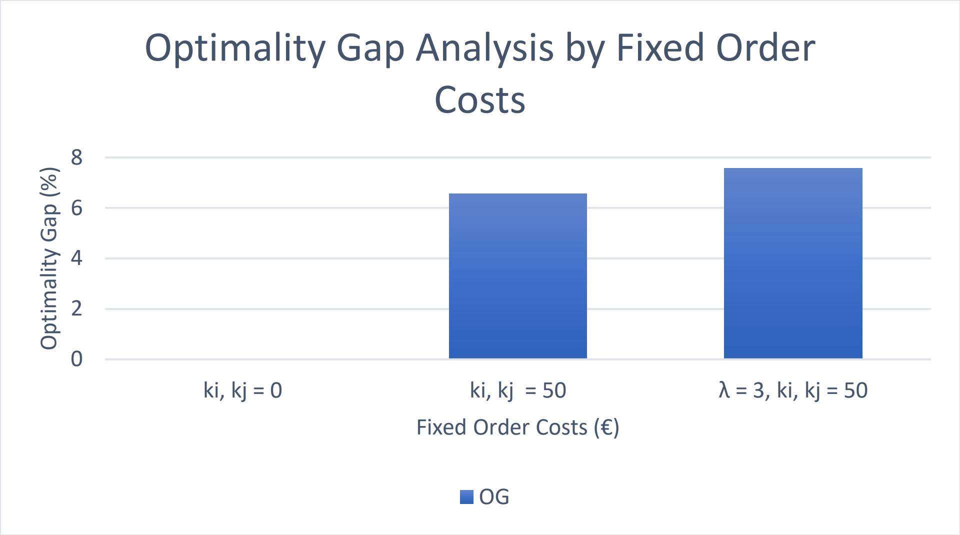

Figure 6: Optimality gap analysis by fixed order costs

Lost sales costs in relation to outdate costs

$w_i, w_j \geq p_i, p_j$

$w_i, w_j = 10, p_i, p_j = 0$:

Setting the lost sales costs to zero, results in not ordering any products throughout the whole optimization process. This is due to the fact that the costs for ordering and holding products is higher than not ordering anything at all, as there is no associated penalty for not meeting customer demand. The full table is presented in Appendix C.

$w_i, w_j = 10, p_i, p_j = 5$:

The outcomes remain unchanged when the lost sales costs are set to five, as they were previously. Ordering no products at all still results in lower total costs than ordering products to satisfy customer demand. Full table is presented in Appendix D.

$w_i, w_j = 10, p_i, p_j = 9$:

Now, setting the lost sales costs to nine, the model is ordering products to satisfy customer demand. As is presented in table 26. Satisfying customer demand results now in lower total costs, than not ordering anything. As the lost sales costs are still lower than the outdate costs, the model prioritizes minimizing the waste by not ordering excessive products instead of making significant efforts to fulfil all the customer demand. Specifically, the warehouse orders 46 products, which represents a decrease from the 50 products ordered in the base case.

Table 26: KPI's for lost sales costs of nine

	W	Retailer 1	Retailer 2	Retailer 3	Retailer 4	Retailer 5	Average
FR (%)	100	93.96	92.34	94.33	94.30	92.55	93.50
CSL (%)	100	84	82	85	82.5	83	83.3
W (units)	0	0	0	0	0	0	0
Q (units)	46						
TC (€)	67449						
OG (%)	2.81						

$w_i, w_j = p_i, p_j$

$w_i, w_j = 10, p_i, p_j = 10$:

When the lost sales costs are set to ten, and the outdate costs and lost sales costs are equalized, the resulting outcomes are somewhat similar to those observed in the table 26. The model orders products to satisfy customer demand while also considering the trade-off between meeting demand and minimizing waste. A notable difference in this scenario is that the total number of products ordered by the warehouse has increased by one, in comparison to the previous case. Results are presented in table 27.

Table 27: KPI's for lost sales costs of ten

	W	**Retailer 1**	**Retailer 2**	**Retailer 3**	**Retailer 4**	**Retailer 5**	**Average**
FR (%)	100	93.18	93.96	94.13	93.80	94.22	93.86
CSL (%)	100	81.5	82	84	82	84.5	82.8
W (units)	0	0	0	0	0	0	0
Q (units)	47						
TC (€)	69319						
OG (%)	2.85						

$w_i, w_j \leq p_i, p_j$

$w_i, w_j = 10, p_i, p_j = 11$:

Raising the lost sales costs to 11 yields again results somewhat similar to those in the previous scenario. The difference being that the total number of products ordered by the warehouse again increased by one unit, visible in table 28. Moreover, the fill-rate and cycle service level improved slightly, while the waste remained zero across the supply chain. The higher optimality gap indicates that the optimization process faced greater challenges in achieving an optimal solution.

Table 28: KPI's for lost sales costs of eleven

	W	**Retailer 1**	**Retailer 2**	**Retailer 3**	**Retailer 4**	**Retailer 5**	**Average**
FR (%)	100	92.70	95.70	94.73	93.68	95.34	94.43
CSL (%)	100	82	87	84	81	88	84.4
W (units)	0	0	0	0	0	0	0
Q (units)	48						
TC (€)	70598.5						
OG (%)	3.37						

$w_i, w_j = 10, p_i, p_j = 50$:

Increasing the lost sales costs to 50 results in the outcomes presented in table 29 . The cycle service level and fill rate being close to 100% suggests that the model is able to satisfy almost all customer demand during the optimization process. In this case, the lost sales costs are much higher than the outdate costs, which means that the model is placing a high priority on meeting customer demand to avoid incurring the higher penalty for lost sales. But with avoiding this high penalty, the model orders more products and thus risking waste. The average waste is 35.2 units, indicating that the model is generating significantly more waste compared to the previous cases and the base case. In addition, the total number of products ordered by the warehouse increased to 51, indicating that the retailers are ordering more products and this is reflected in the high fill rate and service level.

Table 29: KPI's for lost sales costs of 50

	W	Retailer 1	Retailer 2	Retailer 3	Retailer 4	Retailer 5	Average
FR (%)	100	100	100	100	99.95	100	99.99
CSL (%)	100	100	100	100	99.5	100	99.99
W (units)	0	42	5	34	47	48	35.2
Q (units)	51						
TC (€)	70813						
OG (%)	3.98						

As the lost sales costs increase, the optimality gap also increases, indicating that the model encounters greater difficulty in achieving an optimal solution with higher lost sales costs. To ensure that this statement is valid, the lost sales costs are increased to 200. Results are presented in the table below.

Table 30: KPI's for lost sales costs of 200

	W	Retailer 1	Retailer 2	Retailer 3	Retailer 4	Retailer 5	Average
FR (%)	100	99.89	99.40	99.72	99.64	99.96	99.72
CSL (%)	100	99	98	99	98	99.5	98.7
W (units)	0	41	57	11	35	59	40.6
Q (units)	51						
TC (€)	77825						
OG (%)	12.43						

As is visible in table 30, the optimality gap is 12.43%. Confirming that the model is encountering greater difficulty in achieving an optimal solution when the lost sales costs are high. In figure 7, a representation of the outdate and lost sales costs are presented in relation to the average amount of waste that is produced in the supply chain. Figure 8 illustrates that as the costs associated with lost sales rise, there is a simultaneous increase observed in both the cycle service level and the optimality gap.

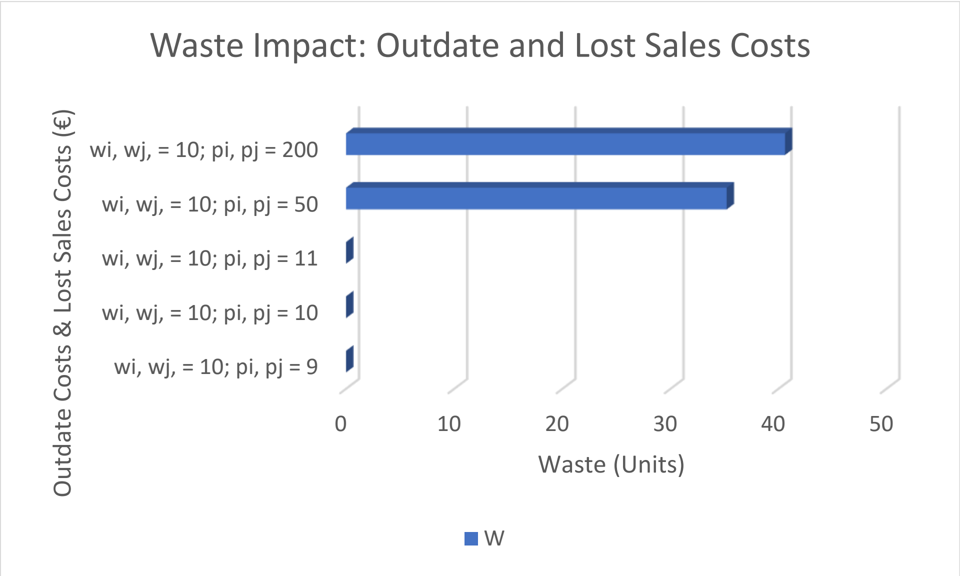

Figure 7: Waste Impact for outdate and lost sales costs

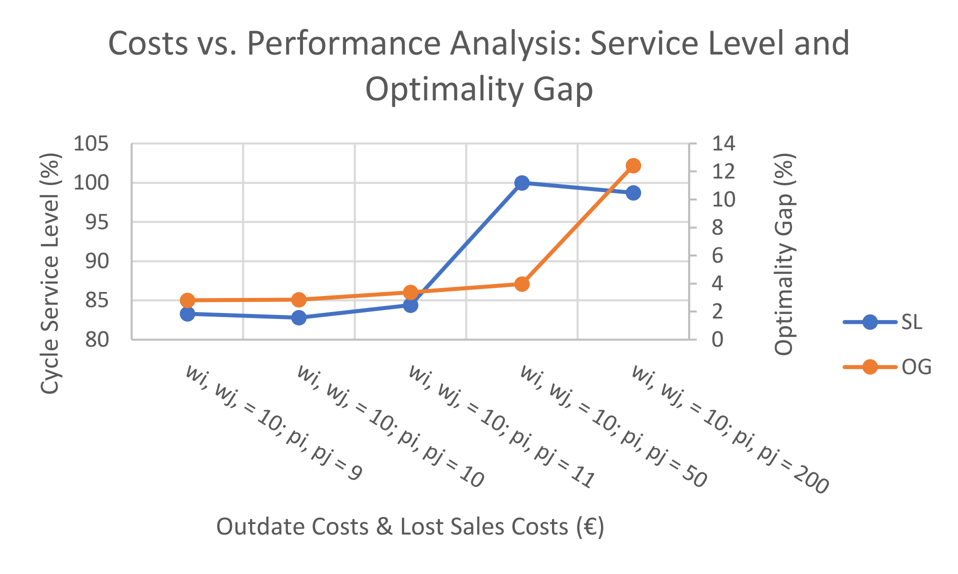

Figure 8: Cost and performance analysis

Shelf life

$M = 3, v = 1$

Decreasing the maximum age of the product by one to $M = 3$ and at the same time decreasing the minimum amount of periods leftover when the items are delivered at the retailer to $v = 1$, gives the outcomes presented in table 31 . Comparing table 31 to the base case, it is shown that optimality gap is significantly lower in this case, an explanation for that is that the model can more easily reach an optimal solution because of less variables within the model. Because index m now has the value three instead of four and v is one instead of two, the variables $It_{jmt}, It_{imt}, TIout_{mt}$, and $Iout_{imt}$ all have less variables and therefore, the complexity of the model is reduced.

Table 31: KPI's for shelf life of 3 and periods leftover of 1

	W	Retailer 1	Retailer 2	Retailer 3	Retailer 4	Retailer 5	Average
FR (%)	100	99.17	97.16	99.07	99.33	98.34	98.61
CSL (%)	100	97	91.5	96.5	95.5	95	95.1
W (units)	0	31	27	21	8	20	21.4
Q (units)	50						
TC (€)	72697.5						
OG (%)	1.75						

$M = 3, v = 2$

The difference between table 31 and 32, is the value of the parameter v, which is the maximum allowable age of a product when it arrives at the retailer's warehouse. In the first table, $v = 1$, which means that the warehouse can only dispatch products that are at most two periods old ($M - 1$). In the second table, $v = 2$, which means that the warehouse can dispatch products that are at most one period old ($M - 2$). This implies that the products in the second table have a longer usable lifespan when they arrive at the retailer's warehouse. Because $v = 2$, the variables $It_{jmt}, TIout_{mt}$, and $Iout_{imt}$ have less variables since index m ranges from one to $M - v$. This leads to a reduction in complexity and so improved optimization capabilities within the time limit.

The impact of this difference is reflected in the model's performance. In table 32 , the fill rate and cycle service level metrics are generally higher. The waste metric is also lower, indicating that the model is better able to manage inventory and reduce excess inventory that may result in waste.

Table 32: KPI's for shelf life of 3 and periods leftover of 2

	W	Retailer 1	Retailer 2	Retailer 3	Retailer 4	Retailer 5	Average
FR (%)	100	99.71	98.99	99.35	97.88	99.60	99.11
CSL (%)	100	98	96.5	96.5	94	98.5	96.7
W (units)	0	3	0	2	6	2	2.6
Q (units)	51						
TC (€)	72406.25						
OG (%)	1.51						

Reorder point

Providing the model with the ability to calculate the reorder levels is reflected in the resulting optimality gap. As the number of decision variables increases, the mathematical model that represents the problem becomes more complex, and solving it requires more computational time. The experiments consistently used an average reorder level of six for retailers, indicating that the reorder formula is effective. Similarly, the warehouse's ordering of the same amount of products each period and having the same demand results in a reorder level of zero, which is also deemed appropriate in this research.

Table 33: KPI's for free reorder point

	W	Retailer 1	Retailer 2	Retailer 3	Retailer 4	Retailer 5	Average
FR (%)	100	98.27	98.68	99.13	98.97	97.10	98.43
CSL (%)	100	95.5	95.5	94.5	95.5	91.5	94.5
W (units)	0	0	19	1	7	0	5.4
$\boldsymbol{s_j}, \boldsymbol{s_i}$	0	8	7	5	6	4	6
Q (units)	49						
TC (€)	69815,5						
OG (%)	3.37						

5. Discussion

In this study, a branch and bound (B&B) algorithm is used to optimize a multi-echelon inventory supply chain with perishable products in order to reduce the bullwhip effect. The research focused on evaluating the algorithm's performance through time-restricted experiments and model sensitivity experiments. Lower optimality gaps (5%) led to accurate, high-quality solutions with high fill rates and cycle service levels, while higher gaps (10% and 20%) decreased fill rates and cycle service levels and increased total costs, but computational times were lower. Indicating that lowering the optimality gap leads to better overall performance of the inventory model however, the trade-off for this increase in performance comes with the cost of longer computational times. Time limits played a major role, with longer limits improving solver performance and reducing optimality gaps. Using higher demand values when optimizing the supply chain, resulted in lower optimality gaps and improved optimization capabilities. Setting fixed order costs to zero and decreasing shelf life, causes to model to have improved solver performance within the given time limit of 15 minutes, due to less variables within the model. This indicates that reducing complexity by simplifying cost considerations and shelf life constraints can lead to better results within limited computational time. Increasing lost sales costs, causes the optimality gap to increase as well, indicating that the model encounters greater difficulty in achieving an optimal solution with higher lost sales costs. The findings of the model sensitivity experiments highlight the trade-offs and considerations involved in multi-echelon inventory optimization.

Using this model with deterministic demand causes it to be computationally less intensive than stochastic models. This is advantageous in this case because of time restrictions in this research. The findings conclude that using a time limit of 15 minutes with the given parameters and not reducing the complexity of the model, outcomes are not within 2% to an optimal one even with deterministic demand, thereby the computational time required for finding an optimal solution was prohibitively long, making it impossible to calculate an optimal solution for the problem in this research. On the other hand, the deterministic model assumes known demand, neglecting the uncertainty in real-world situations.

Reflecting on the research of Cohen & Moon (1991) and this research, in both studies a MILP model with deterministic demand is formulated and solved using an algorithm. Both studies illustrate the performance of the solution algorithm and demonstrate the effectiveness of the model by changing supply chain costs. While in this research a B&B algorithm is used within the solver Gurobi to solve the problem, Cohen & Moon (1991) use the Benders decomposition algorithm and for comparison purposes uses a B&B to solve the problem. From the comparison, it is visible that the B&B algorithm had lower computational time (7.25 seconds) when the model had less variables (30 binary and 81 continuous) in comparison to the Benders decomposition algorithm (25.53 seconds). But the B&B algorithm had higher computational time (76.81 seconds) with more variables (60 binary and 204 continuous) in comparison with the Benders decomposition algorithm (48.60 seconds). Concluding from the article of Cohen & Moon (1991) the B&B execution time grew rapidly with the problem size. For further research, this problem could be solved using the Benders algorithm in order to check if computational times are lower.

It is important to note that we did not use a fixed seed for the simulation experiments, which has implications for the comparability of the results. In this case, the decision not to use a fixed seed was made because of the potential impact of different random demand sequences on the outcomes of the simulation. Without a fixed seed, the random demand sequences used in different simulation runs leads to differences in the generated demand patterns and so affecting the key performance indicators. To address this issue and enhance the methodology of future research in this area, it is recommend using a fixed seed in experiments. This will allow for a more standardized comparison of results.

Thereby, T, the maximum number of periods for the finite time horizon, was set to 200. However, it is important to take into account that this value might be too low to obtain optimal / near-optimal or reliable results. The value of T should be sufficiently large to capture the dynamics and behaviour of the model. In this case, the decision to set it at 200 periods was based on the limitation of the computational time. But this choice could have limitations in terms of achieving accurate and robust results. With this, the results can lack statistical significance, meaning that the observed differences could have occurred by chance. Because of that, the conclusions drawn from the simulation are weakened and makes it challenging to make reliable comparisons or generalize the findings to real-world scenarios. To address this concern and improve the reliability of future research, it is recommended to assess the appropriate length of the time horizon and increase T accordingly.

The decision to set a zero lead time assumes that there is no delay between placing an order and receiving the inventory or in this case, ordering at the end of period t and receiving it at the beginning of period $t + 1$. While this simplification can be appropriate in this case, it disregards the uncertainties associated with real-world supply chains. In practice, lead times can vary due to factors such as supplier variability, transportation and logistics problems, and seasonality and demand fluctuations.

Additionally, the use of a full FIFO policy implies that inventory is allocated and sold strictly based on the order in which it was received. While this policy may be suitable in certain perishable inventory settings for example in warehouses or wholesalers, it may not reflect the actual allocation by how it goes in real-life situations at retailers. In real-life situations, customers at retailers have the freedom to select the products they purchase, and therefore, a full FIFO policy is not followed. Which could result in more waste.

Considering all unmet demand as lost sales assumes that no backorders or alternative fulfilment methods are available. This assumption simplifies the modelling process by treating any unmet demand as permanently lost, but it may not accurately represent the situations in real-world retailers. In reality, retailers often employ substitutions to fulfil unmet demand and maintain customer satisfaction for perishable products. Neglecting these aspects may lead to an overestimation of total costs and an underestimation of cycle service level.

6. Conclusion

In conclusion, this study focussed on a multi-echelon inventory optimization problem for a perishable product with a fixed lifetime. Several assumptions were made: zero lead time, implementation of First-In-First-Out issuing policy for all products, and considering all unmet demand as lost sales. In order to address this challenge, a mixed-integer linear programming model was created. To solve the MILP model, we used the MILP solver provided by Gurobi to handle the optimization process. And used a branch and bound algorithm using Python as the programming language. This article is contributing to the literature by providing an efficient and practical solving method within a MILP model tailored to address the multi-echelon inventory optimization problem with perishable products using a FIFO policy and lost sales.

The time-restricted experiments presented in this research focussed on evaluating the performance of the algorithm by considering optimality gaps and time limits. The results showed the effectiveness of the algorithm in generating near-optimal solutions for the MILP problem. The optimality gap played a big role in determining the solution quality in terms of fill rate, cycle service level and total costs. For the case $T = 200$, the algorithm produced solutions with optimality gaps of 5%, 10%, and 20%. Tables were provided to show the outcomes for different optimality gaps, including key performance indicators such as total costs, fill-rate, cycle service-level, waste, and computational time. The analysis of the results revealed that lower optimality gaps (5%) led to more accurate and high-quality solutions. Retailers achieved high fill rates and cycle service levels. However, as the optimality gap increased (10% and 20%), the fill rates and cycle service levels decreased, resulting in higher total costs. The results indicated a trade-off between solution quality and computational time. Higher optimality gaps resulted in lower fill rates and cycle service levels, but shorter computational times.

Time limits also play a big role in the optimization process for complex and computationally intensive problems. By setting a maximum time for the algorithm to run, we ensure that the optimization process does not continue indefinitely. In this study, time limits were set on 5 and 15 minutes, 1, 2, and 6 hours, and the results were analysed. The results indicate that as the time limit increases, the results improves. This improvement is shown in the increasing fill rates and cycle service levels and the decrease in total costs.

Overall, the findings highlighted the importance of setting an appropriate optimality gap / time limit to balance the solution quality and computational efficiency.

The model sensitivity experiments conducted in this study aimed to investigate the behaviour of the model under different parameters and demand patterns. The experiments were performed with a fixed time limit of 15 minutes to ensure a reasonably short computational time. Multiple findings were obtained from the experiments conducted in this study. Firstly, it was observed that the level of demand has a significant impact on the model's ability to identify an optimal solution. Additionally, the analysis of fixed order costs revealed that setting both warehouse and retailer fixed order costs to zero resulted in lower optimality gaps. Suggesting that the model’s ability to optimize the decision variables becomes easier when the model is provided with less variables. The experiments also investigated the relationship between lost sales costs and outdate costs. Different scenarios were examined, wherein the lost sales costs were varied relative to the outdate costs. When the lost sales costs were set to zero or kept low, the model choose to not order any products, as the costs associated with ordering and holding inventory were relatively higher. However, when the lost sales costs were higher, yet still lower than the outdate costs, the model prioritized waste reduction and placed orders to meet some customer demand. Increasing the lost sale costs, and coming to the point where lost sales costs are higher than outdate costs, the model struggles more to find optimal solutions, which is visible in the optimality gaps that is increasing accordingly. Thereby, the model prioritizes that all customer demand is met instead of reducing waste.

Overall, the experiments provided valuable insights into the behaviour of the model under different conditions. The results demonstrated the importance of considering demand levels, fixed order costs, and the trade-off between lost sales costs and outdate costs when optimizing supply chains. These findings contribute to a better understanding of the model's performance and can guide decision-making in supply chain management practices in real-life situations, where the demand is relatively stable.

Future research should focus on incorporating stochastic demand into the model to better capture the uncertainty associated with real-life demand scenarios, thereby improving its representativeness. Furthermore, alternative solving algorithms can be employed to address the inventory problem, enabling a comparative analysis with the B&B algorithm to assess the effectiveness of different approaches.

7. References

Ali, S. S., Barman, H., Kaur, R., Tomaskova, H., & Roy, S. K. (2021). Multi-Product Multi Echelon Measurements of Perishable Supply Chain: Fuzzy Non-Linear Programming Approach. *Mathematics*, *9*(17), 2093. https://doi.org/10.3390/math9172093

Alrefaei, M. H., & Diabat, A. H. (2009). A simulated annealing technique for multi-objective simulation optimization. *Applied Mathematics and Computation*, *215*(8), 3029–3035. https://doi.org/10.1016/j.amc.2009.09.051

Attar, A., Raissi, S., & Khalili-Damghani, K. (2016, May 10). Simulation–optimization approach for a continuous-review, base-stock inventory model with general compound demands, random lead times, and lost sales. *SIMULATION*, *92*(6), 547–564. https://doi.org/10.1177/0037549716644055

Andersson, J., & Marklund, J. (2000, December). Decentralized inventory control in a two-level distribution system. *European Journal of Operational Research*, *127*(3), 483–506. https://doi.org/10.1016/s0377-2217(99)00332-x

Andersson, J., & Melchiors, P. (2001). A two-echelon inventory model with lost sales. *International Journal of Production Economics*, *69*(3), 307–315. https://doi.org/10.1016/s0925-5273(00)00031-1

Bhattacharya, R., & Bandyopadhyay, S. (2010). A review of the causes of bullwhip effect in a supply chain. *The International Journal of Advanced Manufacturing Technology*, *54*(9–12), 1245–1261. https://doi.org/10.1007/s00170-010-2987-6

Broekmeulen, R. a. C. M., & Van Donselaar, K. K. (2009). A heuristic to manage perishable inventory with batch ordering, positive lead-times, and time-varying demand. *Computers & Operations Research*, *36*(11), 3013–3018. https://doi.org/10.1016/j.cor.2009.01.017

Chen, L. M., & Sapra, A. (2013). Joint inventory and pricing decisions for perishable products with two-period lifetime. *Naval Research Logistics (NRL)*, *60*(5), 343–366. https://doi.org/10.1002/nav.21538

Chaudhary, V., Kulshrestha, R., & Routroy, S. (2018, March 15). State-of-the-art literature review on inventory models for perishable products. *Journal of Advances in Management Research*, *15*(3), 306–346. https://doi.org/10.1108/jamr-09-2017-0091

Chopra, S., & Meindl, P. (2019). Supply chain management: Strategy, planning, and operation. Boston, MA: Pearson.

Claassen, G. D. H., Gerdessen, J. C., Hendrix, E. M., & van der Vorst, J. G. (2016). On production planning and scheduling in food processing industry: Modelling non-triangular setups and product decay. Computers & Operations Research, 76, 147-154. https://doi.org/10.1016/j.cor.2016.06.017

Clark, A.J., & Scarf, H. (1960) Optimal Policies for a Multi-Echelon Inventory Problem. Management Science, 6, 475-490. https://doi.org/10.1287/mnsc.6.4.475

Cohen, M. A., & Lee, H. L. (1988). Strategic Analysis of Integrated Production-Distribution Systems: Models and Methods. *Operations Research*, *36*(2), 216–228. https://doi.org/10.1287/opre.36.2.216

Cohen, M. A., Kamesam, P. V., Kleindorfer, P. R., Lee, H. L., & Tekerian, A. (1990). Optimizer: IBM's Multi-Echelon Inventory System for Managing Service Logistics. *Interfaces*, *20*(1), 65–82. https://doi.org/10.1287/inte.20.1.65

Cohen, M. A., & Moon, S. (1991). An integrated plant loading model with economies of scale and scope. *European Journal of Operational Research*, *50*(3), 266–279. https://doi.org/10.1016/0377-2217(91)90260-3

Christou, I., Skouri, K., & Lagodimos, A. (2020). Fast evaluation of a periodic review inventory policy. *Computers &Amp; Industrial Engineering*, *144*, 106389. https://doi.org/10.1016/j.cie.2020.106389

Daniel, J. S. R., & Rajendran, C. (2005). A simulation-based genetic algorithm for inventory optimization in a serial supply chain. *International Transactions in Operational Research*, *12*(1), 101–127. https://doi.org/10.1111/j.1475-3995.2005.00492.x

Datta, R., & Regis, R. G. (2016). A surrogate-assisted evolution strategy for constrained multi-objective optimization. *Expert Systems With Applications*, *57*, 270–284. https://doi.org/10.1016/j.eswa.2016.03.044

De Kok, T., Grob, C., Laumanns, M., Minner, S., Rambau, J., & Schade, K. (2018). A typology and literature review on stochastic multi-echelon inventory models. *European Journal of Operational Research*, *269*(3), 955–983. https://doi.org/10.1016/j.ejor.2018.02.047

Dolgui, A., Tiwari, M. K., Sinjana, Y., Kumar, S. K., & Son, Y. J. (2017). Optimising integrated inventory policy for perishable items in a multi-stage supply chain. *International Journal of Production Research*, *56*(1–2), 902–925. https://doi.org/10.1080/00207543.2017.1407500

Duong, L. N., Wood, L. C., & Wang, W. Y. (2015). A Multi-criteria Inventory Management System for Perishable & Substitutable Products. Procedia Manufacturing, 2, 66–76. https://doi.org/10.1016/j.promfg.2015.07.012

Ekanayake, N., Joshi, N., & Thekdi, S. A. (2016). Comparison of single-echelon vs. multi-echelon inventory systems using multi-objective stochastic modelling. *International Journal of Logistics Systems and Management*, *23*(2), 255. https://doi.org/10.1504/ijlsm.2016.073971

Federgruen, A., & Zheng, Y. (1992). An Efficient Algorithm for Computing an Optimal (r, Q) Policy in Continuous Review Stochastic Inventory Systems. *Operations Research*, *40*(4), 808–813. https://doi.org/10.1287/opre.40.4.808

Forrester, J. (1958). Industrial Dynamics A Major Breakthrough for Decision Makers. *Harvard Business Review*, *36*(4), 37–66.

Forsberg, R. (1997, January). Exact evaluation of (R, Q)-policies for two-level inventory systems with Poisson demand. *European Journal of Operational Research*, *96*(1), 130–138. https://doi.org/10.1016/s0377-2217(96)00137-3

Gholami-Zanjani, S. M., Jabalameli, M. S., & Pishvaee, M. S. (2021, February). A resilient-green model for multi-echelon meat supply chain planning. *Computers &Amp; Industrial Engineering*, *152*, 107018. https://doi.org/10.1016/j.cie.2020.107018

Gurobi Optimization, LLC. (2023). *GUROBI OPTIMIZER QUICK START GUIDE*. Retrieved from https://www.gurobi.com/wp-content/plugins/hd_documentations/documentation/10.0/quickstart_windows.pdf

Herbon, A. (2017). Optimal Pricing and Replenishment of an Expiring Inventoried Product under Heterogeneous Consumer Sensitivities. *Decision Sciences*, *49*(3), 522–552. https://doi.org/10.1111/deci.12276

Hill, R., Seifbarghy, M., & Smith, D. (2007, September). A two-echelon inventory model with lost sales. *European Journal of Operational Research*, *181*(2), 753–766. https://doi.org/10.1016/j.ejor.2006.08.017

Jackson, I., Tolujevs, J., & Reggelin, T. (2018). The Combination of Discrete-Event Simulation and Genetic Algorithm for Solving the Stochastic Multi-Product Inventory Optimization Problem. *Transport and Telecommunication Journal*, *19*(3), 233–243. https://doi.org/10.2478/ttj-2018-0020

Jaigirdar, S. M., Das, S., Chowdhury, A. R., Ahmed, S., & Chakrabortty, R. K. (2022). Multi-objective multi-echelon distribution planning for perishable goods supply chain: a case study. *International Journal of Systems Science: Operations & Logistics*, *10*(1). https://doi.org/10.1080/23302674.2021.2020367

Köchel, P., & Nieländer, U. (2005, January). Simulation-based optimisation of multi-echelon inventory systems. *International Journal of Production Economics*, *93–94*, 505–513. https://doi.org/10.1016/j.ijpe.2004.06.046

Kostikas, K., & Fragakis, C. (2004). Genetic Programming Applied to Mixed Integer Programming. *Springer Berlin Heidelberg eBooks*, 113–124. https://doi.org/10.1007/978-3-540-24650-3_11

Kouki, C., Jemaï, Z., & Minner, S. (2015). A lost sales (r, Q) inventory control model for perishables with fixed lifetime and lead time. *International Journal of Production Economics*, *168*, 143–157. https://doi.org/10.1016/j.ijpe.2015.06.010

Kouki, C., Jemai, Z., Sahin, E., & Dallery, Y. (2014). Analysis of a periodic review inventory control system with perishables having random lifetime. *International Journal of Production Research*, *52*(1), 283–298. https://doi.org/10.1080/00207543.2013.839895

La Fata, C. M., & Passannanti, G. (2017). A simulated annealing-based approach for the joint optimization of production/inventory and preventive maintenance policies. *The International Journal of Advanced Manufacturing Technology*, *91*(9–12), 3899–3909. https://doi.org/10.1007/s00170-017-0053-3

Lagodimos, A., Christou, I., & Skouri, K. (2012b). Computing globally optimal (s,S,T) inventory policies. *Omega*, *40*(5), 660–671. https://doi.org/10.1016/j.omega.2011.12.008

Lee, C. C. (2006). Two-warehouse inventory model with deterioration under FIFO dispatching policy. *European Journal of Operational Research*, *174*(2), 861–873. https://doi.org/10.1016/j.ejor.2005.03.027

Lourenção, A., Baptista, E. C., Soler, E. M., Souza, F. G., & Cherri, A. C. (2017). Mixed-integer nonlinear model for multiproduct inventory systems with interior point and branch-and-bound method. *IEEE Latin America Transactions*. https://doi.org/10.1109/tla.2017.7896403

Mitra, S. (2009). Analysis of a two-echelon inventory system with returns☆. *Omega*, *37*(1), 106–115. https://doi.org/10.1016/j.omega.2006.10.002

Noordhoek, M., Dullaert, W., Lai, D. T., & De Leeuw, S. (2018). A simulation–optimization approach for a service-constrained multi-echelon distribution network. *Transportation Research Part E-logistics and Transportation Review*, *114*, 292–311. https://doi.org/10.1016/j.tre.2018.02.006

Pauls-Worm, K. G., Hendrix, E. M., Haijema, R., & van der Vorst, J. G. (2014). An MILP approximation for ordering perishable products with non-stationary demand and service level constraints. *International Journal of Production Economics*, *157*, 133–146. https://doi.org/10.1016/j.ijpe.2014.07.020

Pinkus, C. E. & GEORGE WASHINGTON UNIV WASHINGTON D C PROGRAM IN LOGISTICS. (1971). The Design of Multi-Product Multi-Echelon Inventory Systems Using a Branch-and-Bound Algorithm. *Defense Technical Information Center*, 151. Retrieved from https://apps.dtic.mil/sti/citations/AD0730465

Radhakrishnan, P., Prasad, V. M., & Gopalan, M. R. (2009). Genetic Algorithm Based Inventory Optimization Analysis in Supply Chain Management. *2009 IEEE International Advance Computing Conference*. https://doi.org/10.1109/iadcc.2009.4809047

Sel, Ç., & Bilgen, B. (2015). Quantitative models for supply chain management within dairy industry: a review and discussion. *European Journal of Industrial Engineering*, *9*(5), 561. https://doi.org/10.1504/ejie.2015.071772

Shang, K. H., & Zhou, S. X. (2010). Optimal and Heuristic Echelon (*r*, *nQ*, *T*) Policies in Serial Inventory Systems with Fixed Costs. *Operations Research*, *58*(2), 414–427. https://doi.org/10.1287/opre.1090.0734

Simon, H. A. (1952). On the Application of Servomechanism Theory in the Study of Production Control. *Econometrica*, *20*(2), 247. https://doi.org/10.2307/1907849

Simpson, K. F., “In-process Inventories,” Operations Res., 6 (1958), 863-873.

Sun, H., & Ren, Y. (2005). The impact of forecasting methods on bullwhip effect in supply chain management. *Proceedings. 2005 IEEE International Engineering Management Conference, 2005*. https://doi.org/10.1109/iemc.2005.1559121

Suryawanshi, Y. D., & Hsein, T. (2010, June). *Multi-echelon inventory management for a fresh produce retail supply chain* [Thesis]. Massachusetts Institute of Technology.

Tsai, S. C., Wang, H., & Hung, L. H. (2022). Mixed-integer simulation optimization for multi-echelon inventory problems with lost sales. *Journal of the Operational Research Society*, 1–15. https://doi.org/10.1080/01605682.2022.2141144

Kanchanasuntorn, K., & Techanitisawad, A. (2006, March). An approximate periodic model for fixed-life perishable products in a two-echelon inventory–distribution system. *International Journal of Production Economics*, *100*(1), 101–115. https://doi.org/10.1016/j.ijpe.2004.10.010

Keskinocak, P., & Uzsoy, R. (2010, November 8). *Planning Production and Inventories in the Extended Enterprise: A State of the Art Handbook, Volume 1 (International Series in Operations Research & Management Science, 151)* (2011th ed.). Springer. https://doi.org/10.1007/978-1-4419-6485-4

van der Heijden, M., Diks, E., & de Kok, T. (1999, July 1). Inventory control in multi-echelon divergent systems with random lead times. *OR Spectrum*, *21*(3), 331–359. https://doi.org/10.1007/s002910050093

Xu, G., Feng, J., Chen, F., Wang, H., & Wang, Z. (2019). Simulation-based optimization of control policy on multi-echelon inventory system for fresh agricultural products. *International Journal of Agricultural and Biological Engineering*, *12*(2), 184–194. https://doi.org/10.25165/j.ijabe.20191202.2834

8. Appendix

A) Literature review

In this chapter, a systematic literature review is applied to find what is already researched in the literature about multi-echelon perishable inventory problems and their solving techniques.

We started with collecting literature about multi-echelon models to find appropriate answers for a model that incorporates characteristics of perishable inventory management, without making it too complex to solve within the research time. The papers were searched in Google Scholar and collected using the following keywords: ‘’multi-echelon inventory’’ OR ‘’multi-echelon inventory optimization’’. This resulted in a high outcome of papers, namely 62.400 and 37.800 respectively. Subsequently, literature about perishability is reviewed, mostly with a fixed lifetime within a multi-echelon supply chain. Keywords that were used are: ‘’multi echelon perishable inventory’’, ‘’multi echelon fixed lifetime’’, ‘’multi echelon perishable inventory fixed lifetime’’, ‘’multi echelon inventory optimization perishability’’, ‘’multi echelon inventory optimization fresh products’’. Since the case of lost sales is included in the model, the keywords ‘’lost sales’’ are added to the other keywords using the Boolean connector ‘’AND’’. Then, keywords were identified in those articles and were used to search for more articles. Those keywords include: ‘’Fixed-Life Perishable Problem’’, ‘’multi echelon inventory agricultural products’’.

Thereafter, papers were collected for selecting an appropriate order policy for the echelons. Already in the papers reviewed about perishable multi-echelons models is appropriate information about the selected order policy in that case. Therefore, this information was used again for selecting an appropriate order policy. In addition, papers were searched for an appropriate order policy using the keywords: "order policy perishables multi-echelon", "order policy perishables multi-echelon lost sales", "order policy perishability two-echelon lost sales".

For finding literature about solving techniques, keywords as: ‘’multi echelon inventory problem solving techniques’’ were used. Resulting in 17.500 results, more keywords were added to narrow the view. ‘’Multi echelon inventory problem solving techniques perishability’’, ‘’multi echelon inventory problem solving techniques perishability lost sales’’ were used to search for articles. As articles were reviewed, new keywords were found to search for explicit techniques. Including, ‘’inventory control mathematical programming multi echelon perishability’’, ‘’inventory control genetic algorithm perishability multi-echelon’’, and ‘’branch and bound inventory optimization’’

During the search phase, a large amount of papers appeared when searching with the keywords, especially in the case of finding a structure for the multi-echelon model. Hence, papers were sorted by relevance in Google Scholar. Based on reading the titles and abstracts of the papers on the first few pages in Google Scholar, it was concluded that after an amount of papers on the first few pages, articles were not in the scope of this research anymore and the search for articles was stopped.

B)

	W	**1**	**2**	**3**	**4**	**5**	**Average**
FR (%)	100	99.79	99.62	99.44	99.24	99.32	99.48
SL (%)	100	98.5	98.5	97.5	97	98	97.9
W (units)	0	8	4	10	1	12	7
Q (units)	50						
TC (€)	69480						
OG (%)	2.78						

	W	1	2	3	4	5	Average
FR (%)	100	96.68	98.00	95.85	97.96	96.27	96.95
SL (%)	100	91.5	92.5	86.5	91.5	87.5	89.9
W (units)	0	0	0	0	0	0	0
Q (units)	48						
TC (€)	70838						
OG (%)	5.79						

	W	1	2	3	4	5	Average
FR (%)	100	99.85	99.64	99.81	99.75	99.68	99.75
SL (%)	100	99.5	99.5	99.5	99	98.5	99.2
W (units)	0	4	11	2	5	0	4.4
Q (units)	50						
TC (€)	68432						
OG (%)	2.79						

	W	1	2	3	4	5	Average
FR (%)	100	98.63	98.22	98.21	98.49	98.40	98.39
SL (%)	100	94	93.5	94.5	96.5	94.5	94.6
W (units)	0	4	0	0	0	0	0.8
Q (units)	50						
TC (€)	70534.5						
OG (%)	3.32						

	W	1	2	3	4	5	Average
FR (%)	100	99.70	99.54	99.81	100	99.64	99.74
SL (%)	100	99	98	98.5	100	98	98.7
W (units)	0	21	6	27	24	23	20.2
Q (units)	51						
TC (€)	70256.5						
OG (%)	2.56						

C)

	W	1	2	3	4	5	Average
FR (%)	0	0	0	0	0	0	0
SL (%)	0	0	0	0	0	0	0
W (units)	0	0	0	0	0	0	0
Q (units)	0						
TC (€)	7000						
OG (%)	0						

D)

	W	1	2	3	4	5	Average
FR (%)	0	0	0	0	0	0	0
SL (%)	0	0	0	0	0	0	0
W (units)	0	0	0	0	0	0	0
Q (units)	0						
TC (€)	56397						
OG (%)	0						