Erasmus University Rotterdam

Erasmus School of Economics

Master Thesis Econometrics and Management Science

Operations Research and Quantitative Logistics

Developing Blends between an Item Approach and a System Approach in Spare Parts Management

Author

Y.S. Savli BSc (621791)

Supervisor

Prof. dr. ir. R. Dekker

Internship Supervisor

ir. W. Heijnen

Second assessor

Dr. O. Vicil

August 14, 2023

Gordian Logistic Experts

Erasmus Universiteit Rotterdam

Preface

With this research, performed for both Gordian and the Erasmus School of Economics, I was able to contribute to an optimisation problem which was a realistic problem felt in business. Performing this research, was a big challenge for me. Along the way, I learned a lot about Econometrics, but above all, I got to know myself.

Since I was a little girl, my dream was to make an impact by fixing unsolvable civil challenges. I wanted to combine my desire to contribute to societal problems with my passion for mathematics and solving complex problems. Having successfully finished my Master's Thesis, I am happy to close this long chapter of studies in Rotterdam and Delft. Now I can confidentially say that I have the required 'analytic mindset'.

Therefore, I want to take this opportunity to thank everyone that has joined my journey. First and foremost I want to thank my supervisor: Prof. Dekker from ESE for his support during my Thesis period. He has guided me and helped me get through the tough parts. Also a special thanks to the whole of the ESE Academic Support Team, which were there for guidance and feedback during the thesis project. Furthermore, I want to thank Dr. Vicil for his final assessment of the research.

This journey, which started in Maastricht and ended in Rotterdam with a detour in Delft, was not made possible without my dear friends and family. When I was struggling the most I felt their warmth in my heart. Therefore, I hope that I have made them proud by obtaining my Master's degree and that I can somehow pay them back for all their love and support. On that account, I want to thank my mother Betül, father Yaşar, (not so) little brother Furkan and aunt Ayşe. But also much gratitude and appreciation towards my parents-in-law and all the other relatives, family and close friends whom I want to thank!

Last but not least, I want to thank my husband Yavuz, for his love and support during this whole journey. He made sure to motivate, guide and assist me wherever possible and therefore I want to thank him deeply!

Yasemin Simge Cinek-Şavlı

Delft, August 14, 2023

Abstract

Determining base stock levels can be a challenging task in spare parts management. This research contains several approaches to determine base stock levels under a target service level. We distinguish between approaches with a system availability target and an item availability target, which are called a system approach and an item approach, respectively. Two types of blends between these approaches under fill rate targets are developed within this research, referred to as the Basic Blend Approach (BBA) and the Advanced Blend Approach (ABA). In the BBA we apply a system approach on only a subset of items in a system. This results in lower investment costs for mainly expensive slow-moving items. The ABA applies a newly introduced algorithm to determine optimal class availability targets, whilst satisfying a system availability target, for items divided into classes based on their price and demand frequency per year. This approach is easy to interpret and apply and obtains low stock levels for expensive slow-moving items.

Nomenclature

Abbreviations & Acronyms

ABA	Advanced Blend Approach
BBA	Basic Blend Approach
CA	Class Approach
IA	Item Approach
FR	Fill rate
$Gordian$	Gordian Logistics Experts B.V.
$METRIC$	Multi-Echelon Technique for Recoverable Item Control
MIA	Multi-Item Approach
LS	Local Search
SKU	Stock Keeping Unit

Symbols

€	Euro
μ_i	Average yearly demand of SKU i
BO_i	A random variable indicating the number of back orders of SKU i
$C(\mathbf{S})$	The cost function denoted by $\sum_{i\in I} C_i(S_i)$
c_i	Purchase price of SKU i
$C_i(S_i)$	Investment cost of SKU i under base stock S_i, denoted by $c_i S_i$
$\mathbb{E}BO^{obj}$	Target level for $\mathbb{E}BO(\mathbf{S})$
$\mathbb{E}BO_i(S_i)$	Mean number of back orders of SKU i
$\mathbb{E}BO(\mathbf{S})$	Aggregate mean number of back orders
FR^{obj}	Target level for $FR(\mathbf{S})$

$FR_i(S_i)$	Item fill rate of SKU i
$FR(\mathbf{S})$	Aggregate system fill rate
OH_i	Random variable indicating the stationary on hand stock for item i
I	Set of SKUs
$\|I\|$	Number of SKUs
L_i	Average lead time of SKU i
M	Total demand of system, so $M = \sum_{i \in I} \mu_i$
$\mathbf{S}$	Vector consisting of all base-stock levels denoted by $(S_1, S_2, \ldots, S_{\|I\|})$
$\mathcal{S}$	Solution space denoted by $\{\mathbf{S} \mid S_i \in \mathbb{N}_0, \forall i \in I\}$
S_i	Base-stock level for SKU i, with $i \in I$ and $S_i \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$
$S-1, S$	One-for-one replenishment policy
X_i	Random variable indicating the stationary number of copies ordered of item i

Contents

1 Introduction

Service logistics can be defined as the controlling part of the service chain. The service chain is a series of services offered for maintenance, repair and disposal of purchases from purchase to the end of life. The logistics component ensures that people, resources and materials in the service chain are available at the right place at the right time. However, resource and inventory management can be a challenging task. The task becomes even more challenging when the components in the inventory are not moved frequently, consume a large space and/or are very expensive per unit. Such components are called 'slow-movers'. In spare parts management one deals quite often with these slow-movers, being one of its most complex elements.

Inventory management of slow-movers can be challenging for a few reasons. To understand these challenges, it is important to underline that slow-movers are usually very expensive. The inventory ties up working capital, storage space, and other resources that can be used more effectively elsewhere. The indicated increases the holding costs and reduces the overall profitability of the business. The holding cost also called the inventory or storage costs, are the costs related to storing inventory. Hence, a good base stock level can prevent any overstocking or understocking. Another cause that makes the inventory management of slow-movers challenging, is that these items often have a high degree of demand variability, which makes it difficult to accurately forecast future demand. Aforementioned can also lead to the same understocking or overstocking behaviour, leading to inventory imbalances. Hence, much research has focused on the inventory management of slow-moving items. Within this thesis, the goal is to minimise the total investment costs subject to constraints of expected back orders or the fill rate levels. The total investment costs are determined by the cost function consisting of the sum of the base stock level per item multiplied by the item's purchase price.

The current research is part of a thesis internship at Gordian Logistic Experts B.V. (Gordian). Gordian is a logistics and supply chain consulting firm, specialising in spare parts management. They provide a wide range of services to help companies optimise their logistics operations management and supply chain management. A lot of their clients are interested in minimising the stock levels of their spare parts while maintaining high service levels. Gordian uses the single-item single-echelon method as explained in Sherbrooke (2006) to achieve those objectives.

In this research, this method is called the 'Class Approach', and is explained in Section 4.3. The main idea of the Class Approach is that all parts are classified according to their average demand frequency and price. In consultation with their client, Gordian sets target service levels for each class. After classifying Gordian uses an item approach as explained in Van Houtum and Kranenburg (2015) to determine the minimum stock level of each item to achieve its class target service level. In this research, this method is referred to as the 'Item Approach' and is explained in Section 4.2.

One popular request of Gordian's clients is minimising their stock levels while maintaining an overall high service level for a system of items, for example, a landing system of an aeroplane. In this research, this problem is referred to as the 'Multi-Item Problem' and is explained in Section 4.1 following Van Houtum and Kranenburg (2015). In Section 4.4, this problem is solved using a Greedy-based approximation algorithm and we call this approach the 'Multi-Item Approach'. In literature, solving the Multi-Item Problem is also known as a 'System Approach'. An advantage of the Multi-Item Approach is that it allows low stock levels for expensive slow-movers and high stock levels for cheap fast-movers. Hence, it leads to lower total costs. However, this approach is much more complex and harder to implement. Furthermore, when the price gap of the items in a system is large, the Multi-Item Approach tends to keep no stock of the expensive items.

Contrary to the Multi-Item Approach, the Class Approach calculates the base stock for just one item giving a target service measure for one item. Furthermore, the Class Approach is flexible, faster and easy to implement as the items do not depend on the availability of other items. However, for expensive slow-movers, this is a rather harsh approach. It may lead to high inventory costs, as higher safety stock levels are maintained to ensure no stock-outs occur.

Hence, the goal of this thesis is to create a blend between these approaches to get the best of both worlds. In Section 4.5, two different blends are introduced. The first one is referred to as the 'Basic Blend Approach'. The goal of the Basic Blend Approach is to lower the costs for the very expensive and very slow-moving items compared to the Class Approach, while keeping their stock levels (mostly) more than zero. The second blend is referred to as the 'Advanced Blend Approach'. The goal of this blend is to determine optimal class target service levels while maintaining the system target service level. This approach is a great way of combining the currently used Class Approach at Gordian and the Multi-Item Approach.

Chapters overview

In the remainder of this thesis, we start with a detailed problem description in Chapter 2. This chapter includes the research questions and all assumptions necessary for the research. Thereafter, a literature review is given in Chapter 3, containing related work. In Chapter 4 the relevant econometric and mathematical methods are explained. Hereafter, in Chapter 5 an analysis of the data used in this thesis is given. The numerical results are given in Chapter 6. And this thesis is concluded with a conclusion and discussion in Chapters 7 and 8, respectively.

2 Problem Description

One main problem in spare parts management is finding the best trade-off between holding inventories of parts, which involves costs, and being able to repair failures quickly, hence avoiding high downtime costs. As it is difficult to assess downtime costs for individual spare parts, companies usually restrict themselves to availability targets for individual or groups of spare parts. The issue is to determine which targets to use. This is difficult as there may be many distinct parts and both the total investment costs as well as the system availability are nonlinear functions of the number of parts in stock.

At this moment Gordian uses a method, which we refer to as the 'Class Approach' to determine the stock levels for parts. In this approach for every part, an availability target is set based on a parts price and a demand frequency, after which the required stock level is determined. This is a harsh approach for expensive slow-movers since the stock levels are likely to be high, which leads to higher total costs. Considering the Multi-Item Problem could be a solution for the high stock levels of the expensive slow-mover. We refer to the approach to solve the Multi-Item Problem as the 'Multi-Item Approach'. A disadvantage of this approach is that the stock of items with a very low demand frequency and a high cost is likely to be set to zero. And the stock of items with very high demand and very low costs is likely to be set to a (too) high number.

Hence, that is why Gordian has not implemented a Multi-Item Approach yet and is interested in a blend between the two approaches.

Research questions

The goal of this research is to develop a blend between an Item Approach and a Multi-Item Approach for spare parts inventory control. We aim to minimise the total investment costs subject to expected back orders and fill rate levels, leading to the following research questions.

Main research question:

How can we develop a blend between the single-item approach and a system approach in spare parts management?

To answer the main research question, we will answer the following sub-research questions first:

(i) What are item and system approaches for spare parts?

(ii) What are different methods to develop a blend between an item approach and a system approach?

(iii) Which method is the most suitable and the most relevant for clients of Gordian Logistic Experts B.V.?

Research strategy

To achieve the answers to the research questions, we first have to research the existing optimisation models in the literature. Once we have enough background knowledge, we can select the methods, which are most relevant for this research. In combination with the already existing models, we will create new models to be able to answer the main research question. In our case, the new model will be a blend between the Item Approach and the Multi-Item Approach.

To evaluate the performance of the models we need a data set on which we can apply these models. Therefore, we first need to clean the data set provided by Gordian. Then, we will analyse the data to get the right information necessary for the models.

Terminology

Throughout this paper, a few more terms are used for a spare part, specifically *item*, *part* and *Stock Keeping Unit (SKU)*. The latter is a term commonly used in inventory management, supply chain management, and retail operations to track and manage products at various stages of the supply chain. It refers to all items that are stocked in case of failure or replacement.
A *system* is considered to be an assembly of parts in for example a machine or aeroplane. So, the components of a system are called the SKUs of a system.
Multiple service measures exist to measure the availability of a part. The most common and interpretable service measure widely used in practice is the *fill rate* (FR). The FR is the fraction of demand that can be satisfied immediately from stock on hand. Another common service measure is the *expected back orders*, which is used in literature more often. Hence, these are the service measures applied in this thesis.

Assumptions

To answer the research questions, together with Gordian we summarise the assumptions of the problem. These assumptions are common for these types of problems. In the literature review in the next section, we will explain which assumptions are made in which articles.

Assumption 1. *The problem is a single-echelon problem. A single-echelon problem refers to a situation with only one level or layer of inventory in the supply chain. This means there is only one inventory holding location, such as a warehouse or distribution centre, between the supplier and the end customer.*

Assumption 2. *A continuous infinite time horizon* $[0, \infty)$ *is considered.*

Assumption 3. *Demand for different items occurs according to independent Poisson processes as explained in Feeney and Sherbrooke (1966). The demand rate is known and constant.*

Assumption 4. *The lead times for different items are independent.*

Assumption 5. *The cost function, which is to be minimised, is determined by the cost function consisting of the sum of the base stock level per item multiplied by the item's purchase price. Other costs are ignored due to the base-stock model.*

Assumption 6. *A repair-by-replacement policy is applied for all items. This policy is a maintenance strategy whereby a component or system is replaced entirely when it fails, rather than repair or refurbishment attempts. This policy is often used for components or systems that are critical to the operation of a larger system and that have a high likelihood of failure.*

Assumption 7. *The mean demand, the lead time and the price of all items are known.*

Assumption 8. *All items are considered equally critical. An item is critical when its failure causes the entire system to fail.*

Assumption 9. *For individual spare parts, a one-for-one replenishment model also called the* $(S-1, 1)$*-inventory control model, is applied. For this model, the goal is to maintain a constant inventory level by restocking exactly what is sold. This means that for every unit of a product that is sold, one unit is replenished, keeping the inventory at a consistent level. In this case, ordering costs are determined by the demand and can therefore be left out of the optimisation.*

3 Literature Review

The literature on spare parts inventory policies is rich. Many papers have been published. The interested reader is referred to the comprehensive general reviews on spare parts management in Silver et al. (1998), Sherbrooke (2006), Axsäter (2015) and Bounou et al. (2017).

To select papers relevant to the research questions we first looked for review papers on the topic, next we did a Google keyword search and finally, we did a cited reference search for papers quoting the papers we found so far. As keywords, we used combinations of multi-item approach, system approach for fast and slow-movers and system availability constraints together with spare parts. An important reference turned out to be the book by Van Houtum and Kranenburg (2015). They provide a detailed description of the multi-item problem with system availability targets. The main service availability measures they use are the expected back orders. However, in ”Section 2.7.4” they also consider the fill rate as an availability measure.
In the following sections, we provide the results of our literature research, starting with a general overview of spare parts inventory control. Thereafter, we focus on the different methods used in this research in the order they appear, namely: Multi-Item Approach, Item Approach and Class Approach. As for the blends, no literature has been used.

General overview

A common assumption of inventory control models for spare parts, is that spare parts have small demand rates and high item costs. This allows for the use of an $(S-1, S)$ inventory policy, also called a one-for-one replenishment policy. A one-for-one replenishment policy means that every item is replenished as soon as failure occurs, trying to keep an inventory position of S. The most well-known study where the $(S-1, S)$-policy is described in the context of spare parts management is the work by Feeney and Sherbrooke (1966). They assume a compound Poisson demand distribution with the expected back orders as a service measure. In their article, they minimise total costs which are based on an estimation of the holding costs and stock performance costs. The authors Smith and Dekker (1997) describe the $(S-1, S)$ where the demand does not occur according to a Poisson process. Instead, they assume that the demand follows a renewal process with deterministic lead times. They conclude that the frequency of orders depends on the variance of inter arrival times.

The book by Silver et al. (1998) introduces general inventory control models in production planning and scheduling. This work contains all relevant formulas for different types of demand distributions relevant to this research.

Different types of service measures are defined in the book by Axsäter (2015). Amongst the service measures defined and used in this research are: the expected back orders, the product fill rate and the system fill rate. These service measures are defined for continuous demand distributions and discrete distributions, hence in our case, we consider that our demand is Poisson distributed. We use the same definitions and terminology for expected back orders and fill rate as Axsäter (2015). Assuming that all parts in a system are equally critical, Sherbrooke (1992) showed that maximising the system fill rate is approximately equivalent to minimising the sum of expected back orders. Hence, in this research, we use both the expected back orders and the fill rate as service measures.

Multi-Item Approach

One of the first references of the Multi-Item approach, in literature, also referred to as the System Approach, is by Karush (1957). The Multi-Item approach is the approach to solving a multi-item problem with system availability targets. He presented a Greedy based method, to maximise the fill rate of a system under budget constraints, whereby our constraint is to satisfy a target fill rate. A similar approach is presented by Rustenburg et al. (1998) in his post-doctoral research.

The most well-known paper is presented by Sherbrooke (1968). He describes the multi-item two-echelon problem, where he considers one central warehouse and multiple local warehouses. This mathematical model is known as a METRIC model (Multi-Echelon Technique for Recoverable Item Control) in literature. The Multi-Item Approach in our research is similar to the METRIC model of Sherbrooke (1968) with single-echelon, where system investment costs are minimised under service level targets. The Greedy algorithm to solve the multi-item problem used in this research is based upon the Greedy-algorithm presented by Sherbrooke (2006), whereby our research is based upon the book of Van Houtum and Kranenburg (2015). Sherbrooke (2006) refers to this algorithm as the 'Mariginal Analysis'. The multi-item problem is a nonlinear integer programming problem. Besides the Greedy algorithm, different techniques and approximation algorithms exist to get feasible solutions to this complex problem. Specifically, the Lagrangian relaxation and the Dantzig-Wolfe decomposition are next to the Greedy method the most popular methods for the multi-item problem.

One of the earliest papers about the Lagrangian Relaxation is by Everett III (1963). He concludes that the Lagrange method is useful because it reduces the optimisation problem to unconstrained and independent maximisation problems, which are easier to solve. For more insights to this method, the reader is referred to the paper by Fisher (1981). The main idea of the Dantzig-Wolfe decomposition is that it decomposes large-scale optimisation problems into smaller subproblems, yielding low computational effort. In his post-doctoral research, Basten and van Houtum (2014) uses this method for the multi-item problem with system availability constraints. But we restrict ourselves to the Greedy algorithm, since it is the easiest to apply and interpret.

Item Approaches

An item approach is an approach where we determine base stock levels for items with an individual availability target. The base stock levels can easily be determined by stochastic formulas presented by for example Axsäter (2015) and Silver et al. (1998). However, the item approach with expected back orders as service level presented in this research follows the item approach in Van Houtum and Kranenburg (2015). The individual item target service levels are defined such that a system target level is reached. They present an algorithm to determine the base stock levels with a step-by-step approach. The main idea is that the base stock levels are all set to zero and in each iteration, the base stock levels are increased by one until the target service level is reached. A similar approach is presented in the book of Axsäter (2015). In the item approach explained by Thonemann et al. (2002) all items have the same individual fill rate and they refer to this approach as the item approach with constant fill rate policy. The item approach with fill rate as service level presented in this paper follows the same approach. In their article, Kiesmüller et al. (2011) also cover a single item approach, however they determine minimum order quantities instead of minimum base stock levels.

Class Approaches

We refer to the Class Approach as an item approach where items are classified according to their demand and price. Afterwards, for each class, a service level target is set and the base stock levels of the items are determined based on their corresponding class target. In the book of Silver et al. (1998), they define an "ABC-Classification matrix", which is a 3×3- matrix according to price and demand.

In an article by Porras and Dekker (2008) classification is applied by considering classes based on price, demand and criticality. The Class Approach as used by Gordian and also in this research, is based upon the "ABC-Classification matrix" in the work of Silver et al. (1998). The authors Cardós et al. (2015) compare different types of classification for the multi-item problem. Based on numerical experiments they conclude that the ABC-classification with one or two criteria, being the demand and/or price of items are the most suitable classification techniques for practitioners.

4 Methodology

In this chapter, all the theoretical methods are explained to achieve the answers to the research questions of this thesis. The chapter starts with Section 4.1 where all notation used in this thesis are introduced and the problem is formulated mathematically. Next, the concept of an Item Approach is explained in Section 4.2 and an extension of the Item Approach, which is called the Class Approach is explained in Section 4.3. After that, in Section 4.4, the Multi-Item Approach is explained. This is the approach for solving the Multi-Item Problem. And finally, two methods to create a blend between the Item Approach and the Multi-Item Approach are introduced in Section 4.5.

4.1 Notation and Mathematical Formulation

In this section, the notation of the variables is introduced following Van Houtum and Kranenburg (2015). Furthermore, this section contains a mathematical formulation of the problem. All variables are displayed in Table 1. The authors Van Houtum and Kranenburg (2015) refer to the critical components of a system as Stock Keeping Units (SKUs). But in general, all items that are stocked, are called SKUs in inventory management.

Assume that we have a set of SKUs I, with the number of SKUs denoted by $|I|$. To make it convenient for notation, each SKU $i \in I$ will be numbered as $i = 1, 2, \ldots, |I|$.

The goal is to determine a minimum base-stock level for each SKU $i \in I$. The base-stock level for a part is the minimum quantity of that part that a company needs in stock or on order to meet its demand during lead time. It is the inventory level that is maintained when no new orders are placed for that item. The base-stock level for an SKU i will be denoted by S_i. We assume that S_i is an integral number greater than or equal to zero, so $S_i \in \mathbb{N}_0$. The vector $\mathbf{S} = (S_1, \ldots, S_{|I|})$ illustrates the base-stock level of all items in I. By Assumption 5 we define the total investment costs $C(\mathbf{S})$ under policy S by:

$$C(\mathbf{S}) = \sum_{i \in I} C_i(S_i) = \sum_{i \in I} c_i S_i, \tag{1}$$

where c_i is the purchase price of SKU i in €'s. Since the total investment costs are to be minimised, the cost function $C(\mathbf{S})$ is the objective function of this problem. The constraint of the problem is to achieve an availability target while minimising the costs. Gordian Logistic Experts B.V. (Gordian) expresses the availability of a part by the fill rate (FR), which is equal to the fraction of demand that is satisfied immediately from stock on hand.

The fill rate is a more common service measure in practice. An alternative measure for availability is the number of back orders BO_i, which is equal to the number of missing parts. The latter also takes the length of unavailability into account. The problem will be defined for both the expected back orders and the fill rate as stated by Van Houtum and Kranenburg (2015).

4.1.1 Expected back orders as Service Level

The aggregate mean number of back orders $\mathbb{E}BO(\mathbf{S})$ in steady state, is denoted by:

$$\mathbb{E}BO(\mathbf{S}) = \sum_{i \in I} \mathbb{E}BO_i(S_i), \tag{2}$$

with $\mathbb{E}BO_i(S_i)$ equal to the mean number of back orders for SKU i. Let $\mathbb{E}BO^{obj}$ be the target level of back orders. Then the mathematical formulation of the current optimisation problem is defined as follows:

$$\begin{aligned} \min \quad & C(\mathbf{S}) \\ \text{s.t.} \quad & \mathbb{E}BO(\mathbf{S}) \leq \mathbb{E}BO^{obj}, \\ & \mathbf{S} \in \mathcal{S}, \end{aligned} \tag{3}$$

where $\mathcal{S}$ is the solution space defined by $\{\mathbf{S} \mid S_i \in \mathbb{N}_0, \forall i \in I\}$. Notice that this problem is a nonlinear integer programming problem since the decision variables are integral, the constraints are nonlinear and the objective is a linear function.

Let X_i be the pipeline stock, which is a random variable indicating the stationary number of copies of item i on outstanding orders. Let μ_i and L_i be the mean demand and the lead time of item i, respectively. Then $\mu_i L_i$ is the mean of the pipeline stock X_i.

From Assumption 3, it follows that demand for items occurs according to a Poisson process and each item is an average time of L_i in the pipeline. So the repair or delivery pipeline is an $M|G|\infty$ queueing system. Hence, originated from the book of Van Houtum and Kranenburg (2015) Palm's theorem may be applied which is defined in Palm (1938) as follows:

Palm's Theorem: *If jobs arrive according to a Poisson process with rate λ at a service system and if the times that the jobs remain in the service system are independent and identically distributed according to a given general distribution with mean $\mathbb{E}(W)$, then the steady-state distribution for the total number of jobs in the service system is Poisson with mean $\lambda\mathbb{E}(W)$.*

So using Palm's theorem, it follows that X_i is Poisson distributed. So, the probability distribution of X_i can be denoted as follows:

$$\mathbb{P}[X_i = x] = \frac{(\mu_i L_i)^x}{x!} e^{-\mu_i L_i}, \text{ with } x \in \mathbb{N}_0. \tag{4}$$

Let OH_i be the stock on hand, which is a random variable indicating the number of ready-for-use parts. Following Feeney and Sherbrooke (1966), the stationary distribution of the stock on hand is given by:

$$\mathbb{P}[OH_i = x] = \begin{cases} \sum_{y=S_i}^{\infty} \mathbb{P}[X_i = y], \text{ if } x = 0; \\ \mathbb{P}[X_i = S_i - x], \text{ if } x \in \mathbb{N}, x \leq S_i. \end{cases} \tag{5}$$

Then the stationary distribution of the random variable BO_i, indicating the number of back-ordered demand, is defined by:

$$\mathbb{P}[BO_i = x] = \begin{cases} \sum_{y=0}^{S_i} \mathbb{P}[X_i = y], \ if \ x = 0; \\ \mathbb{P}[X_i = x + S_i], \ if \ x \in \mathbb{N}. \end{cases} \tag{6}$$

The mean back order $\mathbb{E}BO_i(S_i)$, can now be calculated by:

$$\mathbb{E}BO_i(S_i) = \mathbb{E}[BO_i(S_i)] = \sum_{x=S_i+1}^{\infty} (x - S_i)\mathbb{P}[X_i = x], S_i \in \mathbb{N}_0.$$

For computational purposes, we rewrite this equation as:

$$\mathbb{E}BO_i(S_i) = \mu_i L_i - S_i + \sum_{x=0}^{S_i} (S_i - x)\mathbb{P}[X_i = x], S_i \in \mathbb{N}_0. \tag{7}$$

An important lemma to be able to apply the algorithms in the following sections originates from Van Houtum and Kranenburg (2015) and is stated as follows:

Lemma 4.1. *For each SKU $i \in I$, the mean number of back orders $\mathbb{E}BO_i(S_i)$ is decreasing and convex for $S_i \in \mathbb{N}_0$.*

Proof. See Appendix B. □

4.1.2 Fill Rate as Service Level

As mentioned before, Gordian and many other companies use the product fill rate (FR) as a service level. Hence, in this section, we define the problem with FR as service level, again following Van Houtum and Kranenburg (2015). The aggregate FR of a system is the probability that demand with an arbitrary size for the total system of items is satisfied immediately from stock on hand.

Let $FR_i(S_i)$ be the item FR for item i and let $M := \sum_{i \in I} \mu_i$ be the total demand of the system. Then, the aggregate fill rate $FR(\mathbf{S})$ is denoted by

$$FR(\mathbf{S}) = \sum_{i \in I} \frac{\mu_i}{M} \cdot FR_i(S_i). \tag{8}$$

From Assumption 3 it follows that the demands of all items arrive according to a Poisson process. The PASTA property described by Wolff (1982) states that demand arriving arbitrarily, observes the system in a steady state. Hence, the probability that you have positive stock on hand is equal to the probability that the arriving demand is less than the stock level and thus can be satisfied immediately from stock. So, the item fill rate is then defined by

$$FR_i(S_i) = \sum_{x=0}^{S_i - 1} \mathbb{P}[X_i = x]. \tag{9}$$

The new optimisation problem using the FR is now defined as

$$\begin{aligned} \min \quad & C(\mathbf{S}) \\ \text{s.t.} \quad & FR(\mathbf{S}) \geq FR^{obj}, \\ & \mathbf{S} \in \mathcal{S}', \end{aligned} \tag{10}$$

where $\mathcal{S}'$ is the solution space of the problem, which is slightly different than the solution space of Problem 3. The function for the item fill rate $FR_i(S_i)$ has domain $\mathbb{N}_0$ for an item $i \in I$. In the following Lemma originated from Van Houtum and Kranenburg (2015) it is stated that $FR_i(S_i)$ is increasing on $\mathbb{N}_0$ and concave for $S_i \geq \max\{\lceil \mu_i L_i - 1 \rceil, 0\}$. Notice that, $\lceil x \rceil$ denotes the rounded up value for $x \in \mathbb{R}$.

Lemma 4.2. *For each SKU $i \in I$, the item fill rate $FR_i(S_i)$ is increasing on $\mathbb{N}_0$ and concave for $S_i \geq \max\{\lceil \mu_i L_i - 1 \rceil, 0\}$.*

Proof. See Appendix B. □

The average pipeline stock is represented by $\mu_i L_i$. If $\mu_i L_i \leq 1$, then $\max\{\lceil \mu_i L_i - 1 \rceil, 0\} = 0$. Hence, $FR_i(S_i)$ will be concave on its domain. If we have $\mu_i L_i > 1$, then $\max\{\lceil \mu_i L_i - 1 \rceil, 0\} > 0$. Then solutions $\mathbf{S}$ with $S_i \leq \max\{\lceil \mu_i L_i - 1 \rceil, 0\}$ are removed from solution space $\mathcal{S}'$. In general, for slow-moving items, the average pipeline stock will not be greater than one. And if they are, the solutions that are excluded from the solution space have an item fill rate value $FR_i(S_i)$ and are irrelevant for problems with a high target service level.

So it follows that $\mathcal{S}'$ is defined by $\{\mathbf{S} = (S_1, \ldots, S_{|I|}) \mid S_i \geq \max\{\lceil \mu_i L_i - 1 \rceil, 0\}, \forall i \in I\}$.

Table 1: Notation

Notation	
I	Set of SKUs
$\|I\|$	Number of SKUs
S_i	Base-stock level for SKU i, with $i \in I$ and $S_i \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$
$\mathbf{S}$	Vector consisting of all base-stock levels denoted by $(S_1, S_2, \ldots, S_{\|I\|})$
$\mathcal{S}$	Solution space denoted by $\{\mathbf{S} \mid S_i \in \mathbb{N}_0, \forall i \in I\}$
X_i	Random variable indicating the stationary number of copies ordered of item i
OH_i	Random variable indicating the stationary on hand stock for item i
c_i	Purchase price of SKU i
$C_i(S_i)$	Investment cost of SKU i under base stock S_i, denoted by $c_i S_i$
$C(\mathbf{S})$	The cost function denoted by $\sum_{i \in I} C_i(S_i)$
μ_i	Average demand of SKU i
L_i	Average lead time of SKU i
M	Total demand of system, so $M = \sum_{i \in I} \mu_i$
BO_i	A random variable indicating the number of back orders of SKU i
$\mathbb{E}BO_i(S_i)$	Mean number of back orders of SKU i
$\mathbb{E}BO(\mathbf{S})$	Aggregate mean number of back orders
$\mathbb{E}BO^{obj}$	Target level for $\mathbb{E}BO(\mathbf{S})$
$FR_i(S_i)$	Item fill rate of SKU i
$FR(\mathbf{S})$	Aggregate system fill rate
FR^{obj}	Target level for $FR(\mathbf{S})$

4.2 Item Approach

The first and most straightforward approach to determine the base-stock level for all items in the system is to consider each item separately. Each item is assigned its own target service level. We will call this type of approach an 'Item Approach' following Van Houtum and Kranenburg (2015). The advantage of such an approach is that it easily scales up with the number of items, yet on the other hand, it is not clear which targets should be set for the items. In this section, we discuss an Item Approach with both the expected backorders and the fill rate as service level. In the latter all items are set the same item fill rate target equal to the system fill rate target. To meet a target service level one only needs a demand rate and lead time. Hence the price of items is not taken into account, which is a disadvantage. So using the same target service level for all components is likely to be sub-optimal. An example to showcase the methods is given for both service levels.

4.2.1 Expected Back orders as Service Level

In the Item Approach each item is considered separately. One can decompose the constraint of Problem 3, such that each item has its own constraint and the solution is still feasible. Let $\mathbb{E}BO^{obj}$ be the system target service level. Recall the constraint of Problem 3:

$$\sum_{i \in I} \mathbb{E}BO_i(S_i) = \mathbb{E}BO(\mathbf{S}) \leq \mathbb{E}BO^{obj}.$$

If all items i have $\mathbb{E}BO_i(S_i)$ less than or equal to $\frac{\mu_i}{M}\mathbb{E}BO^{obj}$, then $\mathbb{E}BO(\mathbf{S}) = \sum_{i \in I} \mathbb{E}BO_i(S_i) \leq \mathbb{E}BO^{obj}$. So, for each item an item target equal to $\mathbb{E}BO_i^{obj} := \frac{\mu_i}{M}\mathbb{E}BO^{obj}$ is set.

The optimisation problem then becomes:

$$\begin{aligned} \min \quad & C(\mathbf{S}) \\ \text{s.t.} \quad & \mathbb{E}BO_i(S_i) \leq \mathbb{E}BO_i^{obj} \forall_i, \\ & \mathbf{S} \in \mathcal{S}, \end{aligned} \tag{11}$$

This problem can be solved very easily, since it is a straightforward decision problem. The idea is basically to search for each item i the smallest S_i that satisfies the target and that can be done independently from all other items. In Appendix A.1 a pseudo code and an explanation of all the steps is given.

To demonstrate the Item Approach, we give an example in the next section.

Example 1

Consider a system containing 4 items, with its properties displayed in Table 2. Items 1 and 2 have high demand with low and high costs, respectively. And, items 3 and 4 have low demand with low and high costs, respectively. Let $\mathbb{E}BO^{obj}$ be the system target level of expected backorders. Set for each SKU $i = 1, \ldots, 4$ item targets $\mathbb{E}BO_i^{obj} := \frac{\mu_i}{M} \cdot \mathbb{E}BO^{obj}$, with $M = \sum_{i=1}^{4} \mu_i = 55$.

Table 2: **(Example 1)** Properties of a system with 4 items

SKU id	μ_i (per year)	c_i (€)	L_i (years)
1	24	0.10	0.08
2	28	20.40	0.08
3	1	0.12	0.08
4	2	18.11	0.08

We apply the Item Approach for target expected backorders levels 0.1 and 0.05. In Table 3 the solutions are shown for the two target levels. For target expected backorders level $\mathbb{E}BO_i^{obj} = 0.1$ the solution is: $\mathbf{S} = (5, 5, 2, 2)$ with realised expected backorders $\mathbb{E}BO_i(S_i) = (0.019, 0.038, 0.000082, 0.00063)$. And for target level 0.05, the solution is $\mathbf{S} = (5, 6, 2, 2)$ with realised expected backorders $\mathbb{E}BO_i(S_i) = (0.019, 0.011, 0.000082, 0.00063)$. These solutions are achieved by applying Algorithm 3 using the Java Programming Language.

Using Equations 1 and 2 the total number of expected backorders for the system is

$$\mathbb{E}BO(\mathbf{S}) = \sum_{i=1}^{4} \mathbb{E}BO_i(S_i) = \begin{cases} 0.057, \text{ if } \mathbb{E}BO^{obj} = 0.1 \; ; \\ 0.03, \text{ if } \mathbb{E}BO^{obj} = 0.05 \; , \end{cases}$$

with the total investment costs equal to

$$C(\mathbf{S}) = \sum_{i \in I} c_i S_i = \begin{cases} €138,96, \text{ if } \mathbb{E}BO^{obj} = 0.1 \; ; \\ €159,36, \text{ if } \mathbb{E}BO^{obj} = 0.05 \; . \end{cases}$$

It is trivial that the target level $\mathbb{E}BO^{obj}$ is an upper bound to $\mathbb{E}BO(\mathbf{S})$. We denote the gap to the upper bound by $GTT_{\mathbb{E}BO^{obj}}$ and we define $GTT_{\mathbb{E}BO^{obj}}$ by:

$$GTT_{\mathbb{E}BO^{obj}}(\mathbf{S}) := \frac{|\mathbb{E}BO^{obj} - \mathbb{E}BO(\mathbf{S})|}{\mathbb{E}BO^{obj}} \cdot 100\%. \tag{12}$$

Table 3: **(Example 1 - Cont'd)** Solutions Item Approach

	$\mathbb{E}BO^{obj} = 0.1$			$\mathbb{E}BO^{obj} = 0.05$		
SKU id	S_i	$\mathbb{E}BO_i^{obj}$	$\mathbb{E}BO_i(S_i)$	S_i	$\mathbb{E}BO_i^{obj}$	$\mathbb{E}BO_i(S_i)$
1	5	0.0436	0.0185	5	0.022	0.019
2	5	0.0509	0.038	6	0.025	0.011
3	2	0.0018	0.000082	2	0.00091	0.000082
4	2	0.0036	0.00063	2	0.0018	0.00063

Let $\mathbf{S}_{\mathbb{E}BO^{obj}}$ be the solution obtained by applying the Item Approach with target service level $\mathbb{E}BO^{obj}$. In this example we have

$$GTT_{0.1}(\mathbf{S}_{0.1}) = \frac{|0.1-0.057|}{0.1} \cdot 100\% = 43\% \quad \text{and} \quad GTT_{0.05}(\mathbf{S}_{0.05}) = \frac{|0.05-0.03|}{0.1} \cdot 100\% = 40\%.$$

Notice that the gaps are rather high, while we want this gap to be as small as possible. Hence, this is another reason why a system approach could be more suitable for a system of items since we expect the gap to the targets to be a lot lower.

4.2.2 Fill Rate as Service Level

In this section we set the same target fill rate for each item i equal to the system target fill rate, so $FR_i^{obj} := FR^{obj}$ for all $i \in I$. Then, from the definition of the aggregate fill rate the system target fill rate is satisfied as well. The optimisation problem for the Item Approach with the fill rate as service level is:

$$\begin{aligned} \min \quad & C(\mathbf{S}) \\ \text{s.t.} \quad & FR_i(S_i) \geq FR_i^{obj}, \\ & \mathbf{S} \in \mathcal{S}'. \end{aligned} \tag{13}$$

This is again an easy decision problem, which can be solved for each item separately. In Appendix A.1 a pseudocode with explanation is given to solve Problem 13. To demonstrate this approach we continue with the Example of previous section.

Example 1 (*Continued*)

Let FR^{obj} be the system target fill rate for the items in Table 2. Using the Java Programming Language and the algorithm explained in Appendix A.1, we achieve the following solutions for the Item Approach with fill rate as service level.

We apply the Item Approach for target levels 75%, 90% and 90%. In Table 4 the solutions are shown for the different target levels. For target level 75%, the solution is: $\mathbf{S} = (4, 4, 1, 1)$ with realised fill rates $FR_i(S_i) = (0.871, 0.811, 0.923, 0.852)$. For target level 90%, the solution is: $\mathbf{S} = (5, 5, 1, 2)$ with realised fill rates $FR_i(S_i) = (0.954, 0.923, 0.923, 0.988)$. And for target level 90%, the solution is $\mathbf{S} = (7, 7, 2, 3)$ with realised fill rates $FR_i(S_i) = (0.996, 0.992, 0.997, 0.999)$.

Table 4: **(Example 1 - Cont'd)** solutions Item Approach with FR as service level

	$FR^{obj} = 75\%$		$FR^{obj} = 90\%$		$FR^{obj} = 90\%$	
SKU id	S_i	$FR_i(S_i)$	S_i	$FR_i(S_i)$	S_i	$FR_i(S_i)$
1	4	0.871	5	0.954	7	0.996
2	4	0.811	5	0.923	7	0.992
3	1	0.923	1	0.923	2	0.997
4	1	0.852	2	0.988	3	0.999

Using Equations 1 and 8 the aggregate fill rate for the system is

$$FR(\mathbf{S}) = \sum_{i \in I} \frac{\mu_i}{M} \cdot FR_i(S_i) = \begin{cases} 0.841, \text{ if } FR_i^{obj} = 0.75 \ \forall_i; \\ 0.939, \text{ if } FR_i^{obj} = 0.90 \ \forall_i, \\ 0.994, \text{ if } FR_i^{obj} = 0.99 \ \forall_i, \end{cases}$$

with the total costs equal to

$$C(\mathbf{S}) = \sum_{i \in I} c_i S_i = \begin{cases} €100,23, \text{ if } FR_i^{obj} = 0.75] \ \forall_i; \\ €138,84, \text{ if } FR_i^{obj} = 0.90 \ \forall_i, \\ €198,07, \text{ if } FR_i^{obj} = 0.99 \ \forall_i. \end{cases}$$

It is trivial that the target level FR^{obj} is a lower bound to $FR(\mathbf{S})$. We denote the gap to the lower bound by $GTT_{FR^{obj}}$ and we define $GTT_{FR^{obj}}$ by:

$$GTT_{FR^{obj}} := \frac{|FR(\mathbf{S}) - FR^{obj}|}{FR^{obj}} \cdot 100\%. \tag{14}$$

In this example we have

$$\begin{aligned} GTT_{75\%} &= \tfrac{|0.841-0.75|}{0.75} \cdot 100\% &= 12.1\%, \\ GTT_{90\%} &= \tfrac{|0.939-0.90|}{0.90} \cdot 100\% &= 4.3\% \quad \text{and} \\ GTT_{90\%} &= \tfrac{|0.994-0.99|}{0.99} \cdot 100\% &= 0.4\%. \end{aligned}$$

4.3 Class Approach

This section is dedicated to the approach that Gordian Logistic Experts B.V. (Gordian) uses to determine base stock levels. In this approach, items are classified, based on similar characteristics, like demand, price, lead time and priority. Classes can be based on one characteristic or multiple. After classification for each class, a target is set and the Item Approach is applied for all items with their corresponding class targets. We will refer to this approach as the 'Class Approach'.

From personal communication with Jan Willem Rustenburg from Gordian Logistic Experts B.V. (Gordian) follows that the Class Approach is used in practice because it is more interpretable, flexible and easy to implement. However, the choices of thresholds separating the classes and the target fill rates are not theoretically substantiated. Gordian sets the boundaries of the classes in cooperation with their clients. However, the targets per class are determined more intuitively. They always try to satisfy a high total service level and adjust their targets per class manually, until they achieve that level. So the choice of target levels can be optimised. In Section 4.5.2 we introduce an algorithm to determine optimal targets per class.

Classification

Presently Gordian uses classes based on price and demand. Per characteristic three classes are distinguished, using threshold values, based on a set of rules to logically determine the class boundaries. Gordian uses the fill rate (FR) as a service level, thus we will also use the FR as a service level for the Class Approach. Let K_j be class j, with $j = 1, \ldots, 9$. Per class, a fixed FR target $FR^{obj}_{K_j}$ is set, which is applied to each item $i \in K_j$. This manner of classification is also called the ABC-Classification as described in the book of Silver et al. (1998). It is a 3×3-matrix, with the demand frequency classification as its columns and price classification as its rows.

In Figure 1 an example of a classification matrix is displayed. The demand frequency is the average demand requests per year based on the total historical demand. However, the demand can also be classified according to the yearly demand instead of the demand frequency. This is up to the preference of the stakeholders. The price refers to the purchase price per item. The columns are denoted by **A, B** and **C** and the rows are denoted by **1, 2** and **3**. The classes are defined as follows:

A: items with average number of demand requests between 13 and ∞ times per year

B: items with average number of demand requests between 4 and 12 times per year

C: items with average number of demand requests between 1 and 3 per year

1: items with price between 0 and 30 euros

2: items with price between 31 and 500 euros

3: items with price between 500 and ∞ euros

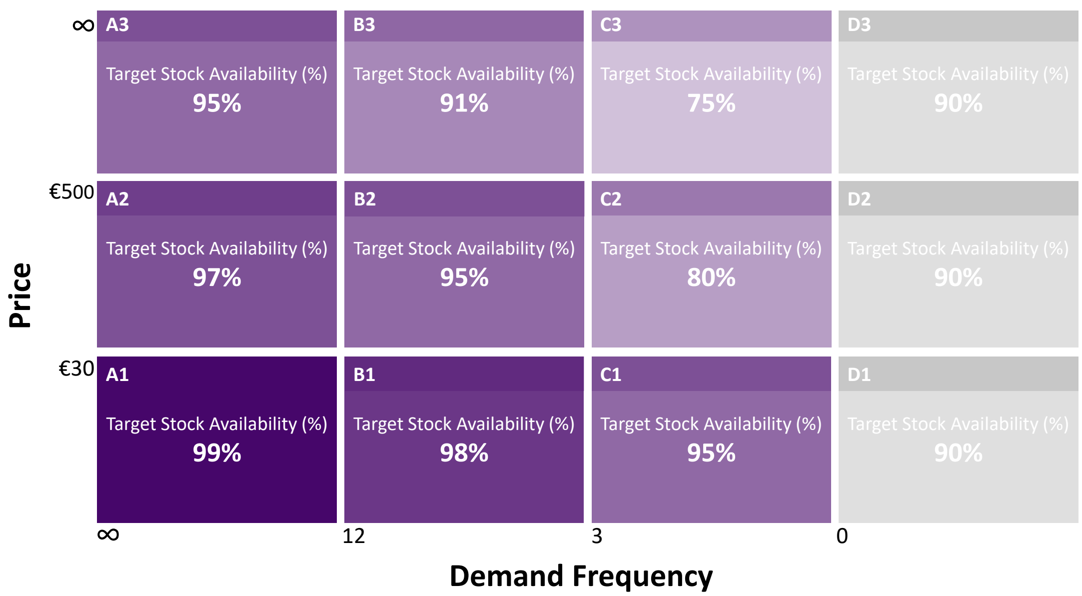


Figure 1: Classification Matrix of Gordian

Gordian also considers an extra column **D** with items with an average number of demand requests per year equal to zero. But this is out of the scope of this research. So only the 3×3-matrix will be considered from now on. The boundaries and targets of this classification matrix are the classification corresponding to the data set we will use in this research analysed in Section 5.

The optimisation problem corresponding to the Class Approach can be defined as follows:

$$\begin{aligned} \min \quad & C(\mathbf{S}) \\ \text{s.t.} \quad & FR_i(S_i) \geq FR_{K_j}^{obj}, \ \forall i \in K_j, \ \forall j = 1, \ldots, 9, \\ & \mathbf{S} \in \mathcal{S}'. \end{aligned} \tag{15}$$

Throughout this thesis, classes $\{K_1, \ldots, K_9\}$ correspond to classes $\{A1, A2, A3, B1, B2, B3, C1, C2, C3\}$. We use K_j with $j = 1, \ldots, 9$ for mathematical formulation.
See Appendix A.2 for a pseudocode of the algorithm to solve Problem 15. To demonstrate the Class Approach we give an example in the following section.

Example 2

Consider a system containing 20 items, with its properties displayed in Table 21 in Appendix D.1. We start with classifying the items based on their demand frequency and price according to the classification matrix of Gordian, as seen in Figure 1. The class fill rate targets are $\{FR_{A1}^{obj}, \ldots, FR_{C_3}^{obj}\} = \{99\%, 97\%, 95\%, 98\%, 95\%, 91\%, 95\%, 80\%, 75\%\}$. Notice that we use the demand frequency f_i only for classification. To determine the fill rate $FR_i(S_i)$ we still use the yearly demand μ_i.

Using Equations 1 and 8 and Algorithm 5, the Class Approach gives aggregate fill rate

$$FR(\mathbf{S}) = \sum_{i \in I} \frac{\mu_i}{M} \cdot FR_i(S_i) = 96.5\%$$

with the total investment costs equal to

$$C(\mathbf{S}) = \sum_{i \in I} c_i S_i = €9,187.99.$$

To make a good comparison with the Item Approach we compute the total stock per class denoted by S_{Class}. In Table 5 the stock levels per class, the total cost and the aggregate fill rate obtained by the Class Approach (CA) and the Item Approach (IA) are compared.

We see that the Class Approach has the lowest total cost concerning the aggregate fill rate compared to the Item Approach with target fill rates 75%, 90% and 99%.

Table 5: **(Example 2)** Total stock per class S_{Class}, the total cost $C(\mathbf{S})$ and the aggregate fill rate $FR(\mathbf{S})$ obtained by Class Approach (CA) compared to the Item Approach ($\text{IA}_{FR^{obj}}$) with different fill rate targets FR^{obj}.

	S_{A1}	S_{A2}	S_{A3}	S_{B1}	S_{B2}	S_{B3}	S_{C1}	S_{C2}	S_{C3}	$C(\mathbf{S})$ (€)	$FR(\mathbf{S})$ (%)
CA	3	6	0	6	6	3	49	4	6	9,187.99	96.5
$\text{IA}_{75\%}$	2	4	0	3	4	2	41	4	6	8,394.38	80.5
$\text{IA}_{90\%}$	2	5	0	4	5	3	47	6	7	9,409.94	92.8
$\text{IA}_{99\%}$	3	7	0	6	7	4	55	8	11	13,863.78	99.2

4.4 Multi-Item Approach

Besides the fill rate for just one item, clients of Gordian are interested in keeping up the fill rate of a whole system. This problem is formulated in Section 4.1 for both the fill rate as well as the expected back orders as a service measure. In this thesis, the approach to solving Problems 3 and 10 is referred to as the 'Multi-Item Approach'. In literature, this approach is also referred to as the 'System Approach'. The essence of a Multi-Item Approach is that a target for a whole system is defined, which can be either a total fill rate target, a constraint of total back orders or a constraint of total inventory value. In this research, the aggregate mean back orders and the aggregate fill rate are both used as target service levels.

The main problem in solving the Multi-Item Problem exactly, is that the constraints are non-linear functions in base-stock levels. Therefore, we need to come up with different mathematical algorithms to solve this problem close to optimal. The authors Van Houtum and Kranenburg (2015) present several approaches in their article, viz. a Greedy algorithm, Lagrangian relaxation and Dantzig-Wolfe decomposition. From Sherbrooke (2006), it follows that in general a Greedy algorithm gives almost the same solutions as the other two methods due to strong similarities. Moreover, the Greedy Algorithm is the easiest to apply and interpret. It is important to notice that the Greedy Algorithm is an approximation algorithm and does not guarantee optimality, but it provides a solution 'close-to-optimal'. Hence, in this research, the Greedy Algorithm presented in Van Houtum and Kranenburg (2015) will be applied.

4.4.1 Greedy algorithm

A Greedy algorithm or a Greedy heuristic is a simple optimisation strategy that makes locally optimal choices at each step with the hope of finding a global optimum. It iteratively selects the best available option at each decision point without reconsidering previous choices. In Appendix, A.3 pseudocodes of the Greedy Algorithm from Van Houtum and Kranenburg (2015) with a detailed description are given for Problems 3 and 10. The main idea of the algorithm is that it searches for the item that has the highest contribution to the service level when the item is increased by one. Recall that an (S-1,S)-policy is applied for all items in a system. The algorithm searches for solutions $\mathbf{S} = (S_1, \dots, S_{|I|})$ which keep the total costs low, but satisfy the target service level. One can also look for the item with the largest contribution per € investment. We demonstrate the Greedy algorithm with both service levels in the following two Examples.

Example 1 (*continued*)

In this example, we consider the system of items displayed in Table 2 again. Applying Algorithm 6 using the Java Programming Language we find the solutions displayed in Table 6.

Table 6: **(Example 1 - Cont'd)** Solutions Multi-Item Approach

	$\mathbb{E}BO^{obj} = 0.1$		$\mathbb{E}BO^{obj} = 0.05$	
SKU id	S_i	$\mathbb{E}BO_i(S_i)$	S_i	$\mathbb{E}BO_i(S_i)$
1	8	2.00e-4	9	4.02e-5
2	5	3.79e-2	6	1.13e-2
3	2	8.20e-5	2	8.26e-5
4	1	1.21e-2	1	1.21e-2

Using Equations 1 and 2 the total number of expected back orders for the system is

$$\mathbb{E}BO(\mathbf{S}) = \sum_{i=1}^{4} \mathbb{E}BO_i(S_i) = \begin{cases} 0.05, & \text{if } \mathbb{E}BO^{obj} = 0.1 \\ 0.02, & \text{if } \mathbb{E}BO^{obj} = 0.05 \end{cases},$$

with the total investment costs equal to

$$C(\mathbf{S}) = \sum_{i \in I} c_i S_i = \begin{cases} €121,15, & \text{if } \mathbb{E}BO^{obj} = 0.1 \\ €141,65, & \text{if } \mathbb{E}BO^{obj} = 0.05 \,. \end{cases}$$

Example 2 (*Continued*)

We consider the system of 20 items displayed in Table 21 in Appendix D. Using Algorithm 7 we generate solutions for the Multi-Item Problem. The results of the Multi-Item Approach for different system target fill rates $\text{MIA}_{FR^{obj}}$ are displayed in Table 7. One can immediately notice that the total costs are really low for the Multi-Item Approach compared to the Class and Item Approaches. This follows from zero to low stock levels for the classes with high prices A3, B3 and C3. So, the Multi-Item Approach does provide really low total costs while maintaining a high overall target. But, it is not realistic to set the base stock level of (almost) all expensive items to zero. A blend between the Approaches could be a solution to both overstocking and understocking of expensive items in a system.

Table 7: **(Example 2 - Cont'd)** Total stock per class S_{Class}, the total cost $C(\mathbf{S})$ and the aggregate fill rate $FR(\mathbf{S})$ obtained by Class Approach (CA) compared to the Item Approach ($\text{IA}_{FR^{obj}}$) with different fill rate targets FR^{obj}.

	S_{A1}	S_{A2}	S_{A3}	S_{B1}	S_{B2}	S_{B3}	S_{C1}	S_{C2}	S_{C3}	$C(\mathbf{S})$ (€)	$FR(\mathbf{S})$ (%)
$\text{MIA}_{75\%}$	1	2	0	1	1	0	44	0	0	113.92	75.6
$\text{MIA}_{90\%}$	4	2	0	4	1	0	62	0	0	126.25	90.2
$\text{MIA}_{99\%}$	6	8	0	8	8	1	79	7	3	1,912.08	99.0
CA	3	6	0	6	6	3	49	4	6	9,187.99	96.5
$\text{IA}_{75\%}$	2	4	0	3	4	2	41	4	6	8,394.38	80.5
$\text{IA}_{90\%}$	2	5	0	4	5	3	47	6	7	9,409.94	92.8
$\text{IA}_{99\%}$	3	7	0	6	7	4	55	8	11	13,863.78	99.2

4.4.2 Local Search

As mentioned before, the Greedy Algorithm does not guarantee optimality. Hence, we present a Local Search (LS) Algorithm to try to improve the solution of the Multi-Item Approach presented by Van Houtum and Kranenburg (2015).
This algorithm still does not guarantee optimality. Note that we only present a LS algorithm for the Multi-Item Approach with the expected back orders as a service level.

Contrary to Global Optimisation Algorithms, the Local Search Algorithm focuses on an initial value and then searches for improvements in the neighbourhood of this particular value. This process is repeated iteratively until no further improvement can be obtained. The LS Algorithm is much faster than a Global Optimisation Algorithm. However, it could yield a sub-optimal solution. The algorithm for the Local Search applied in our case is described in the following steps:

STEP 1 An initial solution $\mathbf{S} = (S_1, \ldots, S_{|I|})$ with corresponding aggregate back orders $\mathbb{E}BO(\mathbf{S})$ and total cost $C(\mathbf{S})$ is chosen. In our case, this solution follows from Greedy Algorithm 6, with $\mathbb{E}BO(\mathbf{S}) \leq \mathbb{E}BO^{obj}$. This follows from the definition of the Algorithm.

STEP 2 To minimise the costs the target has to be approached as near as possible, whilst the system target should not be exceeded. So, the neighbourhood of the solution is scanned for global optima, by making small local changes to the initial solution. We search for the smallest contribution to the aggregate back orders when the stock level of an item is decreased by one and compute the corresponding $\mathbb{E}BO(\mathbf{S})$ and $C(\mathbf{S})$.

STEP 3 This process is repeated until the termination criterium is met. The termination criterium is that each item has no further improvement that can be found within the LS Algorithm, whilst the system target should not be exceeded.

STEP 4 The output for the best-improved solution is given by the algorithm.

In Step 2 we assign all items i a value Φ_i, which is equal to the increase in back orders concerning its cost. This value is similar to the value of Γ_i in Greedy algorithm 6. But now we search for the smallest contribution, by decreasing the stock level of one item by one. Using Equation 7, the increase in back orders $\Delta\mathbb{E}BO_i(S_i)$ can be defined as

$$\Delta\mathbb{E}BO_i(S_i) = \mathbb{E}BO_i(S_i) - \mathbb{E}BO_i(S_i - 1) = -\sum_{x=S_i+1}^{\infty} \mathbb{P}[X_i = x] = -\left(1 - \sum_{x=0}^{S_i} \mathbb{P}[X_i = x]\right).$$

So, we define Φ_i by

$$\Phi_i := \frac{-\left(1 - \sum_{x=0}^{S_i} \mathbb{P}[X_i = x]\right)}{c_i}. \tag{16}$$

In Algorithm 1 we present a pseudocode for the LS algorithm. We continue with Example 1 to show the effect of the LS algorithm.

Algorithm 1 *(Local Search Algorithm)*

Input Initial solution $\mathbf{S} = (S_1, \ldots, S_{|I|})$, Target level $\mathbb{E}BO^{obj}$

Step 1: Compute $C(\mathbf{S})$ and $\mathbb{E}BO(\mathbf{S})$.

Step 2: For all $i \in I$ define Φ_i with new S_i;

$k :=\text{argmin}_{i\in I}\Phi_i$; $\mathbf{S} := \mathbf{S} - \mathbf{e}_k$.

Step 3: Determine $C(\mathbf{S})$ and $\mathbb{E}BO(\mathbf{S})$;

If 'system target is exceeded', then STOP, else go to Step 2.

Output $\mathbf{S} = (S_1, \ldots, S_{|I|})$, $\mathbb{E}BO(\mathbf{S})$ and $C(\mathbf{S})$

Example 1 (*Continued*)

The solutions of the Local Search algorithm compared to the Greedy algorithm for the Multi-Item Problem are compared in Table 8.

Table 8: **(Example 1 - Cont'd)** Solutions Multi-Item and LS Algorithm

	$\mathbb{E}BO^{obj} = 0.1$		$\mathbb{E}BO^{obj} = 0.05$	
SKU id	S_i	S_i with LS	S_i	S_i with LS
1	8	8	***9***	***8***
2	5	5	6	6
3	2	2	2	2
4	1	1	1	1

We see that we did not find a better solution for the Multi-Item Approach with a system target level of 0.1. However, we did find a better solution for target level 0.05. The total expected back orders for the improved solution are:

$$\mathbb{E}BO(\mathbf{S}) = \sum_{i=1}^{4} \mathbb{E}BO_i(S_i) = 0.024, \quad \text{with} \quad C(\mathbf{S}) = \sum_{i\in I} c_i S_i = €141,55.$$

4.5 Blends Between Approaches

As seen in the previous sections the Item, Class- and Multi-Item Approach are good methods to determine stock levels. But for example, the Multi-Item Approach does provide really low total costs while maintaining a high overall target. However, it is not realistic to set the base stock level of (almost) all expensive items to zero. The Class-Approach on the other hand does prevent setting the stock levels of expensive items to zero, however might be too optimistic. A blend between the Approaches could be a solution to both overstocking and understocking of expensive items in a system. The goal is to create a blend between these approaches to be able to lower the costs. In this section, we propose two methods to create a blend between the approaches. The first one is straightforward where we leave out items from several classes from the Multi-Item Approach. This method will be called the 'Basic Blend Approach'. The second method is more advanced, where we optimise the class targets under the system target. We will refer to this method as the 'Advanced Blend Approach'. So far we have not seen any results of such approaches in literature.

4.5.1 Basic Blend Approach

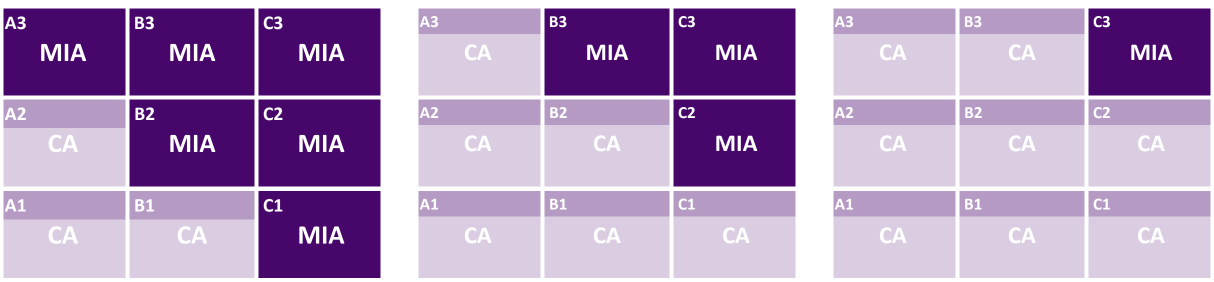
The Basic Blend Approach (BBA) is a combination of the Class Approach and the Multi-Item Approach, whereby some of the classes are left out of the Multi-Item Approach. We apply the Class Approach to the classes that are being left out. The reason for leaving out classes or items from the Multi-Item Approach is that this approach tends to overcompensate by increasing the number of cheap and fast-moving items. The BBA is a request of Gordian Logistic Experts B.V. (Gordian) to compare the stock levels of especially classes with expensive slow-movers from the classification matrix in Figure 1. They presume that with the Class Approach, the base stock levels are too high and can be improved with such an approach. With the BBA we want to analyse how the stock levels of especially class B3, C2 and C3 differ when the Multi-Item Approach is only applied on a subset of the classes.

For the Basic Blend Approach, we apply the following steps:

STEP 1 A system target service level is set and the items are divided into classes defined by the classification system of Gordian in Figure 1. For the sake of applicability and interpretability, we use the fill rate (FR) as a service level for this approach.

STEP 2 The system of items is divided into 2 sets of items. In Figure 2 we see three different cases of leaving out items from the Multi-Item Approach (MIA). These choices are made in cooperation with Gordian based on the criticality of the classes.

STEP 3 We apply the Multi-Item Approach with fill rate service level for the items in the classes with a dark purple colour. And the Class Approach (CA) for the items with a light purple colour.



(a) BBA - I (b) BBA - II (c) BBA - III

Figure 2: Three cases of the Basic Blend Approach, where MIA stands for the Multi-Item Approach and CA stands for the Class Approach.

We demonstrate the results of the Basic Blend Approach in the following Example.

Example 2 (*Continued*)

We consider the system of 20 items displayed in Table 21 in Appendix D.1. For the Multi-Item Approach, we set the target fill rate $FR^{obj} = 0.9$.

Applying the Class Approach and the Multi-Item Approach according to the three cases in Figure 2 we obtain the results displayed in Table 9. We compare the total costs $C(\mathbf{S})$, the total realised $FR(\mathbf{S})$ and the total stock levels per class S_{Class} of the Basic Blend Approaches (BBA-I, BBA-II and BBA-III) with the Multi-Item Approach (MIA), the Class Approach (CA) and the Item Approach (IA).

Table 9: **(Example 2 - Cont'd)** Total stock per class S_{Class}, the total cost $C(\mathbf{S})$ and the aggregate fill rate $FR(\mathbf{S})$ obtained by BBA-I,II and III compared to the Class Approach (CA)

	S_{A1}	S_{A2}	S_{A3}	S_{B1}	S_{B2}	S_{B3}	S_{C1}	S_{C2}	S_{C3}	$C(\mathbf{S})$ (€)	$FR(\mathbf{S})$ (%)
BBA-$\text{I}_{75\%}$	3	6	0	6	1	0	41	0	0	321.72	80.8
BBA-$\text{II}_{75\%}$	3	6	0	6	6	2	49	7	4	2,691.91	96.2
BBA-$\text{III}_{75\%}$	3	6	0	6	6	3	49	4	6	6.437.83	96.5
BBA-$\text{I}_{90\%}$	3	6	0	6	1	0	55	0	0	324.76	93.0
BBA-$\text{II}_{90\%}$	3	6	0	6	6	3	49	7	6	6,492.98	96.6
BBA-$\text{III}_{90\%}$	3	6	0	6	6	3	49	4	8	9,682.00	96.6
CA	3	6	0	6	6	3	49	4	6	9,187.99	96.5

Notice that, $BBA - III_{75\%}$ and the Class Approach have exactly the stock levels per class, but the total costs of the Class Approach are a lot higher. This is because items with SKU ids 9 and 10 have base stock levels 1 and 2 for the Class Approach and base stock levels 2 and 1 for the Basic Blend Approach, respectively. Both items 9 and 10 are classified in class C3, so the total stock level of class C3 remains the same. However, the purchase price of item 10 is a lot higher than the purchase price of item 9. And thus, the total cost of the $BBA - III_{75\%}$ is a lot lower than the total cost of the Class Approach. Furthermore, both methods achieve the same aggregate fill rate for the total system. The difference of especially these two methods is that in the Class Approach, each item in class C3 has an item target fill rate of 75% and in the $BBA - III_{75\%}$ all items in class C3 have a total target of 75%.

4.5.2 Advanced Blend Approach

In this section, we propose an approach to find optimal class fill rate (FR) targets $FR_{K_j}^{obj}$, such that when the Class Approach is applied using the optimised targets, the system target FR is achieved. We refer to this approach as the Advanced Blend Approach (ABA). Notice that we use the fill rate as a service level for the ABA, but the same can be applied when using the mean aggregate back orders as a service level. For this approach we use classes $A_1, A_2, A_3, \ldots, C_1, C_2, C_3$ as defined by Gordian Logistic Experts B.V. (Gordian) explained in Section 4.3. For mathematical formulation we define classes K_j with $j \in J = \{1, \ldots, 9\}$.

Notice that the ABA is another optimisation problem, which can be defined as follows. The objective of this new optimisation problem is to minimise the target service levels of classes K_j. The realised FR of the system after applying the Class Approach with targets FR_{K_j} is defined as $FR(\mathbf{S}) = \sum_{j \in J} \sum_{i \in K_j} FR_i(S_i)$. So the constraint of the problem is that the aggregate system fill rate $FR(\mathbf{S})$ should be at least as great as the system target fill rate FR^{obj}. See Problem 17 for a mathematical formulation of the optimisation problem corresponding to the ABA.

$$
\begin{aligned}
\min \quad & C(\mathbf{S}) \\
\text{s.t.} \quad & S_i = \min\{k \in \mathbb{N} : FR_i(k) \geq FR_{K_j}^{obj}\}, \forall i \in K_j, \\
& FR(\mathbf{S}) \geq FR^{obj}, \\
& 0 \leq FR_{K_j}^{obj} \leq 1, \\
& \mathbf{S} \in \mathcal{S}',
\end{aligned}
\tag{17}
$$

with $\mathcal{S}'$ defined as $\{\mathbf{S} = (S_1, \ldots, S_{|I|}) \mid S_i \geq \max\{\lceil \mu_i L_i - 1 \rceil, 0\}, \forall i \in I\}$. This is again a nonlinear integer programming problem. To be able to solve this problem we introduce a new Greedy based approximation algorithm, explained in the next section.

Greedy Algorithm for ABA

Recall the definition of the aggregate system fill rate and the item fill rates:

$$FR(\mathbf{S}) \quad = \quad \sum_{i \in I} \tfrac{\mu_i}{M} \cdot FR_i(S_i) \quad \text{with} \quad FR_i(S_i) \quad = \quad \sum_{x=0}^{S_i - 1} \mathbb{P}[X_i = x].$$

From Lemma 4.2 it follows that $FR_i(S_i)$ is a concave and increasing function. Hence, a greedy algorithm can be applied to generate efficient solutions to Problem 17.

As an input, we need the average demand μ_i per year, the frequency of the demand f_i, the purchase prices c_i in €, the lead time L_i per year and the system fill rate target FR^{obj}. We start with an initial solution $\mathbf{S} = (S_1, \ldots, S_{|I|})$ with $S_i = \max\{\lceil \mu_i L_i - 1 \rceil, 0, \forall i \in I\}$. This solution generates the lowest possible investment. As initial FR targets for the classes, we set $FR_{K_j}^{obj} := \sum_{x=0}^{S_i - 1} \mathbb{P}[X_i = x]$ generated with our initial solution $\mathbf{S}$. These class FR targets are a lower bound for the problem. Then in each iteration, we choose one class to increase its target by 1%. The choice of this class is based on the highest added value per class. This principle is also known as the "*Biggest Bang For The Buck*" and is also used in the Multi-Item Approach in Section 4.4. We denote this added value by Θ_j and we define Θ_j as follows. The subscript j stands for the j^{th} class K_j.

Define ΔS_i as the increase in the stock level of SKU $i \in K_j$ when the class target fill rate $FR_{K_j}^{obj}$ is increased by 1%. Then we define Θ_j as

$$\Theta_j = \frac{\sum_{i \in K_j} \Delta S_i \cdot \mu_i}{M \cdot \sum_{i \in K_j} \cdot c_i}, \tag{18}$$

with $M = \sum_{i \in I} \mu_i$.

Class K_j with the highest corresponding value for Θ_j is selected and its fill rate target $FR_{K_j}^{obj}$ is increased by 1%. With our new class FR target we generate a new solution $\mathbf{S}$ and its corresponding realised fill rate $FR(\mathbf{S})$. We continue this process until the total FR for the system $FR(\mathbf{S})$ is greater than or equal to the system FR target FR^{obj}. One issue is that because of the integrality of S_i, a small increase in the target FR can result in high stock levels and thus high costs. Therefore, after each iteration, the class FR target is set to the lowest realised item fill rate in that class. In Algorithm 2 a pseudo code for the ABA is given.

Algorithm 2 *(Algorithm for Advanced Blend Approach)*

Input Average demand μ_i, Cost c_i, Lead time L_i, System target service level FR^{obj}

Step 1: Classification Classify all $i \in I$ in classes K_j, with $j = 1, \ldots, 9$ and $I = \bigcup_{j=1}^{9} K_j$

Step 2: Initialisation For all $j = 1, \ldots, 9$ and for all $i \in K_j$ set $S_i := \max\{\lceil \mu_i L_i - 1 \rceil, 0\}$;

Compute $C(\mathbf{S})$, $FR(\mathbf{S})$ and set for all K_j: $FR_{K_j}^{obj} := \sum_{x=0}^{S_i - 1} \mathbb{P}[X_i = x]$ for all $i \in K_j$.

Step 3: For all K_j define Θ_j with new S_i;

$k :=$argmax$_j \Theta_j$; $l :=$ argmax$_i FR_i(S_i)$ with $i \in K_k$; $FR_{K_k}^{obj} := FR_l(S_l)$.

Step 3: Determine $C(\mathbf{S})$ and $FR(\mathbf{S})$ by applying Class Approach with new targets;

If 'system target is reached', then STOP, else go to Step 3.

Output $\{FR_{K_1}^{obj}, \ldots, FR_{K_9}^{obj}\}$, $\mathbf{S} = (S_1, \ldots, S_{|I|})$, $FR(\mathbf{S})$ and $C(\mathbf{S})$

We demonstrate the Advanced Blend Approach in the following Example.

Example 2 (*continued*)

We consider the system of 20 items displayed in Table 21 in Appendix D.1. In Table 22 in Appendix D.1 we present the optimised class target fill rates for different system target fill rates. One can immediately notice that the targets for classes A3, B3, C2 and C3 are all 0% for system fill rate targets 80% until 96%. Notice that in our example no items are classified to class A3, hence the target fill rates of class A3 are all 0.

However, the other classes are nonempty but still have an optimal class target of 0%. Finally, in Table 23 in Appendix D.1 we compare the results of the $ABA_{FR^{obj}}$ to the Multi-Item Approach ($MIA_{FR^{obj}}$) and Class Approach for three different target fill rates FR^{obj}.

In the next chapter we continue with a data analysis of the data set on which we will apply all approaches described in this chapter.

5 Data Analysis

All data sets used for this research are provided by clients of Gordian Logistic Experts B.V. (Gordian). Due to confidentiality agreements, the company names are non-disclosed. Furthermore, the prices are scaled such that the privacy is not intruded.

In this section, one large data set is analysed. The data set is provided by a low-cost airline and contains a lot of data that is not needed for this research.

The general information that is relevant and needed is:

- the number of parts,
- the total period of the collected data,
- the number of orders during the total period, so the order frequency,
- the order and supply moments, to determine the lead time,
- the order quantities,
- the price of a part, and
- the location of a part (in a system)

To ensure completeness, we make the following assumptions considering the data:

Assumption 10. *Only unplanned demand is considered.*

Assumption 11. *Only items with a yearly demand greater than* 0 *are considered.*

Assumption 12. *Only items located at the main location are considered.*

Assumption 13. *When the supply time of an item is not reliable, the predicted lead time will be used for calculations.*

Assumption 14. *An item is considered to be part of just one system.*

Assumption 15. *One month has 30 days and one year has 365 days.*

Assumption 16. *All demand is considered to be of equal importance, whether it is from a different location or from the main location itself.*

The data set includes (statistical) information about the spare parts of aeroplanes. The parts range from a little lamp used above a seat to a large engine for example. For an overview of the numbers of this data set, see Table 10. There is a total of 60 months of historical data available.

The first month of available data is September 2017 and the last month of data is August 2022. The average demand lead time is calculated over this time period. For programming purposes, the items are sorted in alphabetical order and they are assigned a number i.

Table 10: Statistics

Airline Data	
Nr. of spare parts	39,003
Nr. of parts at main location	33,047
Nr. of systems/aeroplanes	39
Nr. of aeroplane types	2
Price cheapest part	€0.0002
Price most expensive part	€1,121,210

The company owns 39 aeroplanes and has a total of $39,003$ different spare parts. However, there are just two types of aeroplanes. Furthermore, the airline has 4 different locations to stock their spare parts. We only consider the main location since the remaining 3 locations have the main location as their preferred supplier. Moreover, the main location has the largest number of parts. Specifically, $33,047$ parts are located at the main location.

The company did not provide any information about what kind of systems there are in an aeroplane. Moreover, no information about which part belongs to which system is present. So for this research, every aeroplane will be considered one system. More specifically, we select the data of one aeroplane to use for the numerical results. We consider a subset of the large data set where the average demand is nonzero and unplanned at the main location. After applying Assumptions 10, 11 and 12 to the selected aeroplane, there are 4701 items in our data set.

In Figure 7 in Appendix C a scatter plot of relevant items of our aeroplane is given. The items are classified according to the classification matrix of Gordian in Figure 1. The different classes are represented by different colours. Notice that, this scatter plot is resized to display the items more clearly.

The cheapest and most expensive items in the aeroplane are €0,0002 and €695,844, respectively. Furthermore, the items with the lowest and highest yearly demand have an average demand of 0.2 and $23,339.62$ per year, respectively. Notice that the prices and the average demand have a large gap between the highest and lowest values, so the Multi-Item Approach is likely to be unbalanced as well. In Table 11 the number of parts and the prices of the cheapest and most expensive parts of each class are displayed. Notice that we have a very unbalanced data set in terms of the number of items in each class.

Table 11: Data aeroplane

	$A1$	$A2$	$A3$	$B1$	$B2$	$B3$	$C1$	$C2$	$C3$
Nr. of parts	130	54	31	275	114	92	2084	1062	859
Price cheapest part (€)	0.01	34.08	550.29	0.01	33.09	504.33	2e-04	33.06	501.52
Price most expensive part (€)	31.95	406.25	50,775.92	30	480.66	121,612	32.83	500.97	695,844

6 Numerical Results

This section contains all numerical results and analyses of the different approaches explained in Chapter 4 using the data set which is analysed in Chapter 5. The results are all generated using the Java programming Language and the figures are all created in RStudio. First, we present the results for each method separately and we finish with a comparison of all the methods.

6.1 Item Approach

In this section, we present the numerical results of the Item Approach explained in Section 4.2.

6.1.1 Expected backorders as Service Level

Following Algorithm 3 we generate solutions for the Item Approach with expected backorders as service level. Let $\mathbb{E}BO^{obj}$ be the system target expected backorders. In Table 12 the total costs and the realised aggregate mean backorders are given for $\mathbb{E}BO^{obj} \in [0, \dots, 2.5]$. We see that the realised aggregate backorders have a large gap to their corresponding target.

Table 12: The total costs $C(\mathbf{S})$ and aggregate mean backorders $\mathbb{E}BO(\mathbf{S})$ for different target levels $\mathbb{E}BO^{obj}$ generated following the Item Approach.

$\mathbb{E}BO^{obj}$	0.0	0.25	0.5	0.75	1.0	1.5	1.75	2.0	2.25	2.5
$C(\mathbf{S})$ (€ x Millions)	22.00	10.07	9.83	9.66	9.40	9.32	9.06	9.03	8.97	8.92
$\mathbb{E}BO(\mathbf{S})$	0.0	0.192	0.381	0.587	0.778	1.168	1.361	1.574	1.782	1.972

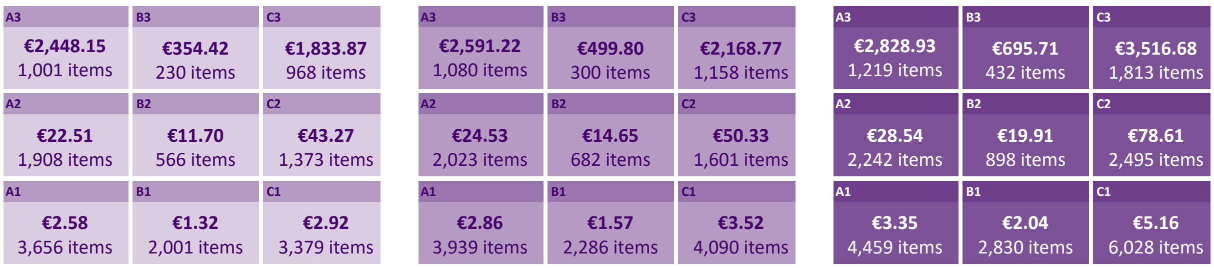
In Figure 9 in D.2 the total costs $C(\mathbf{S})$ are depicted for different values of $\mathbb{E}BO^{obj}$. We see that the total costs are lower for higher targets, which is an obvious result.

6.1.2 Fill Rate as Service Level

Following Algorithm 4 we generate solutions for the Item Approach with the fill rate as service level. Let FR^{obj} be the system target fill rate. In Table 13 the total cost and the aggregate fill rate are presented for different levels of FR^{obj}. See Figure 8 in the Appendix D.2 for a plot of the total costs plotted against fill rate targets. In Figure 3 we displayed the total stock and total costs per class for three different target levels to showcase how the costs are divided over the classes.

Table 13: The total costs $C(\mathbf{S})$ and total realised fill rate $FR(\mathbf{S})$ for different target levels FR^{obj} generated following the Item Approach.

FR^{obj}	60%	65%	70%	75%	80%	85%	90%	95%	98%	99%
$C(\mathbf{S})$ **(€ x Millions)**	4.37	4.44	4.53	4.72	4.90	5.10	5.36	5.93	6.86	7.18
$FR(\mathbf{S})$ **(%)**	65.0	69.4	73.8	78.4	82.8	87.3	91.5	95.9	98.4	99.2



(a) $FR^{obj} = 75\%$ (b) $FR^{obj} = 90\%$ (c) $FR^{obj} = 99\%$

Figure 3: The total costs $C(\mathbf{S})$ (in € ($\times$1000)) and stock per class for target fill rates of 75%, 90% and 99%, respectively.

The computation time for the Item Approach is 0 seconds for our data set with $|I| = 4701$.

6.2 Class Approach

For the Class Approach, we use the classification matrix of Gordian as explained in Section 4.3. As a service level, we use the fill rate. In Figure 4 the results of the Class Approach using Algorithm 5 in Appendix A.2 are displayed. Notice that in the Class Approach, we use the fixed class targets from Figure 1.

The aggregate fill rate for the system with corresponding total costs is equal to

$$FR(\mathbf{S}) = 97.8\% \quad \text{with} \quad C(\mathbf{S}) = \textstyle\sum_{i=1}^{4701} c_i S_i = €5,095,928.98.$$

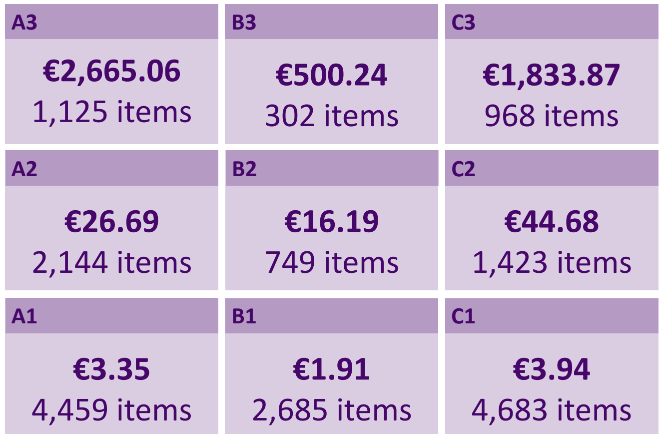


Figure 4: The total costs $C(\mathbf{S}$ in (x1000) and stock per class generated using the Class Approach

From Table 13 it follows that for an aggregate fill rate of 98 % the total costs are approximately €6.86 Million, which is considerably higher than the total cost following the Class Approach. Lastly, the computation time for the Class Approach is 0 seconds for our data set with $|I| = 4701$.

6.3 Multi-Item Approach

In this section, we present the numerical results of the Multi-Item Approach explained in Section 4.4.

6.3.1 Expected Backorders as Service Level with Local Search

Following Algorithm 6 we applied the Multi-Item Approach (MIA) with target expected backorders $\mathbb{E}BO^{obj} = 2.0$. In Figure 5 we depicted a subset of the solutions generated by the Greedy Algorithm which did not satisfy the target.

In Section 4.4.2 we introduced a Local Search (LS) Algorithm to attempt to improve the solutions generated by Algorithm 6. We apply this algorithm with a target expected back orders $\mathbb{E}BO^{obj} = 2.0$.

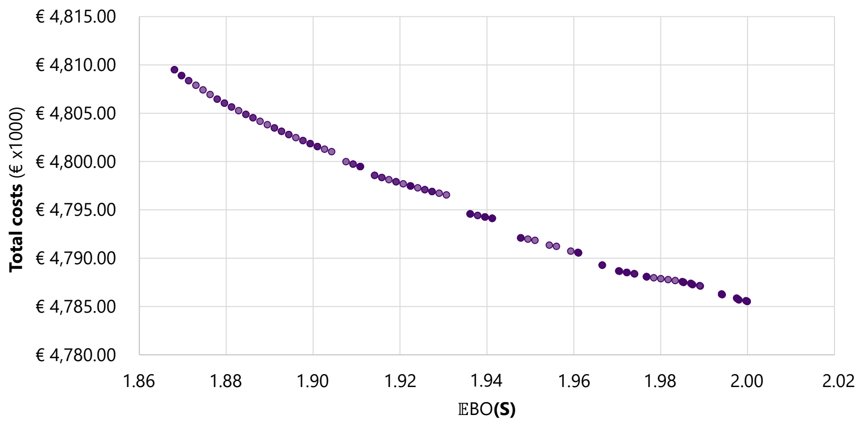


Figure 5: Total costs ($C(\mathbf{S})$ (€ x1000)) for different expected back orders ($\mathbb{E}BO(\mathbf{S})$) obtained by the Multi-Item Approach

In Table 15 we show the total stock levels per class, the total costs, the aggregate expected backorders and the computation time of both approaches. The total cost generated following from the MIA is approximately €4.81 Million. The LS algorithm lowers the total costs to approximately €4.79 Million, which is a decrease of approximately €$200,000.00$. Notice that the LS algorithm found a feasible solution with corresponding aggregate expected backorders very close to the target. Namely, $\mathbb{E}BO(\mathbf{S}) = 1.999$, whilst the aggregate expected backorders following from the Multi-Item Approach is $\mathbb{E}BO(\mathbf{S}) = 1.868$. We see that only the stock levels of classes C1, C2 and C3 have a noticeable decrease after applying the LS algorithm.

Table 14: Total stock per class S_{Class}, the total cost $C(\mathbf{S})$ and the expected back orders $\mathbb{E}BO(\mathbf{S})$ obtained by Multi-Item Approach compared to the Multi-Item Approach with LS for a target service level $\mathbb{E}BO^{obj} = 2.0$.

	S_{A1}	S_{A2}	S_{A3}	S_{B1}	S_{B2}	S_{B3}	S_{C1}	S_{C2}	S_{C3}	$C(\mathbf{S})$ (€)	$\mathbb{E}BO(\mathbf{S})$	Computation Time (s)
MIA	5,797	2,558	1,265	4,296	1,171	429	10,307	2,862	1,428	4,809,499.30	1.868	4,663
MIA - LS	5,797	2,558	1,265	4,296	1,171	429	10,227	2,827	1,373	4,785,516.55	1.999	27

Lastly, the computation time of the MIA with $|I| = 4701$ is 78 minutes, which is noticeably high compared to the Item Approach and Class Approach. The LS algorithm takes only 27 seconds since it starts with an initial solution close to the target.

6.3.2 Fill Rate as Service Level

Let FR^{obj} be the system target fill rate. We denote the Multi-Item Approach with system target FR^{obj} as $MIA_{FR^{obj}}$. Following Algorithm 4 we generated solutions for three different targets, namely 75%, 90% and 99%. These solutions are displayed in Table 15. To showcase how the total stock is divided over the classes, we also displayed the total stock per class denoted by S_{Class}. We see that the total costs vary approximately €3,300.00 between targets 75% and 90%, whilst the costs vary approximately €360,000.00 between targets 90% and 99%. In Figure 6 we plotted different target fill rates between 40% and 100 % against the total costs. Hence, to achieve higher service levels the algorithm has a considerably high computation time. To achieve a system target of 99%, the algorithm runs for 41 minutes. Furthermore, the total costs in case BBA-III are higher compared to the other cases. To demonstrate these differences in stock levels and costs, see Figures 10 - 12.

Table 15: Total stock per class S_{Class}, the total cost $C(\mathbf{S})$ and the realised fill rate $FR(\mathbf{S})$ obtained with the Multi-Item Approach at different target fill rates FR^{obj}.

	S_{A1}	S_{A2}	S_{A3}	S_{B1}	S_{B2}	S_{B3}	S_{C1}	S_{C2}	S_{C3}	$C(\mathbf{S})$ (€)	$FR(\mathbf{S})$	Computation Time (s)
MIA$_{75\%}$	3,979	1,734	879	2,175	317	52	2,175	142	10	2,326,490.76	75.0	454
MIA$_{90\%}$	4,693	1,851	879	2,873	353	52	4,204	153	10	2,329,785.36	90.0	999
MIA$_{99\%}$	5,616	2,437	1,087	3,955	969	172	8,504	1,785	218	2,690,354.52	99.0	2,477

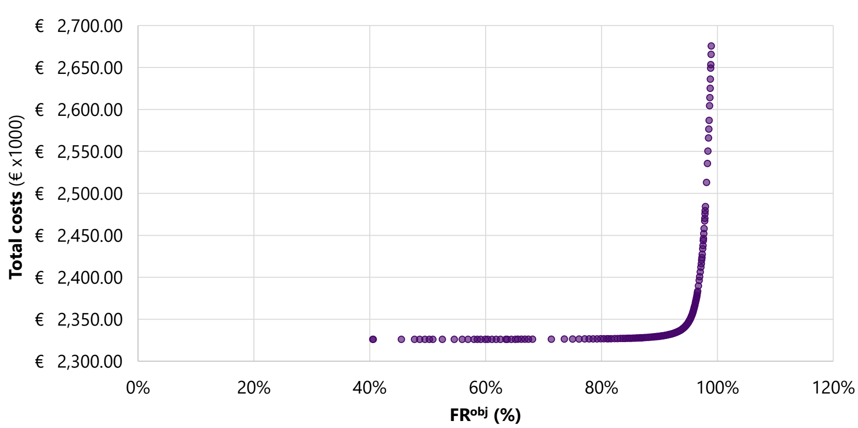


Figure 6: Total costs $C(\mathbf{S})$ (€ x1000) for different Fill Rate targets FR^{obj} (%)

6.4 Basic Blend Approach

In this section, we present the results of the Basic Blend Approach (BBA) as explained in Section 4.5.1. In the BBA we consider three cases BBA-I, BBA-II and BBA-III. For all three cases, we generate solutions for three different target fill rates for the MIA, namely 75%, 90% and 99%. These solutions are presented in Table 16. Notice that for the target fill rate 99%, the total realised fill rates of all three cases are lower than 99%. This is because the system target is only set for a subset of the classes. The remaining classes have different class targets corresponding to the Class Approach. Only class A1 has a class target of 99%, the other classes have lower targets. Hence, the overall fill rate can be lower than 99%. The computation time of the BBA is not as low as that of the Class Approach, which has a computation time of 0 seconds. Especially for BBA-I, we have high computation times, since we apply the Multi-Item Approach on a large subset of all the items. The Multi-Item Approach has computation times as seen in the previous section.

Table 16: An overall comparison for the Basic Blend Approaches for different target fill rates (FR^{obj}), the realised target fill rates ($FR(\mathbf{S})$, realised total costs ($C(\mathbf{S})$) and computation time.

Methods	FR^{obj} **(%)**	$FR(\mathbf{S})$ **(%)**	$C(\mathbf{S})$ **(€)**	**Computation Time (s)**
BBA-I	75	86.57	€ 2,341,777.74	693
BBA-II	75	97.28	€ 2,839,748.99	101
BBA-III	75	97.68	€ 3,475,047.20	19
Methods	FR^{obj} **(%)**	$FR(\mathbf{S})$ **(%)**	$C(\mathbf{S})$ **(€)**	**Computation Time (s)**
BBA-I	90	93.52	€ 2,457,500.37	1,220
BBA-II	90	97.77	€ 3,042,881.86	179
BBA-III	90	97.78	€ 3,810,178.95	30
Methods	FR^{obj} **(%)**	$FR(\mathbf{S})$ **(%)**	$C(\mathbf{S})$ **(€)**	**Computation Time (s)**
BBA-I	99	98.56	€ 3,355,322.10	1,849
BBA-II	99	98.07	€ 4,223,135.21	320
BBA-III	99	97.84	€ 5,027,538.01	53

Notice that the light-coloured classes in Figures 10 - 12 all have the same stock levels and costs as the Class Approach as shown in Appendix D.3. The Basic Blend Approach focuses on the differences in the total costs of the dark-coloured classes compared to the Class Approach. The stock levels of class C3 are considerably higher for BBA-III compared to BBA-I and BBA-II. The most expensive items are in class C3, so for higher stock levels it has the highest costs. Compared to the Class Approach classes B3 and C3 have lower total costs for all BBA cases. In the Class Approach class C3 has a target of 75%. In BBA-III the overall system target for the whole class is 75%. The total costs of the class decrease with €68,000.00 whilst the total stock level increases with 1691. This is because the items in class C3 still have large price gaps. Table 11 follows that the gap between the price of the most expensive and the cheapest part in C3 is €695.342,48.

6.5 Advanced Blend Approach

In this section, we present the results of the Advanced Blend Approach introduced in Section 4.5.2. Using Algorithm 2, we generate the optimised class target fill rates for three different system targets, namely 75%, 90% and 99%. See Table 17 for these optimised class targets. In Table 24 in Appendix D.4, we present the optimised class targets for all integer system targets between 75% and 99%. Notice that the ABA generates near zero optimised class targets, except for system target 99%.

From the Class Approach applied to this particular data set, it follows that the aggregate fill rate is 97.8%. If we look at Table 24, the optimised class targets corresponding to system target 97% are slightly different from the class targets in the Class Approach. Specifically, the optimised targets for classes $A3$, $B3$, and $C3$ are a lot lower, whilst the remaining classes have a slightly higher optimised class target.

Table 17: Optimal Class Targets for the different Target Fill Rates (FR^{obj}) obtained by the Advanced Blend Approach

FR^{obj} (%)	**A1** (%)	**A2** (%)	**A3** (%)	**B1** (%)	**B2** (%)	**B3** (%)	**C1** (%)	**C2** (%)	**C3** (%)
75	99.5	49.0	45.0	99.7	32.9	12.1	56.8	16.8	0.9
90	99.5	90.5	45.0	99.7	32.9	12.1	99.3	16.8	0.9
99	99.5	99.7	99.1	99.7	99.2	50.6	99.3	99.7	70.3

When we look at Figures 13 and 14 in Appendix D.4 we see that the costs of classes $A3$, $B3$, $C3$ are higher than the costs of the other classes, while their stock levels are really low. The reason for that is our data set is very unbalanced. For example, class C3 has many items with very low demand concerning their costs. So the algorithm is not likely to select class C3 to increase the class target fill rate.

In Table 18 the aggregate fill rate with corresponding total cost obtained by the ABA is presented for target fill rates 75%, 90% and 99%. The table also contains the corresponding computation times. Notice that the ABA is a noticeably fast approach since the computation times are all under 1 minute for our data set with $|I| = 4701$.

Table 18: Different Target Fill Rates (FR^{obj}) with the realised target fill rates, realised total costs and computation time obtained by the Advanced Blend Approach

FR^{obj} **(%)**	$FR(\mathbf{S})$ **(%)**	$C(\mathbf{S})$ **(€)**	**Computation Time (s)**
75	75.02	€ 2,332,341.65	12
90	90.02	€ 2,339,564.55	16
99	99.00	€ 4,938,916.84	24

6.6 Comparison

In this section, we compare the results presented in sections 6.1-6.5. In Table 19 the total costs, the aggregate fill rates and the corresponding computation times of all methods are displayed and compared to the results of the Class Approach. The table follows that the Item Approach performs the worst. It obtains the highest total investment costs concerning the aggregate fill rates. However, the computation time of the Item Approach is together with the Class Approach, the lowest compared to the other approaches.

The Multi-Item Approach results in the lowest total investment costs concerning the aggregate fill rates. However, the computation time of this approach together with the Basic Blend Approach - Case I are the highest compared to the other approaches.

Table 19: The total cost $C(\mathbf{S})$ and the aggregate fill rate $FR(\mathbf{S})$ obtained by comparison in between all approaches.

Approach	$C(\mathbf{S})$ (€)	$FR(\mathbf{S})$ (%)	Computation Time (s)
CA	**5,095,928.98**	**97.8**	**0**
$\text{IA}_{75\%}$	4,720,733.82	78.4	0
$\text{MIA}_{75\%}$	2,326,490.76	75.0	454
$\text{BBA-I}_{75\%}$	2,341,777,74	86.6	693
$\text{BBA-II}_{75\%}$	2,839,748.99	97.3	101
$\text{BBA-III}_{75\%}$	3,475,047.20	97.7	19
$\text{ABA}_{75\%}$	2,332,341.65	75.0	12
$\text{IA}_{90\%}$	5,357,231.65	91.5	0
$\text{MIA}_{90\%}$	2,329,785.36	90.0	999
$\text{BBA-I}_{90\%}$	2,457,500.37	93.5	1,220
$\text{BBA-II}_{90\%}$	3,042,881.86	97.8	179
$\text{BBA-III}_{90\%}$	3,810,178.95	97.8	30
$\text{ABA}_{90\%}$	2,339,564.55	90.0	16
$\text{IA}_{99\%}$	7,178,931.49	99.2	0
$\text{MIA}_{99\%}$	2,690,354.52	99.0	2,477
$\text{BBA-I}_{99\%}$	3,335,322.10	98.6	1,849
$\text{BBA-II}_{99\%}$	4,223,135.21	98.1	320
$\text{BBA-III}_{99\%}$	5,027,538.01	97.8	53
$\text{ABA}_{99\%}$	4,938,916.84	99.0	24

In Table 20 the stock levels per class are compared for all approaches. The Multi-Item Approach has the lowest total investment costs, as the determined stock levels of the classes with expensive items are the lowest for the Multi-Item Approach. On the other hand, the Multi-Item Approach has the highest stock levels for class A1 with system target fill rates of 90%.
Comparing the Basic Blend Approach and the Advanced Blend Approach to the Class Approach, it follows that both blends obtain lower total investment costs for the three specific target fill rates. A trade-off between the total investment costs and the total stock levels per class can be of help by selecting an approach to determine minimum base stock levels while satisfying high service measures.

Table 20: Total stock per class S_{Class} by comparison in between all approaches.

Approach	S_{A1}	S_{A2}	S_{A3}	S_{B1}	S_{B2}	S_{B3}	S_{C1}	S_{C2}	S_{C3}
CA	**4,459**	**2,144**	**1,125**	**2,685**	**749**	**302**	**4,683**	**1,423**	**968**
$\text{IA}_{75\%}$	3,656	1,908	1,001	2,001	566	230	3,379	1,373	968
$\text{MIA}_{75\%}$	3,979	1,734	879	2,175	317	52	2,175	142	10
$\text{BBA-I}_{75\%}$	4,459	2,144	879	2,685	520	52	5,580	251	10
$\text{BBA-II}_{75\%}$	4,459	2,144	999	2,685	869	118	7,781	1,335	63
$\text{BBA-III}_{75\%}$	4,459	2,144	1,233	2,685	1,119	330	9,813	2,438	818
$\text{ABA}_{75\%}$	4,586	1,776	879	3,083	317	52	3,026	142	10
$\text{IA}_{90\%}$	3,939	2,023	1,080	2,286	682	300	4,090	1,601	1,158
$\text{MIA}_{90\%}$	4,693	1,851	879	2,873	353	52	4,204	153	10
$\text{BBA-I}_{90\%}$	4,459	2,144	1,125	2,685	749	134	4,683	1,495	106
$\text{BBA-II}_{90\%}$	4,459	2,144	1,125	2,685	749	265	4,683	2,131	514
$\text{BBA-III}_{90\%}$	4,459	2,144	1,125	2,685	749	437	4,683	3,041	1,358
$\text{ABA}_{90\%}$	4,586	2,028	879	3,083	317	52	6,292	142	10
$\text{IA}_{99\%}$	4,459	2,242	1,219	2,830	898	432	6,028	2,495	1,813
$\text{MIA}_{99\%}$	5,616	2,437	1,087	3,955	969	172	8,504	1,785	218
$\text{BBA-I}_{99\%}$	4,459	2,144	1,125	2,685	749	302	4,683	1,423	689
$\text{BBA-II}_{99\%}$	4,459	2,144	1,125	2,685	749	302	4,683	1,423	1,078
$\text{BBA-III}_{99\%}$	4,459	2,144	1,125	2,685	749	302	4,683	1,423	1,789
$\text{ABA}_{99\%}$	4,586	2,319	1,223	3,083	911	178	6,292	2,901	944

7 Conclusion

This section is dedicated to the conclusion of this research and the answers to our research questions. Recall the main research question:

How can we develop a blend between the single-item approach and
a system approach in spare parts management?

To answer this main question, the definition of item and system approaches in spare parts management have been assessed. The summary of the methods that are assessed with the main findings is as follows:

- *Item Approach* - In an item approach the target service levels are set individually. In this research, we refer to the Item Approach as the approach which sets the same target level for each item. As a result, the stock levels obtained are too high, which results in high costs. The Item Approach is a fast approach, easy to implement and flexible.
- *Class Approach* - The approach we refer to as the Class Approach is also an item approach where equal item target levels are set for all items in a class. However, within this approach, individual targets are set based on the classification of items. It is a solid method often used in practice because it makes sense to give items different target service levels based on their price and demand. Moreover, it is flexible and fast. An important part of this approach is the choice of class target service levels, which is done manually and arbitrary.
- *Multi-Item Approach* - A system approach is an approach where base stock levels for items are set based on a system target service level and no individual item targets are used. We refer to the system approach where we use a Greedy algorithm to solve the multi-item problem as the Multi-Item Approach. In this research, it was found that the Multi-Item Approach results in low total costs and high computation times. The main disadvantage of this approach is that it is not easy to interpret and it results in unbalanced stock levels, especially for expensive slow-movers the stock levels are low, whilst the opposite holds for cheap fast-movers.

We find that the fill rate as a target service level in these approaches is more practical since it is also used by Gordian and a lot of other companies. However, the number of expected back orders can be more useful in a theoretical framework. Hence, we describe the Item and Multi-Item Approach for both service levels. Nevertheless, for comparison reasons, we implemented the following two blends only with the fill rate as a service level.

Within this research, two blend approaches are proposed and compared. The first one is referred to as the Basic Blend Approach and the second one is the Advanced Blend Approach. The two methods are summarised with their respective findings as follows:

- *Basic Blend Approach* - This approach selects a subset of classes defined by Gordian and sets the system fill rate target only for these classes. The remaining classes in the system use the Class Approach to determine base stock levels. With the Basic Blend Approach the problem of unbalanced results as obtained with the Multi-Item Approach for the total system, is solved. However, the method is not easy to interpret, because part of the approach still uses a Multi-Item Approach. Furthermore, it has relatively high computation times. Another relevant point to consider is the choice of the subset of classes or items. All three considered cases result in lower costs. Only the third case has a relevant decrease in stock levels in the classes with expensive slow-movers.
- *Advanced Blend Approach* - In the Advanced Blend Approach we developed an algorithm to determine optimal class fill rate targets, whilst satisfying a system fill rate target. This approach is easier to interpret, is more flexible and yields lower total costs than the Class Approach. In addition to this, the ABA has lower computation times than the Multi-Item Approach and the Basic Blend Approach. However, the method can still yield unbalanced results with low stock levels for expensive slow-movers, because the generated optimal class target fill rate for expensive items is low. Therefore, this approach can serve as a starting point for the usage of one of the other approaches, whereby initial values for the class target fill rates are obtained.

Currently, Gordian finds that the Class Approach is the most practical approach to determining minimum base stock levels for spare parts. They are of the opinion that the Class Approach results in stock levels which are too high for expensive slow-movers. Furthermore, this method is highly dependent on the choice of the class target fill rates. Depending on a trade-off between the total costs and minimum base stock levels, the Basic Blend Approach and the Advanced Blend Approach could both be suitable and relevant for clients of Gordian as blends between an item approach and a system approach. Both approaches result in lower total investment costs compared to the Class Approach of Gordian. We find that the Advanced Blend Approach is the most applicable blend since it has the most similarities with the Class Approach and is the most interpretable one.

8 Discussion and Further Research

In this section, we discuss the limitations of the proposed methods in this research and we give some recommendations for further research. The first discussion point of this research is that we tested the performance of the described methods on just one data set and assumed that the demand rate of all items is constant. However, in practice, the demand rate is usually not constant. Furthermore, the lead times are determined as the average lead time over four years. In further research, one could apply the same methods with variable lead times and demand. For example, for the demand distribution, the Compound Poisson could be applied instead of the Poisson distribution.

The next discussion point is that we apply an $(S-1, S)$-policy for all individual items. However, this choice of policy is only realistic, because we do not consider fixed order costs. Especially for cheap fast-movers, it could be better to follow an (R, Q)-policy. In addition to this, the Greedy algorithm for the Multi-Item problem increases the stock level of one item by one per iteration. Adding a batching size of Q for items with high demand and low cost instead of one will decrease the computation time of the algorithm.

One of our assumptions is that all items are considered to be equally critical. In practice, not all items cause the entire system to fail when a failure occurs. An extension to the proposed methods could be to add weights to the items, based on their criticality.

In the Basic Blend Approach, we assume three cases to make a blend between the Class Approach and the Multi-Item Approach. The choices of these cases are made in cooperation with Gordian Logistic Experts B.V. (Gordian). The idea behind these cases is to lower the stock levels of the classes with expensive slow-movers compared to the Class Approach. An extension of this approach could be to not limit ourselves to the existing classes. For example, one could apply the same approach within the class containing expensive slow-movers.

In the Advanced Blend Approach, we optimise the class target fill rates. However, the thresholds separating the classes could also be optimised, as well as the number of classes. One option is to use the classes in the system approach while requiring that they have the same target. The more classes you define the closer you are compared to the system approach.

Another recommendation for the Advanced Blend Approach is about the presented Greedy algorithm for this approach. In this algorithm class fill rate targets are increased by 1% until the system target fill rate is achieved. In the results we see that the Advanced Blend Approach sets the target fill rate of class C3, containing expensive slow-movers, to 0.9% for system target fill rates lower than 99%. However, for 99% the class target fill rate of C3 jumps to 70.3%. An alternative can be to increase the class target fill rates by 0.1% instead of 1%, yielding in more precise optimised class target fill rates.

Finally, for the Multi-Item Approach with expected back orders as a target service level, we propose a Local Search algorithm to improve the solution obtained by the Greedy algorithm in the Multi-Item Approach. However, the Greedy algorithm already generates efficient solutions to the Multi-Item Problem. In our results, we see that the Local Search Algorithm decreases the stock levels of especially the classes with expensive slow-movers. The disadvantage of the Multi-Item Approach is that it tends to set really low stock levels for these items. So lowering their stock levels is not really an 'improvement'.

Bibliography

Axsäter, S. (2015). *Inventory control*, volume 225. Springer.

Basten, R. J. I. and van Houtum, G.-J. (2014). System-oriented inventory models for spare parts. *Surveys in operations research and management science*, 19(1):34–55.

Bounou, O., El Barkany, A., and El Biyaali, A. (2017). Inventory models for spare parts management: a review. *International Journal of Engineering Research in Africa*, 28:182–198.

Cardós, M., Guijarro, E., and Babiloni, E. (2015). Optimal multi-item fill rate determination for spare parts. In *2015 International Conference on Logistics, Informatics and Service Sciences (LISS)*, pages 1–5. IEEE.

Everett III, H. (1963). Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations research*, 11(3):399–417.

Feeney, G. J. and Sherbrooke, C. C. (1966). The (s-1, s) inventory policy under compound poisson demand. *Management Science*, 12(5):391–411.

Fisher, M. L. (1981). The lagrangian relaxation method for solving integer programming problems. *Management science*, 27(1):1–18.

Fox, B. (1966). Discrete optimization via marginal analysis. *Management science*, 13(3):210–216.

Karush, W. (1957). A queuing model for an inventory problem. *Operations research*, 5(5):693–703.

Kiesmüller, G. P., De Kok, A., and Dabia, S. (2011). Single item inventory control under periodic review and a minimum order quantity. *International Journal of Production Economics*, 133(1):280–285.

Palm, C. (1938). Analysis of the erlang traffic formula for busy-signal arrangements. *Ericsson Technics*, 5(9):39–58.

Porras, E. and Dekker, R. (2008). An inventory control system for spare parts at a refinery: An empirical comparison of different re-order point methods. *European Journal of Operational Research*, 184(1):101–132.

Rustenburg, J. W., Nicolaas, G.-J. J. J. A., and Zijm, W. H. M. (1998). *Spare parts management at the Royal Netherlands Navy: vari-metric and beyond.* BETA, Institute for Business Engineering and Technology Application.

Sherbrooke, C. C. (1968). Metric: A multi-echelon technique for recoverable item control. *Operations research*, 16(1):122–141.

Sherbrooke, C. C. (1992). Multiechelon inventory systems with lateral supply. *Naval Research Logistics (NRL)*, 39(1):29–40.

Sherbrooke, C. C. (2006). *Optimal inventory modeling of systems: multi-echelon techniques*, volume 72. Springer Science & Business Media.

Silver, E. A., Pyke, D. F., Peterson, R., et al. (1998). *Inventory management and production planning and scheduling*, volume 3. Wiley New York.

Smith, M. A. and Dekker, R. (1997). On the (s-1, s) stock model for renewal demand processes: Poisson's poison. *Probability in the Engineering and informational Sciences*, 11(3):375–386.

Thonemann, U. W., Brown, A. O., and Hausman, W. H. (2002). Easy quantification of improved spare parts inventory policies. *Management Science*, 48(9):1213–1225.

Van Houtum, G.-J. and Kranenburg, B. (2015). *Spare parts inventory control under system availability constraints*, volume 227. Springer.

Wolff, R. W. (1982). Poisson arrivals see time averages. *Operations research*, 30(2):223–231.

A Pseudocodes

In this section, we give pseudo codes for the methods used in this thesis.

A.1 Pseudocode Item Approach

In this thesis, the Item Approach is used with both the expected back orders and the fill rate as a service level. Hence, in this section, pseudo codes for both service levels are given. The algorithms for the Item Approach are based on Van Houtum and Kranenburg (2015).

A.1.1 Expected back orders as service level

The algorithm for the Item Approach searches for the smallest S_i such that $\mathbb{E}BO_i(S_i)$ is as small as possible. The input for the algorithm is the average demand μ_i, the purchase price c_i and the lead time L_i of all items $i \in I$. Furthermore, the target service level $\mathbb{E}BO^{obj}$ is given as an input.

The algorithm starts with the initialisation where all base-stock levels are set to zero, i.e. $S_i = 0$ for all items i. So $\mathbf{S} = (0, \ldots, 0)$, with corresponding total investment cost $C(\mathbf{S}) = 0$, which is the lowest possible total cost. The expected back orders for each item with $S_i = 0$ is $\mathbb{E}BO_i(S_i) = \mu_i Ł_i$, which is the highest possible total back orders. Furthermore, the item targets are set to $\mathbb{E}BO_i^{obj} := \frac{\mu_i}{M}\mathbb{E}BO^{obj}$ with $M = \sum_{i \in I} \mu_i$. In the next step of the algorithm, 1 is added to S_i for each item individually, until the item target is reached. From Lemma 4.1 originated from Van Houtum and Kranenburg (2015) we know that $\mathbb{E}BO_i(S_i)$ is decreasing and convex in S_i, so we know that in each iteration, the expected back orders are one step closer to the target. The algorithm ends by computing the total investment cost $C(\mathbf{S})$ and the aggregate means back orders $\mathbb{E}BO(\mathbf{S})$. A pseudocode for the algorithm for the item approach can be found in Algorithm 3.

A.1.2 Fill rate as service level

The algorithm for the Item Approach where the fill rate (FR) is used as a service level is similar to the algorithm above. The main idea is to increase the stock level S_i by 1 until the FR target FR^{obj} is achieved. From Lemma 4.2 originated from Van Houtum and Kranenburg (2015) follows that $FR_i(S_i)$ is increasing and concave on $\max\{\lceil \mu_i L_i - 1 \rceil, 0\}$. So, in the initialisation of the algorithm, set $S_i := \max\{\lceil \mu_i L_i - 1 \rceil, 0\}$ and compute the corresponding $C(\mathbf{S})$ and $FR_(\mathbf{S})$. And for each item the item FR target $FR_i^{obj} = FR^{obj}$ is set.

Algorithm 3 *(Algorithm for Item Approach)*

Input Average demand μ_i, Cost c_i, Lead time L_i, Target service level $\mathbb{E}BO^{obj}$

Step 1: Initialisation For all $i \in I$ set $S_i := 0$, so $\mathbf{S}= (0, \ldots, 0)$ and set $\mathbb{E}BO_i^{obj} := \frac{\mu_i}{M}\mathbb{E}BO^{obj}$;

$\mathbb{E}BO_i(S_i) := \mu_i L_i$; $\mathbb{E}BO(\mathbf{S}) := \sum_{i \in I} \mu_i L_i$ and $C(\mathbf{S}) := 0$.

Step 2: For all $i \in I$;

while $(\mathbb{E}BO_i(S_i) > \mathbb{E}BO_i^{obj})$

do $S_i := S_i + 1$ and compute $\mathbb{E}BO_i(S_i)$ with new S_i

Step 3: Determine $C(\mathbf{S})$ and $\mathbb{E}BO(\mathbf{S})$;

Output $\mathbf{S} = (S_1, \ldots, S_{|I|})$ and $C(\mathbf{S})$.

In the next step, the smallest S_i such that $FR_i(S_i)$ is greater than or equal to FR_i^{obj} is determined. The algorithm ends with computing the total investment cost $C(\mathbf{S})$ and the aggregate FR $FR(\mathbf{S})$. The pseudocode for the Item Approach with the FR as a service level is shown in Algorithm 4.

Algorithm 4 *(Algorithm for Item Approach with FR as service level)*

Input Average demand μ_i, Cost c_i, Lead time L_i, Target service level FR^{obj}

Step 1: Initialisation For all $i \in I$ set $S_i = \max\{\lceil \mu_i L_i - 1 \rceil, 0\}$ and set $FR_i^{obj} = FR^{obj}$;

Determine $FR_i(S_i) := \sum_{x=0}^{S_i - 1} \mathbb{P}[X_i = x]$; $FR(\mathbf{S}) := \sum_{i \in I} \frac{\mu_i}{M} FR_i(S_i)$ and $C(\mathbf{S})$.

Step 2: For all $i \in I$;

while $(FR_i(S_i) < FR_i^{obj})$

do $S_i = S_i + 1$ and compute $FR_i(S_i)$ with new S_i

Step 3: Determine $C(\mathbf{S})$ and $FR(\mathbf{S})$;

Output $\mathbf{S} = (S_1, \ldots, S_{|I|})$, $FR(\mathbf{S})$ and $C(\mathbf{S})$.

A.2 Pseudocode Class Approach with fill rate as service level

In the Class Approach items from different classes get different targets. First, all items $i \in I$ are classified in classes K_j, with $j = 1, \ldots, 9$, according to the classification matrix of Gordian Logistic Experts B.V. in Figure 1. Each class has its fill rate (FR) target $FR_{K_j}^{obj}$, which is defined by Gordian. After classification, we set $S_i = \max\{\lceil \mu_i L_i - 1 \rceil, 0\}$ and compute the corresponding aggregate FR and the total cost, $FR(\mathbf{S})$ and $C(\mathbf{S})$ respectively. For all items $i \in K_j$, we set individual item targets equal to the class target, so $FR_i^{obj} := FR_{K_j}^{obj}$. Then we increase the stock levels S_i by one until the target FR is reached, as in the Item Approach explained above. The algorithm ends by computing the total cost $C(\mathbf{S})$ and the aggregate FR, $FR(\mathbf{S})$. In Algorithm 5 the pseudocode is given for the algorithm to apply the Class Approach.

Algorithm 5 *(Algorithm for Class Approach with FR as service level)*

Input Average demand μ_i, Cost c_i, Lead time L_i, Class target service levels $FR^{obj}_{K_j}$

Step 1: Classification Classify all $i \in I$ in classes K_j, with $j = 1, \dots, 9$ and $I = \bigcup_{j=1}^{9} K_j$

Step 2: Initialisation For all $i \in I$;

Set $S_i := \max\{\lceil \mu_i L_i - 1 \rceil, 0\}$ and

Set for all $j = 1, \dots, 9$ and $i \in K_j$ $FR^{obj}_i := FR^{obj}_{K_j}$; Compute $C(\mathbf{S})$ and $FR(\mathbf{S})$.

Step 2: For all $i \in I$;

while $(FR_i(S_i) < FR^{obj}_i)$

do $S_i = S_i + 1$ and determine $FR_i(S_i)$

Step 3: Determine $C(\mathbf{S})$ and $\mathbb{E}BO(\mathbf{S})$;

Output $\mathbf{S} = (S_1, \dots, S_{|I|})$, $\mathbb{E}BO(\mathbf{S})$ and $C(\mathbf{S})$.

A.3 Pseudocode of Greedy Algorithm for Multi-Item Problem

In this thesis, the Multi-Item Problem is defined with both the expected back orders and the fill rate as service levels. Hence, this section contains pseudocodes of Greedy algorithms with both service levels which originate from Van Houtum and Kranenburg (2015).

A.3.1 Expected back orders as service level

The algorithm that we use to solve the Multi-Item problem is a Greedy-based approximation algorithm which originates from Van Houtum and Kranenburg (2015), which searches through a solution space which contains efficient solutions to the Multi-Item Problem. This follows from Theorem 2 in the article of Fox (1966).

The input of the algorithm is the average demand μ_i, the purchase price c_i, the lead time L_i for item $i \in I$ and the system target level $\mathbb{E}BO^{obj}$. First, the algorithm starts with setting $S_i := 0$ for all $i \in I$. So $\mathbf{S} = (0, \dots, 0)$, with corresponding total cost $C(\mathbf{S}) = 0$ and $\mathbb{E}BO(\mathbf{S}) = \sum_{i \in I} \mu_i L_i$. In the next step of the algorithm, the idea is to increase the stock level of just one item by one, one at a time. This item is chosen based on a greedy search. Each item gets a 'value' Γ_i which is equal to the difference in back orders concerning its cost. Using Equation 7, the increase in back orders $\Delta\mathbb{E}BO_i(S_i)$ can be defined as

$$\Delta\mathbb{E}BO_i(S_i) = \mathbb{E}BO_i(S_i + 1) - \mathbb{E}BO_i(S_i) = -\sum_{x=S_i+1}^{\infty} \mathbb{P}[X_i = x] = -\left(1 - \sum_{x=0}^{S_i} \mathbb{P}[X_i = x]\right).$$

So the decrease in back orders is equal to $-\Delta\mathbb{E}BO_i(S_i)$. The value Γ_i is equal to the largest contribution to the back orders concerning the increase of the total cost. So Γ_i can be defined by

$$\Gamma_i := \frac{(1 - \sum_{x=0}^{S_i} \mathbb{P}[X_i = x])}{c_i}.$$

The item with the biggest value for Γ_i is increased by one and $\mathbb{E}BO(\mathbf{S})$ and $C(\mathbf{S})$ are computed again. From Lemma 4.1 originated from Van Houtum and Kranenburg (2015) follows that $\mathbb{E}BO_i(S_i)$ is decreasing and convex, so this implies that increasing the stock level by one decreases $\mathbb{E}BO_i(S_i)$ and thus decreases $\mathbb{E}BO(\mathbf{S})$. The algorithm stops when $\mathbb{E}BO(\mathbf{S})$ is less than or equal to $\mathbb{E}BO^{obj}$.

The pseudocode of the Greedy Algorithm for the Multi-Item Problem with the expected back orders as a service level as described by Van Houtum and Kranenburg (2015) can be found in Algorithm 6.

Algorithm 6 *(Greedy Algorithm for Multi-Item Problem with* $\mathbb{E}BO$*)*

Input Average demand μ_i, Cost c_i, Lead time L_i, Target level $\mathbb{E}BO^{obj}$

Step 1: For all $i \in I$ set $S_i = 0$, so $\mathbf{S}= (0, \dots, 0)$;

$C(\mathbf{S}) := 0$ and $\mathbb{E}BO(\mathbf{S}) := \sum_{i \in I} \mu_i L_i$.

Step 2: For all $i \in I$ define Γ_i with new S_i;

$k :=\text{argmax}_{i \in I}\Gamma_i$; $\mathbf{S} := \mathbf{S} + \mathbf{e}_k$.

Step 3: Determine $C(\mathbf{S})$ and $\mathbb{E}BO(\mathbf{S})$;

If 'system target is reached', then STOP, else go to Step 2.

Output $\mathbf{S} = (S_1, \dots, S_{|I|})$, $\mathbb{E}BO(\mathbf{S})$ and $C(\mathbf{S})$

A.3.2 Fill rate as service level

The algorithm to solve the Multi-Item Problem where the fill rate (FR) is used as a service level is similar to the algorithm above. It is again a Greedy based approximation algorithm originated from Van Houtum and Kranenburg (2015). The input of the algorithm is the average demand μ_i, the purchase price c_i, the lead time L_i for item $i \in I$ and the system target level FR^{obj}. From Lemma 4.2 originated from Van Houtum and Kranenburg (2015), it follows that $FR_i(S_i)$ is increasing and concave on $\max\{\lceil \mu_i L_i - 1 \rceil, 0\}$. So, in the initialisation the algorithm sets $S_i := \max\{\lceil \mu_i L_i - 1 \rceil, 0\}$ and the corresponding $C(\mathbf{S})$ and $FR_(\mathbf{S})$ is computed.

In the next step of the algorithm, Γ_i which is equal to the increase in the aggregate fill rate divided by the increase in total cost is determined. Using Equation 9, the increase in item fill rate $\Delta FR_i(S_i)$ can be defined as

$$\Delta FR_i(S_i) = FR_i(S_i + 1) - FR_i(S_i) = \mathbb{P}[X_i = S_i].$$

Using Equation 8, the increase in the aggregate fill rate $\Delta_i FR(\mathbf{S})$ is equal to

$$\Delta_i FR(\mathbf{S}) = \frac{\mu_i}{M} \Delta FR_i(S_i) = \frac{\mu_i}{M} \cdot \mathbb{P}[X_i = S_i],$$

with $M = \sum_{i \in I} \mu_i$. Then Γ_i can be defined as

$$\Gamma_i := \frac{\mu_i \mathbb{P}[X_i = S_i]}{M c_i}.$$

The item with the biggest value for Γ_i is increased by one and $FR(\mathbf{S})$ and $C(\mathbf{S})$ are computed again. From Lemma 4.2 which originates from Van Houtum and Kranenburg (2015), it follows that $FR_i(S_i)$ is increasing and concave, so this implies that increasing the stock level by one increases $FR_i(S_i)$ and thus increases $FR(\mathbf{S})$. This step is repeated until $FR(\mathbf{S})$ is greater than or equal to FR^{obj}. The algorithm ends with computing the total cost $C(\mathbf{S})$ and the aggregate FR $FR(\mathbf{S})$. The pseudocode of the Greedy algorithm for the Multi-Item Problem with the FR as a service level originated from Van Houtum and Kranenburg (2015) is shown in Algorithm 7.

Algorithm 7 *(Greedy Algorithm for Multi-Item Problem with FR)*

Input Average demand μ_i, Cost c_i, Lead time L_i, Target level FR^{obj}

Step 1: For all $i \in I$ set $S_i := \max\{\lceil \mu_i L_i - 1 \rceil, 0\}$;

Compute $C(\mathbf{S})$ and $FR(\mathbf{S})$.

Step 2: For all $i \in I$ define Γ_i with new S_i;

$k :=\text{argmax}_{i \in I} \Gamma_i$; $\mathbf{S} := \mathbf{S} + \mathbf{e}_k$.

Step 3: Determine $C(\mathbf{S})$ and $FR(\mathbf{S})$;

If 'system target is reached', then STOP, else go to Step 2.

Output $\mathbf{S} = (S_1, \ldots, S_{|I|})$, $FR(\mathbf{S})$ and $C(\mathbf{S})$

B Mathematical proofs

This section in the appendix includes two mathematical proofs of Lemma 4.1 and Lemma 4.2, respectively. Both the Lemmas and the proofs originate from Van Houtum and Kranenburg (2015).

Proof Lemma 4.1

Proof. Let $i \in I$ be arbitrary. Then from section 4.1 we know that

$$\mathbb{E}BO_i(S_i) = m_i t_i - S_i + \sum_{x=0}^{S_i} (S_i - x)\mathbb{P}[X_i = x].$$

To show $\mathbb{E}BO_i(S_i)$ is decreasing on its whole domain, it is sufficient to show that its derivative $\Delta\mathbb{E}BO_i(S_i)$ is less than or equal to zero for some $i \in I$. This follows from the definition of a decreasing function. So:

$$\begin{aligned} \Delta\mathbb{E}BO_i(S_i) &= \mathbb{E}BO_i(S_i+1) - \mathbb{E}BO_i(S_i) \\ &= -\textstyle\sum_{x=S_i+1}^{\infty} \mathbb{P}[X_i = x] \\ &\leq 0, \end{aligned} \tag{19}$$

with $S_i \in \mathbb{N}_0$. Next, we will show the convexity of $\mathbb{E}BO_i(S_i)$ and this will be done by showing that the second derivative of $\mathbb{E}BO_i(S_i)$ is greater than or equal to zero for some $i \in I$. This follows from the definition of convexity. So:

$$\begin{aligned} \Delta^2\mathbb{E}BO_i(S_i) &= \Delta\mathbb{E}BO_i(S_i+1) - \Delta\mathbb{E}BO_i(S_i) \\ &= \mathbb{P}[X_i = S_i + 1] \\ &\geq 0, \end{aligned} \tag{20}$$

with $S_i \in \mathbb{N}_0$. From (19) and (20) it follows that $\mathbb{E}BO_i(S_i)$ is a decreasing and convex function. □

Proof Lemma 4.2

Proof. Let $i \in I$ be arbitrary. We know that the item fill rate FR_i is defined by

$$FR_i(S_i) = \sum_{x=0}^{S_i-1} P[X_i = x].$$

Thus the difference in FR when adding one to stock level S_i can be denoted by

$$\Delta FR_i(S_i) = FR_i(S_i+1) - FR_i(S_i) = \sum_{x=0}^{S_i} P[X_i = x] - \sum_{x=0}^{S_i-1} P[X_i = x] = P[X_i = S_i], \quad S_i \in \mathbb{N}_0.$$

And by definition of the probability, it follows that

$$P[X_i = S_i] \geq 0, \text{for} S_i \in \mathbb{N}_0,$$

which implies that $FR_i(S_i)$ is an increasing function. To prove that $FR_i(S_i)$ is concave for $S_i \geq \mu_i L_i$ it suffices to show that $\Delta FR_i^2(S_i) \leq 0$. This follows from the definition of a concave function. We have

$$\Delta FR_i^2(S_i) = P[X_i = S_i + 1] - P[X_i = S_i] \quad S_i \in \mathbb{N}_0. \tag{21}$$

We can rewrite $P[X_i = S_i]$ in a recursive way as:

$$P[X_i = 0] = e^{\mu_i L_i},$$
$$P[X_i = S_i + 1] = \frac{\mu_i L_i}{S_i + 1} P[X_i = S_i] \text{ for } S_i \in \mathbb{N}_0.$$

Substituting this into Equation 21 gives

$$\Delta^2 FR(S_i) = \left(\frac{\mu_i L_i}{S_i + 1} - 1 \right) P[X_i = S_i], \quad S_i \in \mathbb{N}_0.$$

This implies that

$$\Delta^2 FR_i(S_i) \leq 0 \Leftrightarrow \frac{\mu_i L_i}{S_i + i} - 1 \leq 0 \Leftrightarrow S_i \geq \mu_i L_i - 1 \tag{22}$$

In other words, the item fill rate $FR_i(S_i)$ is concave for $S_i \geq \mu_i L_i - 1$. We know that S_i is integral and nonnegative, so this implies that $S_i \geq \mu_i L_i - 1$ is equal to $S_i \geq \max\{\lceil \mu_i L_i - 1 \rceil, 0\}$.

So, $FR_i(S_i)$ is concave for $S_i \geq \max\{\lceil \mu_i L_i - 1 \rceil, 0\}$. □

C Data Analysis

In Figure 7 a scatter plot of relevant items in our data set, classified according to their demand frequency and price, following the Classification matrix of Gordian is given.

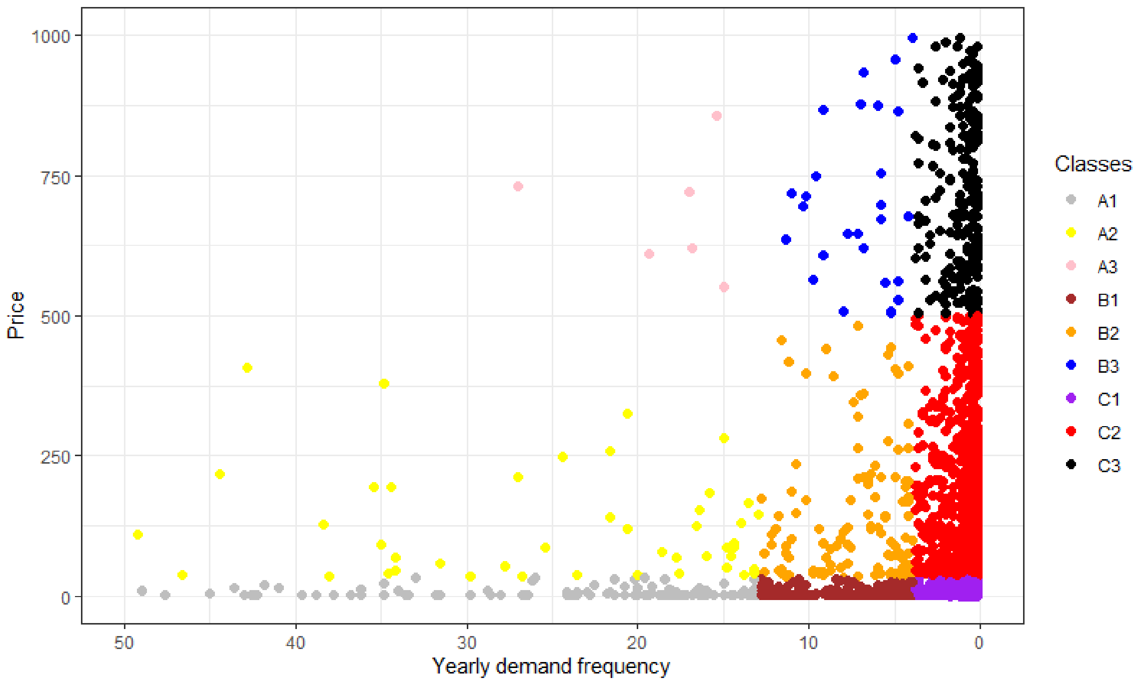


Figure 7: Scatter plot of items in our data set, classified according to their demand frequency and price. The colours represent different classes.

D Numerical Results

This section in the appendix includes several of the numerical results, deemed relevant for the reader. These results include both examples as results obtained from the case used to obtain the numerical results in Section 6.

D.1 Example of 20 items

We consider the system of 20 items displayed in Table 21, as described in Example 2 in Section 4.3.

Table 21: Example of 20 items

SKU id	μ_i (per year)	f_i (per year)	c_i (€)	L_i (years)
1	0.8	0.8	0.002	0.16
2	0.2	0.2	0.002	0.16
3	52.82	28.6	0.408	0.0082
4	9.4	4.2	52.44	0.0082
5	0.4	0.4	106.69	0.0082
6	15.6	1.2	1.93	0.0082
7	14.54	9.2	8.34	0.082
8	0.8	0.8	281.22	0.21
9	1.6	1.6	212.84	0.047
10	2.16	2.4	2963	0.16
11	0.6	0.4	0.002	0.082
12	0.4	0.4	65.91	0.019
13	0.2	0.2	56.62	0.0082
14	0.4	0.4	160.00	0.082
15	4.14	4.2	625.23	0.16
16	26.82	24.4	49.34	0.082
17	19.7	12.6	1.97	0.082
18	2.4	2.2	8.07	0.082
19	380	2.6	0.11	0.085
20	1	1	39.00	0.16

In Table 22 in Appendix D.1 we present the optimised class target fill rates for different system target fill rates.

Table 22: **(Example 2 - Cont'd)** System Targets versus Optimal Class Targets for the ABA

System Target	A1	A2	A3	B1	B2	B3	C1	C2	C3
80%	99.0%	35.4%	0.0%	77.8%	18.4%	0.0%	87.7%	0.0%	0.0%
82%	99.0%	35.4%	0.0%	99.4%	18.4%	0.0%	89.6%	0.0%	0.0%
84%	99.0%	35.4%	0.0%	99.4%	18.4%	0.0%	92.2%	0.0%	0.0%
86%	99.0%	35.4%	0.0%	99.4%	18.4%	0.0%	94.3%	0.0%	0.0%
88%	99.0%	35.4%	0.0%	99.4%	18.4%	0.0%	97.2%	0.0%	0.0%
90%	99.0%	35.4%	0.0%	99.4%	18.4%	0.0%	99.8%	0.0%	0.0%
92%	99.0%	81.8%	0.0%	99.4%	18.4%	0.0%	99.8%	0.0%	0.0%
94%	99.0%	99.2%	0.0%	99.4%	66.4%	0.0%	99.8%	0.0%	0.0%
96%	99.0%	99.2%	0.0%	99.4%	88.1%	0.0%	99.8%	0.0%	0.0%
98%	99.0%	99.2%	0.0%	99.4%	99.2%	50.6%	99.8%	99.8%	0.0%

In Table 23 we compare the results of the $ABA_{FR^{obj}}$ to the Multi-Item Approach ($MIA_{FR^{obj}}$) and Class Approach for three different target fill rates FR^{obj}.

Table 23: **(Example 2 - Cont'd)** Total stock per class S_{Class}, the total cost $C(\mathbf{S})$ and the aggregate fill rate $FR(\mathbf{S})$ obtained by Class Approach (CA) compared to the Item Approach ($\text{IA}_{FR^{obj}}$) with different fill rate targets FR^{obj}.

	S_{A1}	S_{A2}	S_{A3}	S_{B1}	S_{B2}	S_{B3}	S_{C1}	S_{C2}	S_{C3}	$C(\mathbf{S})$ (€)	$FR(\mathbf{S})$ (%)
$\text{MIA}_{75\%}$	1	2	0	1	1	0	44	0	0	113.92	75.6
$\text{MIA}_{90\%}$	4	2	0	4	1	0	62	0	0	126.25	90.2
$\text{MIA}_{99\%}$	6	8	0	8	8	1	79	7	3	1,912.08	99.0
CA	3	6	0	6	6	3	49	4	6	9,187.99	96.5
$\text{ABA}_{75\%}$	2	2	0	1	1	0	44	0	0	116.15	77.7
$\text{ABA}_{90\%}$	3	2	0	6	1	0	60	0	0	131.38	90.3
$\text{ABA}_{99\%}$	3	7	0	6	7	4	60	9	5	7,070.67	99.3

D.2 Numerical Results obtained for the Item Approach

In Figure 8 the total costs for different fill rate targets are shown.

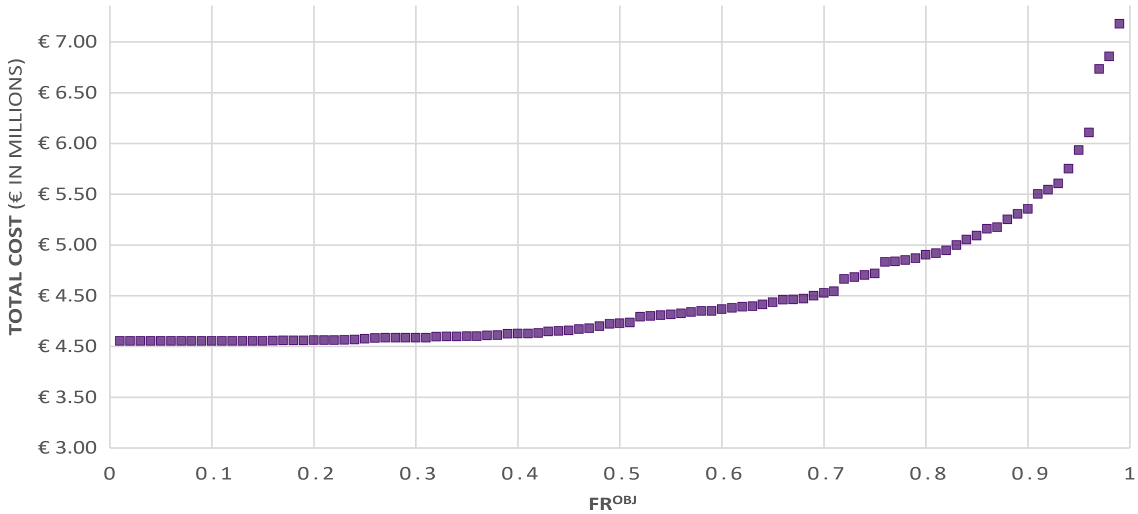


Figure 8: Total costs $C(\mathbf{S})$ plotted against different fill rate targets FR^{obj} following the Item Approach

In Figure 9 the total costs are shown for different expected back order targets.

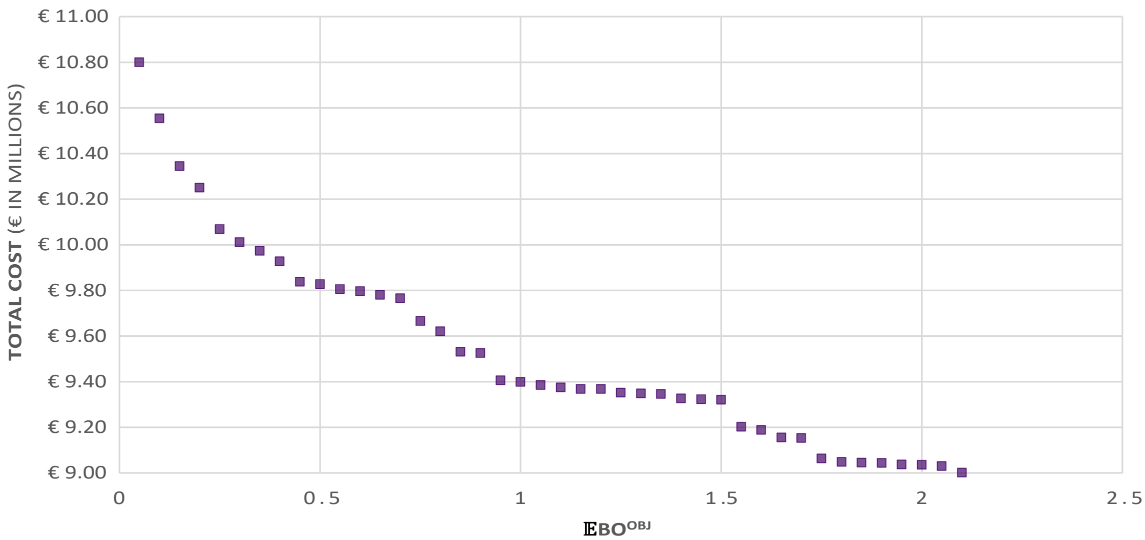
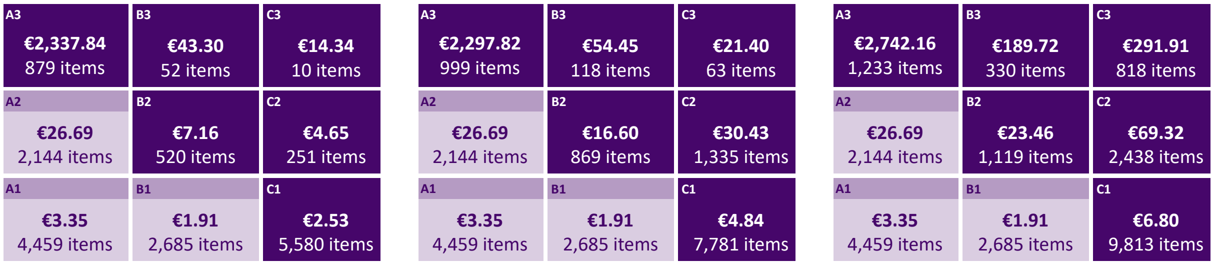


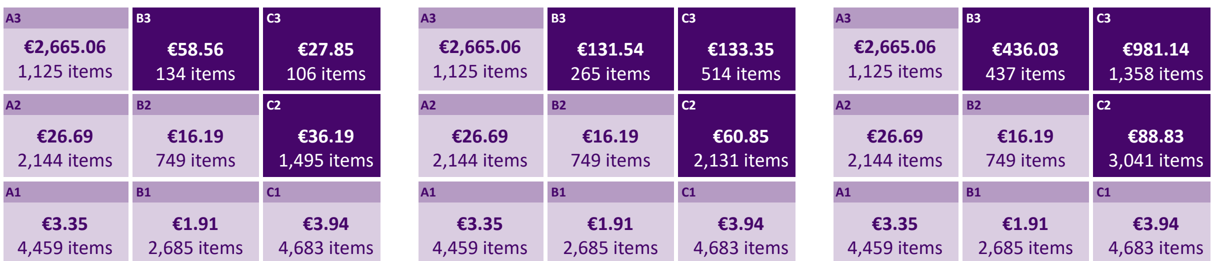
Figure 9: Total costs $C(\mathbf{S})$ plotted against different expected back-order targets $\mathbb{E}BO^{obj}$ following the Item Approach

D.3 Numerical Results obtained for the Class Approach



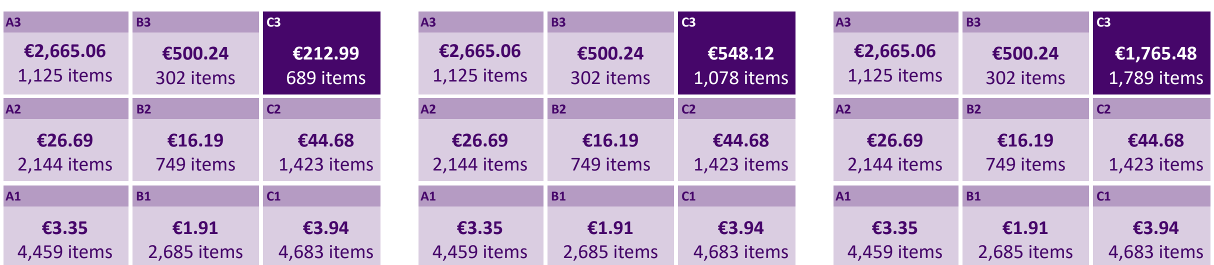
(a) $FR^{obj} = 75\%$ (b) $FR^{obj} = 90\%$ (c) $FR^{obj} = 99\%$

Figure 10: The total costs and total stock per class obtained by the Basic Blend Approach-I (BBA-I) with target levels of 75%, 90% and 99%, respectively.



(a) $FR^{obj} = 75\%$ (b) $FR^{obj} = 90\%$ (c) $FR^{obj} = 99\%$

Figure 11: The total costs and total stock per class obtained by the Basic Blend Approach-II (BBA-II) with target levels of 75%, 90% and 99%, respectively.



(a) $FR^{obj} = 75\%$ (b) $FR^{obj} = 90\%$ (c) $FR^{obj} = 99\%$

Figure 12: The total costs and total stock per class obtained by the Basic Blend Approach-III (BBA-III) with target levels of 75%, 90% and 99%, respectively.

D.4 Numerical results obtained for the Advanced Blend Approach

In Table 24 the optimal class targets for the Advanced Blend Approach at different FR^{obj} are shown.

Table 24: System Targets versus Optimal Class Targets for the Advanced Blend Approach

System Target	A1	A2	A3	B1	B2	B3	C1	C2	C3
75%	99.5%	49.0%	45.0%	99.7%	32.9%	12.1%	56.8%	16.8%	0.9%
76%	99.5%	49.0%	45.0%	99.7%	32.9%	12.1%	66.2%	16.8%	0.9%
77%	99.5%	49.0%	45.0%	99.7%	32.9%	12.1%	74.6%	16.8%	0.9%
78%	99.5%	49.0%	45.0%	99.7%	32.9%	12.1%	83.0%	16.8%	0.9%
79%	99.5%	49.0%	45.0%	99.7%	32.9%	12.1%	92.2%	16.8%	0.9%
80%	99.5%	50.1%	45.0%	99.7%	32.9%	12.1%	99.3%	16.8%	0.9%
81%	99.5%	54.1%	45.0%	99.7%	32.9%	12.1%	99.3%	16.8%	0.9%
82%	99.5%	57.8%	45.0%	99.7%	32.9%	12.1%	99.3%	16.8%	0.9%
83%	99.5%	62.4%	45.0%	99.7%	32.9%	12.1%	99.3%	16.8%	0.9%
84%	99.5%	66.0%	45.0%	99.7%	32.9%	12.1%	99.3%	16.8%	0.9%
85%	99.5%	70.3%	45.0%	99.7%	32.9%	12.1%	99.3%	16.8%	0.9%
86%	99.5%	74.4%	45.0%	99.7%	32.9%	12.1%	99.3%	16.8%	0.9%
87%	99.5%	78.9%	45.0%	99.7%	32.9%	12.1%	99.3%	16.8%	0.9%
88%	99.5%	82.8%	45.0%	99.7%	32.9%	12.1%	99.3%	16.8%	0.9%
89%	99.5%	87.1%	45.0%	99.7%	32.9%	12.1%	99.3%	16.8%	0.9%
90%	99.5%	90.5%	45.0%	99.7%	32.9%	12.1%	99.3%	16.8%	0.9%
91%	99.5%	94.6%	45.0%	99.7%	32.9%	12.1%	99.3%	16.8%	0.9%
92%	99.5%	99.7%	45.0%	99.7%	32.9%	12.1%	99.3%	16.8%	0.9%
93%	99.5%	99.7%	45.0%	99.7%	59.3%	12.1%	99.3%	16.8%	0.9%
94%	99.5%	99.7%	45.0%	99.7%	94.2%	12.1%	99.3%	16.8%	0.9%
95%	99.5%	99.7%	47.2%	99.7%	99.2%	12.1%	99.3%	55.0%	0.9%
96%	99.5%	99.7%	49.8%	99.7%	99.2%	12.1%	99.3%	99.7%	0.9%
97%	99.5%	99.7%	71.9%	99.7%	99.2%	12.1%	99.3%	99.7%	0.9%
98%	99.5%	99.7%	92.9%	99.7%	99.2%	12.1%	99.3%	99.7%	0.9%
99%	99.5%	99.7%	99.1%	99.7%	99.2%	50.6%	99.3%	99.7%	70.3%

When we look at Figures 13 and 14, we see that the costs of classes $A3$, $B3$, $C3$ are higher than the costs of the other classes, while their stock levels are really low. The reason for that is our data set is very unbalanced. For example, class C3 has many items with very low demand concerning their costs. So the algorithm is not likely to select class C3 to increase the class target fill rate.

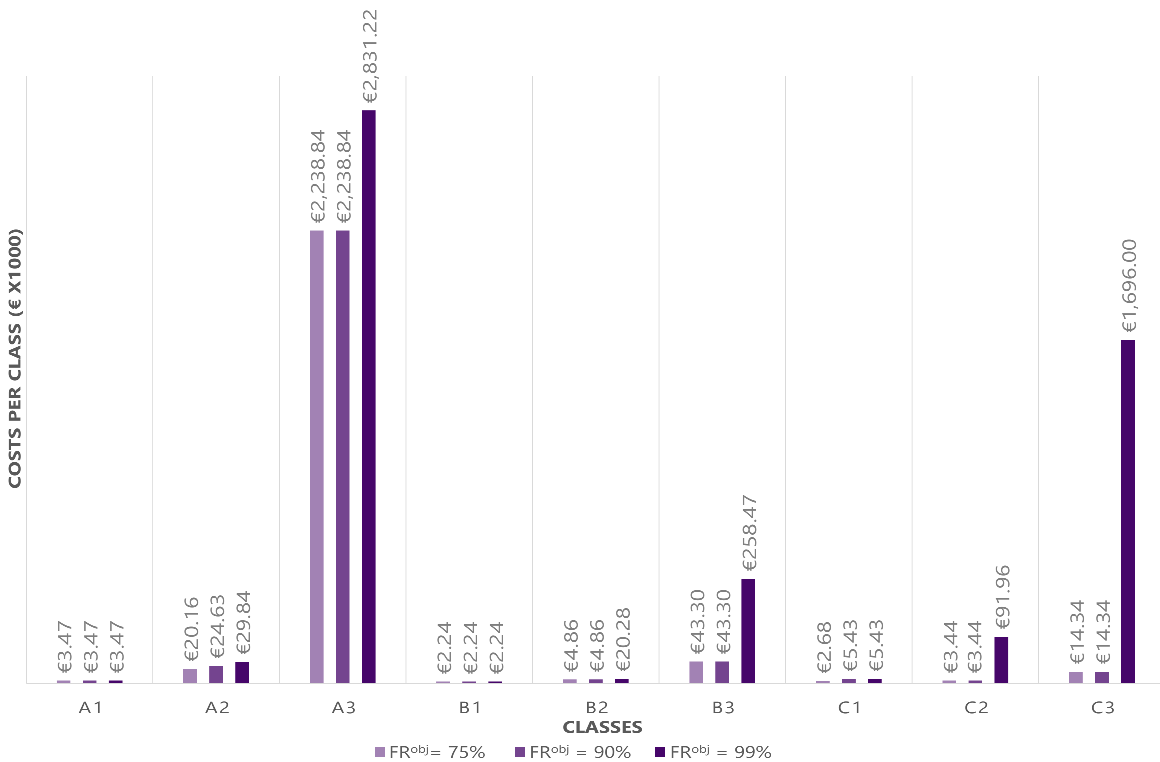


Figure 13: Total costs ($C(\mathbf{S})$) per class for different Fill Rate targets (FR^{obj}) - Advanced Item Approach

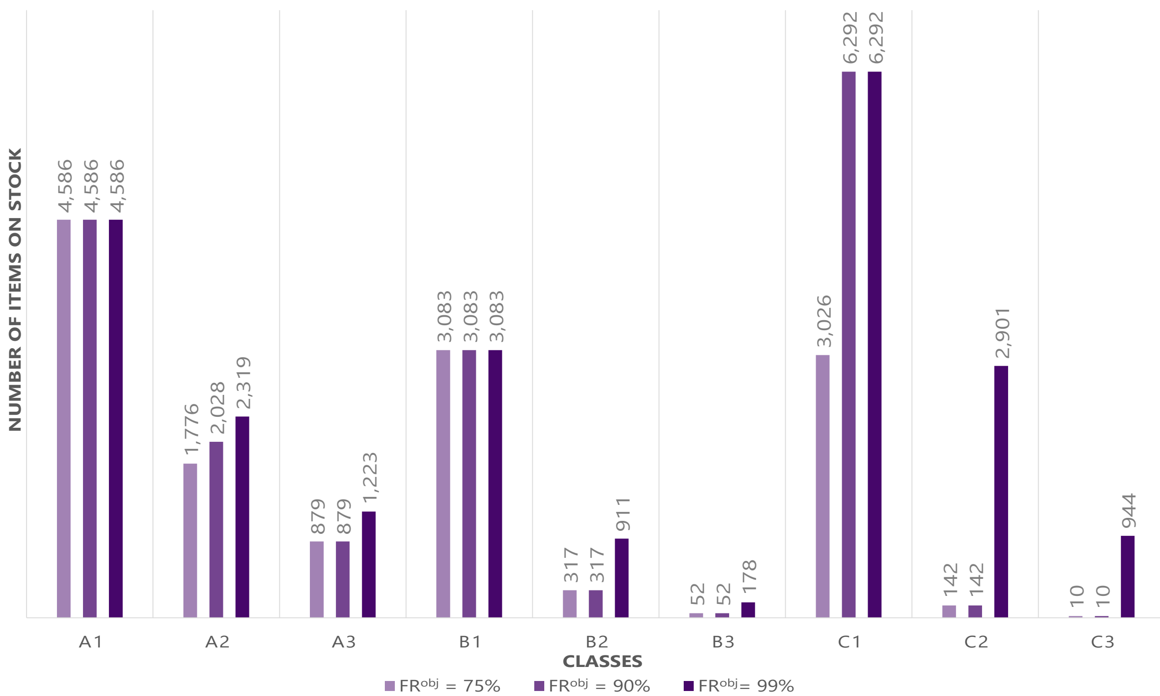


Figure 14: Total stock per class for different Fill Rate targets (FR^{obj}) - Advanced Item Approach