Erasmus University Rotterdam

Master Thesis

Operations Research & Quantitative Logistics

Robust Storage Assignment In 3D Warehouses

Author:
Wendy de Ridder
Student Number: 328100

Supervisor
Erasmus University:
Dr. R. Dekker
&
Second Reader
Erasmus University:
Dr. A. Gharezogli
&
Supervisor Ortec:
Martijn Martens

August 21, 2013

Abstract

This thesis addresses the storage-retrieval problem for a fully automated storage facility. We alter an existing deterministic and a factor-based storage and retrieval model to be applicable to a fully automated multi-deep storage facility. The models are able to handle uncertain supply and variable demand. Some practical attributes are added to the class based models, which can be influenced by the assignment of penalties. The results show that the model based on deterministic supply shows better results than the factor-based allocation model, when applied on a multi-deep facility.

Contents

Table of abbriviations

BOX	OptilogX
SAT	Satellite warehouse
VT	Vertical Transporter
HT	Horizontal Transporter
SIBA	The shuttle used in the SAT
I/O site	Infeed and Outfeed site

Table 1: Table of abbriviations

1 Introduction

The art of the econometrician consists in finding the set of assumptions which are both sufficently specific and sufficiently realistic to allow him to take the best possible advantage of the data available to him.

- Edmond Malinvaud

1.1 Why this study

Operations in warehouses are an essential part of the supply chain of any logistic company. In the last decades, technical development has made it possible to reduce human component in the warehousing process. This has lead to the development of fully automatic warehouses. In this study we will take a closer look at automatic warehouses with multiple floor and multi-deep lanes. The consultancy company *Ortec* has been assigned to the operational and tactical software design of this type of 3-D warehouse. This study will focus on the tactical allocation of the pallets in this 3-D warehouse, based on uncertain supply and variable demand.

As an operations research company *Ortec* has begun a partnership with the engineering company SPIE in the development of a fully automated warehouse. The responsibility of *Ortec* is to design the software which controls the movement of pallets in the warehouse. The first warehouse was placed at a Dutch icecream company in 2010. Ever since there are several warehouses build in The Netherlands, The United Kingdom and France.

As a decision maker on inventory in a warehouse, the whole supply chain movement can be influenced by the inventory management system. It is the responsibility of the manager to anticipate parameter ambiguity and stochastic uncertainty. In other words, he wants to use a robust inventory optimization system.

The operational efficiency of warehouses can play a crucial role in customer service and profit margins for any production company. In traditional warehouses, also called unit-load warehouses, all pallets are stored and retrieved with single pallet quantities. The basic characteristic of such warehouses is that no pallet is placed in front of another. These types of warehouses can be found both upstream and downstream in the supply chain.

Much research has been done on unit-load warehouses, however in this study we will consider multi-deep warehouses. In such warehouses multiple pallets can be stored in front of each other, meaning not every pallet can be reached directly. This type of warehouse is compact and very suitable for products that need to be stored in a cold environment. Up to this point, *Ortec* uses rule based decisions for the allocation and retrieval of pallets. The software does not anticipate on production schedules and there is no form of long term optimization present.

The warehouses considered in this study often have a production facility linked to the warehouse. Despite of carefully prepared production schedules, it can happen that production exceeds the expected quantity. It is desired to incorporate this excessive production and anticipate on it. In other words, the goal of this thesis is to design a robust allocation and retrieval model for a multi-deep warehouse. The model must have the ability to handle uncertain supply and variable demand and will optimize the worst-case scenario.

We start this paper by describing the pallet allocation and retrieval problem and give a description of the warehouses in section 2. In section 3 a literature review is given, describing previous research on pallet allocation problems and comparable problems in the field of container handling on container terminal. The mathematical formulation of the problem with a unit-load warehouse and deterministic demand is given in section 4. In section 5 this model is transformed into a robust allocation and retrieval model with uncertain supply and variable demand. In section 6 we will add the characteristics of a multi-deep warehouse to the model. Also different aspects of the layout of a warehouse are used to tune the model as the user desires. After this, we will validate the model by showing a small example solved by the model described in section 5. Next we will use a practical case Sodebo in section 8 to compair performance results of the different model types. Finally, we will make our conclusions and have a business recommandation for *Ortec*, which says which model performs best in partice.

2 Problem description

Supply chain optimization is a trending topic in operational research. The interaction between all components in the supply chain is an area which can result in a large inefficiency. One of the largest cost containing parts in the supply chain is warehousing. When communication between all components is optimal, many costs can be saved. This thesis will focus on the allocation and retrieval of pallets within a storage warehouse.

Compact, multi deep warehousing systems, similar to one described in Koster et al. (2012), have become more and more popular in the last decade. The extra dimension of multi deep aisles, makes such systems radical different from 2-D systems. Traditional optimized routing algorithms, storage strategies and capacity models cannot be applied on these systems.

The goal of this study is to design an allocation model for a multi-deep warehouse. The transformation of allocation models used in a 2-D warehouse to a multi-deep warehouse will be described. For practical usage of the model, it is important that the model can handle small changes in the supply as this can deviate from the expectation. Also demand is not constant in a warehouse, so it is also important that the model can handle variable demand.

The warehouse system on which this study is inspired, is an automated warehouse consisting of a warehouse and (optional) an OptilogX. The OptilogX is designed by engineering company SPIE and is described extensively at www.spie-nl.com. The OptilogX system is a fully automated storage, order picking and buffer system and is comparable to a sliding-puzzle. There are three main-components that provide the movements in three directions; vertical-transporters, horizontal-shuttles and conveyor-aisles. The design of this system leads to high throughput and sorting capacities in a compact space. This characteristic is mostly valued in warehouses which contain cooled or frozen products, where space is desired to be minimal.

The OptilogX can also be a part of a forward-reserve system where the OptilogX passes the incoming pallets to a storage warehouse (SAT). This warehouse is also fully automated but has much more storage positions, a low throughput capacity, but is cheaper than the OptilogX. These differences mostly originate from the change of using a shuttle which moves into the aisles (SIBA) instead of a chain conveyor in the aisles. In the case studies performed in this thesis an OptilogX is always present, but the pallet allocation and retrieval model will be applied on the SAT warehouse behind the OptilogX.

In case there is an OptilogX, the infeed pallets are transported by a VT to the destination level. There they are transferred on a shuttle which displaces the pallet to the destination aisle. When the destination of the pallet in in the SAT, the pallet is transported through the OptilogX to the shuttle on the other side of the OptilogX. On that side the SU is respectively displaced by a shuttle, a VT and then on the shuttle of the SAT.

The decision to store infeed pallets in the OptilogX or in the SAT is made when the pallets arrive in the forward-reserve system. Part of the pallets can be kept in the forward warehouse which can be used for short term storage. The rest of the pallets will be directly assigned to the SAT warehouse where they can be stored for a longer period.

Due to the third dimension, this type of warehouse is similar to container handling on a shipyard. Different levels within a warehouse can be seen as different storage yards. When containers are stacked on a storage yard, the placement of a container is similar to the placement of a pallet in the warehouse. The retrieval process can be compared as well. When a container has to be retrieved and it is not positioned at the top of a stack, the containers on top of the desired container have to be re-handled first. This is also the case if a desired pallet is not placed at the front of a lane.

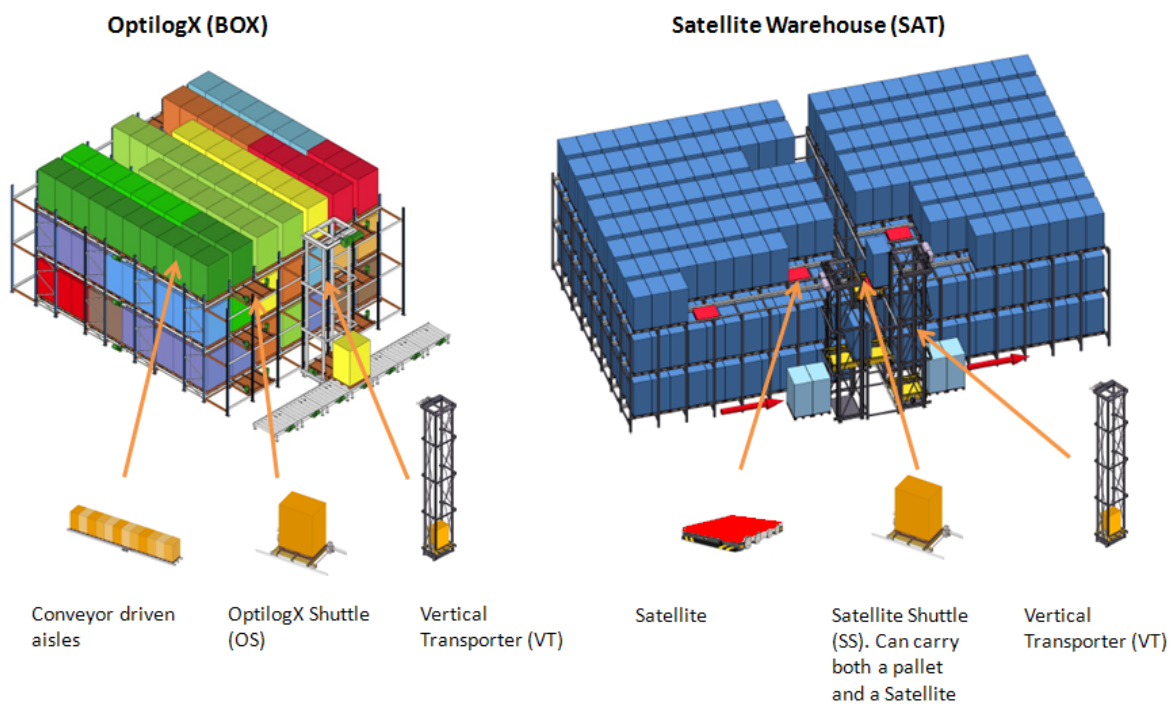


Figure 1: Overview OptilogX and OptilogX-SAT system

2.1 Test case warehouse description

The model will be tested on a practical case: the Obedos warehouse. Obedos produces products for the fresh catering market for supermarkets and hypermarkets and instant food. The Obedos warehouse consists of two components. First, a fast moving storage section (the BOX), which for example is a 17x15x4 storage unit at Obedos. The BOX handles the infeed and outfeed of pallets from outside the warehouse (left unit in Figure 2). The second component of the warehouse (right unit in Figure 2) is a satellite warehouse which can store

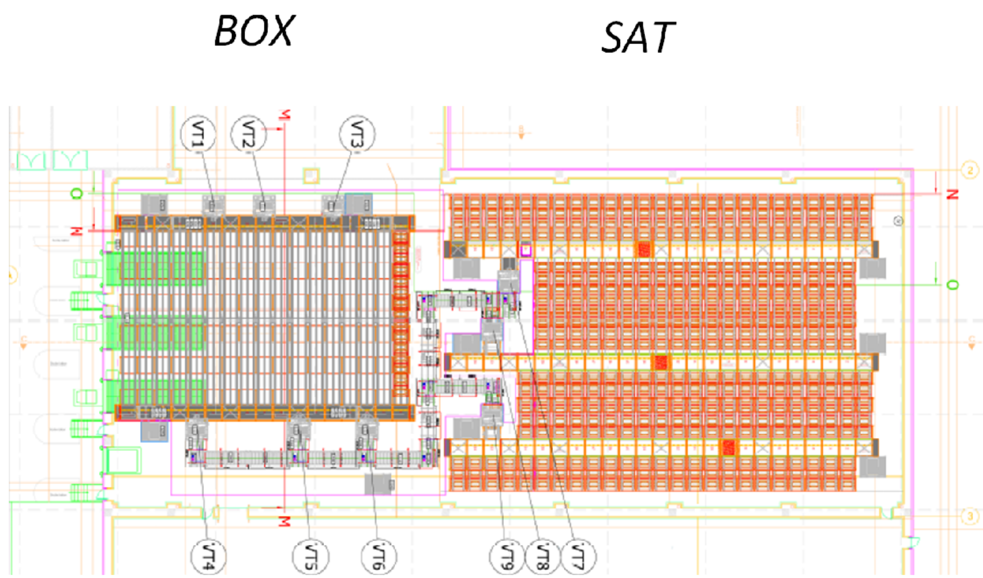


Figure 2: Schematic lay out of the Obedos warehouse

over 2700 pallets at Obedos. This satellite warehouse has 6 levels and pallets can move from the BOX to the SAT on the second level and from the SAT to the BOX on the fourth level. The BOX functions as a buffer zone for incoming batches or outgoing orders, where the SAT is used for long term storage.

2.1.1 Transporters

The pallets are moved through the warehouse via transporters. The box has horizontal transporters on every level at the front and back side of the pallet lanes. An example of this can be seen at the Obedos warehouse shown in Figure 2. On these HTs pallets can be moved within one level. On each level both HTs can receive or give a pallet to three vertical transporters. VTs can transport a pallet over different floors. Pallets enter or leave the BOX via the infeed and outfeed site. The I/O site is positioned on the ground floor under the BOX and can transport pallets from and to the BOX. Pallets can leave the box to the outside the warehouse via VT1 and VT2. VT4 and VT6 can transport pallets between the different levels of the box. These VTs do not move all the way down to the I/O site. This in contrary to VT5, which can move rejected pallets directly to the I/O site. When a pallet is moved from the BOX to the SAT, VT7, VT8 and VT9 can then transport the pallets from level 2 to different levels in the SAT. Even when a pallet is assigned a place on the second level, the pallet can only enter the SAT via VT7-VT9.

The SAT has four areas for storage, where the outer areas can only be accessed by one HT. Two middle areas can be accessed by two HTs. In these areas, pallets can be transported at the top and bottom of a lane. Due to the fact that the lanes are not moved by chain conveyors byt by SIBA, the mechanism does not allow for a row of pallets to move as a whole. Pallets can only be moved one at a time. For this study it is assumed that such a lane can be seen as two separate lanes of which the total length has to be equal to or smaller than the total lane size. The determination of the lane length can be fixed or variable. In the second case the lane length can be optimized depending on the infeed forecasts. This study however will only consider fixed lane depths.

2.2 Warehouse optimization

There are several elements of scheduling in a warehouse. First, products have to be stacked on a pallet. This is called pallet assignment. Second, storage assignment takes place, which means that pallet loads have to be assigned to storage locations. And last, the rules for sequencing storage and retrieval requests which is called interleaving. An optimal scheduling process can take place when all three elements are integrated in one model.

This thesis will focus on the optimization of the SAT warehouse. This optimization process exists out of three parts. First, the pallet placement problem. Given that a certain pallet enters the SAT, what would be its best position for placement. Second, the pallet retrieval problem. When the retrieval of a pallet causes extra pallet movements, the destinations of these movements have to be determined too. Last, when no placement or retrieval operations take place, the layout of the warehouse could be optimized. This process is called housekeeping. In this thesis we will only consider the movement of infeed and outfeed pallets. Housekeeping is not a part of this thesis.

2.3 Data availability

2.3.1 Infeed data

There are different types of infeed orders possible for a warehouse. In the case of Obedos the infeed orders origin at a production center where trucks with capacity of 18 are loaded and transported to the warehouse. These 18 pallets do not have to be of the same product type, but can consist out of multiple products as there are four product lines working in parallel at the production site. A production schedule can be used to determine the infeed of the warehouse, but this schedule is not always followed exactly. In practice production is based on a minimum number of products requested, but some loss of products is accounted for. This means that overproduction often takes place by a number of pallets. Due to this characteristic we introduce uncertain supply into our model.

2.3.2 Outfeed data

Outfeed orders can also occur in different types. First there is the case where there are only outfeed orders of size one. An example of this case is the outfeed procedure at Obedos, where the pallets move from the warehouse to a picking zone. When a pallet is picked empty a new request for a pallet is send to the warehouse. In this case the exact outfeed orders are not predictable, but outfeed over a period of time can be estimated properly.

It is more common that outfeed orders consist of batches of pallets. These outfeed orders can be multi-product or single-product orders. Unilever is an example of this situation. At Unilever trucks arrive to transport pallets to another location outside the warehouse. So in this case the order can be prepared in the BOX and pallets can be retrieved from the SAT and the BOX to store in prepare lanes inside the BOX near the I/O point.

2.3.3 Usage of prior knowledge: The current situation at Obedos

In the current planning software at the Obedos warehouse, prior knowledge about infeed and outfeed orders is not used very well. For infeed orders it can be known in advance which pallets arrive during the day. At the moment, when a pallet batch enters the BOX it is the first time the pallet ID's are scanned. The labels that are scanned contain all information about the pallet. Prior to the moment of scanning no information on the presence of the pallet is known to the warehouse software. Much improvement can be made when infeed and outfeed data over a wider period is available. A comparison between the current situation at Obedos and the results of this study are academically irrelevant and therefore outside the scope of this study.

2.4 Multi-deep storage lanes

One aspect of a 3D warehouse is the presence of multi-deep storage lanes. Pallets which are not stored in front of the lane cannot be accessed directly. This property can be a disadvantage if operators do not handle this wisely. The situation can occur where a pallet is stored in front of a pallet which is selected for retrieval. The desired pallet will have to be excavated by displacing the pallet in front of it. Excavating a pallet will consume much time and movements on a level and is therefore not desirable.

3 Literature

Pallet handling in a multi-deep warehouse has similarities in both pallet handling in a 2-dimensional warehouse and container handling on terminals. This research studies a forward and reserve warehouse which handles the handling of unit-loads in the form of pallets. In this section various literature on these subjects is collected.

3.1 Warehouse literature

In a unit-load warehouse all handled items are in the form of pallets. Each pallet contains only one product type and can be transported with one unit at a time. A comprehensive review on warehouse operations has been made in Goetschalckx et al. (2007). Goetschalckx et al. (2007) splits decisions concerning warehouses up into warehouse design and warehouse operation problems. Gunn et al. (1992) gives an overview of models on warehouse operation. Warehouse design falls outside the scope of this paper and is given for the Obedos case.

In the Obedos case we have two storage areas, also called a forward and reserve facility warehouse design. The allocation of pallets to these areas is described in Goetschalckx et al. (2010) and Sharp et al. (1998). Sharp et al. (1998) formulates the forward-reserve problem as a special case of the knapsack problem and proposes a greedy knapsack heuristic. For practical usages the paper also formulates a continues knapsack problem that considers replenishment limits. In warehouses there is often a maximum number of forklifts or manpower who can move pallets to their allocated places.

In order picking, different objectives can be considered. Leduc et al. (2007) did a literature review on warehouse order picking. Different order picking methods can be used in a warehouse, for example: single-order picking, batching and sort-while-pick, batching and sort- after-pick, single-order picking with zoning, and batching with zoning Sharp et al. (1998). Each order picking method consists of some or all of the following steps: batching, routing and sequencing, and sorting.

Travel time models are often used for the determination of storage locations. Babu et al. (1995) gives a critical review on automated storage and retrieval systems with a special emphasis on travel time models. The most commonly used time travel models are random assignment, full turnover-based assignment and class-based turnover assignment. Random assignment stores a pallet in the closest location. White et al. (1984) have used a statistical approach to develop expressions for travel time. A random storage model can be further improved by optimizing its dwell point, see Egbelu et al. (1991) for an LP-based minimization of the service response time.

Schwarz et al. (1976) model the turnover-based policy under the unrealistic assumption of shared storage, i.e. the storage space allocated to one product can only accommodate its average inventory level; no specific space is reserved to store the maximum inventory of a product. It is said to outperform the ABC-class based storage policy. However, when it is applied to a travel time model based on full turnover-based dedicated storage, Koster et al. (2013) show that the more practical full turnover-based dedicated storage policy outperforms the ABC-class based storage policy.

With class-based turnover assignment, a warehouse is divided into different classes. Pallets with the highest turnover rate are assigned to the class with the lowest associated costs. Methods for deriving the optimal boundaries for two or three storage regions are proposed bySchwarz et al. (1976). Eynan et al. (1989) and Rosenblatt et al. (1994) developed simple recursive procedures to derive optimal boundaries for a general n-class storage rack.

Ramirez et al. (1986) performs a computational study on the optimal stock picking decisions in an automatic S/R system. Like most storage and retrieval models the objective is to minimize picking costs in the form of traveling times. Kanet also considers the minimization of breakdowns, both with and without regard to picking costs. Breakdown costs cover the pallet picking process when an ordered quantity exceeds the volume of a pallet. Poort et al. (1998) made a comparison between the optimal routing of order pickers and a heuristic solution.

Previous literature was only applied on 2-dimensional warehouses. Research has been done as well in the field on 3-dimensional warehouses. The common assumption in this field is that there is one storage and retrieval (S/R) machine which can move horizontal and vertical simultaneously. This way the 3-dimensional pallet allocation problem is similar to the 2-dimensional problem. Cardin et al. (2013) presented a new storage-retrieval method called In-Deep Class Storage for 3-D warehouses. The method is based on the fact that it is more efficient to dedicate the front layers of each lane to the class of the most popular items, rather than dedicating whole bins close to the drop-off station. The disadvantage of this model is the usage of statistical demand, where the moment of retrieval is assumed to be known. Koster et al. (2012) have developed a sequencing heuristic for storing and retrieving unit loads in a 3-D compact automated warehousing system. The system also has an S/R machine which can move in both directions simultaneously. The lanes in the system they researched work like a carousel, so every pallet can be reached directly by the S/R machine without it having to excavate.

Ang et al. (2012) proposes a LP model which handles warehouses with variable supply and uncertain demand in a multi-period setting. A robust optimization model is introduced, which minimizes the worst-case expected total travel in the warehouse with distributional ambiguity of demand. Despite the imprecise specification of demand distributions, the computational experiments show that the model performs close to the expected value given perfect information and significantly outperforms existing heuristics in the literature.

3.2 Container handling literature

Papers on pallet movement in 2-dimensional warehouses and even for 3-dimensional warehouses do not cover the characteristics of the multi-deep lanes of the Obedos warehouse.

Pallets which are not located in the first bin of a lane have to be excavated. This property can be found in container stacking. Containers that are (un)loaded to a ship have to be placed in a shipyard on a maritime terminal first. Liu et al. (2003) solved the container stacking problem using a rolling-horizon approach. For each planning horizon, the problem was decomposed into two levels and each level is formulated as a mathematical programming model. First the distribution of pallets over the yard is determined. Second, the allocation of containers of each vessel to blocks takes place.

In container placement of inbound containers, a shipyard often has to handle containers, each with a different outfeed date. This causes stacks with containers with multiple outfeed dates and excavation of a container is not uncommon. Container movement costs a lot of time, therefore the objective of many papers concerning container placement is to minimize relocations of containers, next to the minimization of travel time. Kim et al. (2006) addressed a dynamic location problem as well as a static location problem. Both a genetic algorithm as simple heuristic rules are suggested and compared in a numerical experiment.

[13] studied the effect of efficient berth and quay crane schedules. This schedule takes arrival time of vessels also into account using a genetic algorithm. One of the assumptions made in this study is that the handling time of containers is known. Kim et al. (2012) handles the optimization of a mixed-block stacking storage system like a container yard. This paper builds its genetic algorithm on the average duration of stay per container type. Choe et al. (2011) proposes a stacking policy in an automated container terminal which consists out of two stages. First block determination and second slot determination. This method is a simple heuristic with weighted decision criteria which can be dynamic adjusted over multiple periods.

The container locating problem can also be approached in more detail if one considers only one ship yard.Schwarze et al. (2012) formulates a mathematical model for a 2-dimensional stacking area for multiple time periods. It also proposes a simple heuristic based on a simple rule of thumb. Each stack has a stack score, which helps to determine where a relocating block should be placed. Hong et al (2006) also works with the principle of having a stack score, which is formulated as a branch and bound algorithm. To limit the solution space of a relocation model, the corridor method proposed by Vo et al. (2009) could be applied.

In most warehouses the lay-out is fixed. In practice however it is possible to vary lane depth in some lanes. Goetschalckx et al. (1991) developed a procedure for selecting lane depths out of a limited number of allowable depths for the case of multi products. This is applicable for the Obedos warehouse also, but falls outside the scope of this thesis.

The paper that has the most similarities with the characteristics of multi-deep warehouses used in this study is the paper of Ang et al. (2012). The properties of variable supply and uncertain demand are incorporated and the warehouse is divided in multiple areas used for allocation. In the Obedos warehouse, these properties are the other way around. Supply is uncertain and demand is variable. Another deviation on the Ang et al. (2012) model is the property of multi deep lanes. We will incorporate this characteristic, but do not use any published literature for this as there is non found at this point.

4 Deterministic demand

In this section we will present a retrieval en allocation model with deterministic supply. This model is based on Ang et al. (2012) and will be the basis for all models described in this study.

4.1 Problem formulation

We start with a deterministic model, where demand and supply are known at the beginning of the first planning period for the entire planning horizon. Suppose there are M products indexed by $i = 1, ..., M$. The planning horizon is divided into T periods indexed by $t = 1, ..., T$. The warehouse is divided in N areas, called classes. We index these classes by $j = 1, ..., N$ and assume that all places in j have equal costs. For each period, we assume all pallets arrive at the start of the period and all pallets demanded from outside the warehouse are retrieved at the end of that period. The model described below can be applied on a unit-load warehouse and is the basis of the allocation model for the multi-deep warehouse. The objective of the model is to minimize the total expected costs over the entire planning horizon. Lets define $\mathcal{N} = \{1, ..., N\}, \mathcal{N}^- = \{1, ..., N-1\}, \mathcal{M} = \{1, ..., M\}, \mathcal{T} = \{1, ..., T\}$ and $\mathcal{T}^+ = \{1, ..., T+1\}$.

Let a_i^t be the number of pallets of product i which arrive at the start of period t. Let $v_{i,j}^t$ be the decision variable determining the number of arriving pallets of product i that are assigned to class j in period t. $w_{i,j}^t$ is the decision variable determining the number of outfeed pallets of product i that are retrieved from class j in period t. The warehouse is divided in different classes j, where all pallets entering the warehouse must be assigned to a class. This can be formulated as $\sum_{j \in \mathcal{N}} v_{ij}^t = a_i^t$, for $i \in \mathcal{M}, t \in \mathcal{T}$. In the same manner we can define the demand d_i^t as the number of pallets of product i that are ordered in period t. We now have $\sum_{j \in \mathcal{N}} w_{ij}^t = d_i^t$, for $i \in \mathcal{M}, t \in \mathcal{T}$. s_j are the storage costs for class j and r_j are the retrieval costs for class j. These costs can be equal, but do not have to be if the infeed and outfeed point are not at the same location in the warehouse for example.

For the inventory we say x_{ij}^t is the number of pallets of product i in class j at the start of period t. Where the start inventory at the beginning of period 1 can be equal to zero, but an extra constraint can be added where x_{ij}^1 is equal to a predefined start inventory. Also we assume that no backorders are allowed, so even after the last period this means $x_{ij}^t \geq 0$, for $i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T}^+$. We can define the inventory for period $t+1$ as $x_{ij}^{t+1} = x_{ij}^t + v_{ij}^t - w_{ij}^t$ for $i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T}$. The last component of this problem is the capacity constraint which cannot be exceeded: $\sum_{i \in \mathcal{M}} (x_{ij}^t + v_{ij}^t) \leq c_j$, for $j \in \mathcal{N}^-, t \in \mathcal{T}$.

The decision variables do not need to have an integer constraint, as the supply and demand parameters are integer, this is automatically the case. The linear optimization problem can be formulated as:

$$\min \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{M}} \sum_{j\in\mathcal{N}} (s_j v_{ij}^t + r_j w_{ij}^t) \tag{1}$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{N}} v_{ij}^t = a_i^t \quad \forall i \in \mathcal{M}, t \in \mathcal{T} \tag{2}$$

$$\sum_{j\in\mathcal{N}} w_{ij}^t = d_i^t \quad \forall i \in \mathcal{M}, t \in \mathcal{T} \tag{3}$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^t - w_{ij}^t \quad \forall i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T} \tag{4}$$

$$x_{ij}^1 = 0 \quad \forall i \in \mathcal{M}, j \in \mathcal{N} \tag{5}$$

$$\sum_{i\in\mathcal{M}} (x_{ij}^t + v_{ij}^t) \leq c_j \quad \forall j \in \mathcal{N}^-, t \in \mathcal{T} \tag{6}$$

$$x_{ij}^t \geq 0 \quad \forall i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T}^+ \tag{7}$$

$$v_{ij}^t, w_{ij}^t \geq 0 \quad \forall i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T} \tag{8}$$

In this model it is assumed that any shortage of inventory does not occur. It is the responsibility of the supplier to make sure there is sufficient inventory to meet the demand for every period. Equivalently,

$$\sum_{T=1}^{t} d_i^T \leq \sum_{T=1}^{t} a_i^T \quad \forall i \in \mathcal{M}, t \in \mathcal{T} \tag{9}$$

[1] has proven that the linear program is only feasible if and only if equation (9) holds. In case the initial inventory is not equal to zero, equation (9) can be rewritten as $\sum_{T=1}^{t} d_i^T \leq \sum_{T=1}^{t} a_i^T + \sum_{T=1}^{t} x_{ij}^1 \quad \forall i \in \mathcal{M}, t \in \mathcal{T}$.

As this model is a linear programming problem, it is solvable in polynomial time. The size however, can become very large if one of the sets contains a large number of elements. The size of the problem can become $\mathcal{T} \times \mathcal{M} \times \mathcal{N} \times 3$, because it applies to each of the 3 decision variables.

5 Stochastic demand

5.1 Factor-based demand model

In practice, supply and demand are often not known in advance. In this section we will base the model on uncertain demand and variable supply based on [1]. An estimation of the expected supply can be made based on production orders or historical data. The model described in this section will be able to handle this uncertain supply. In this section a factor based demand model will be introduced where supply for each product in period t is dependent on factors $\tilde{\mathrm{z}}_k$, $k = 1, ..., K_t$. K_t represents the number of demand factors used to model the supply up to period t, where $1 \leq K_1 \leq K_2 \leq ... \leq K_T$. At the end of period t $\tilde{\mathrm{z}}_k$, $k = 1, ..., K_t$ are known and new uncertain factors $\tilde{\mathrm{z}}_k$, $k = K_t, ..., K_{t+1}$ are introduced. For notational purpose we define $\kappa_t \equiv \{1, ..., K_t\}$, $\kappa_t^0 \equiv \{0, ..., K_t\}$.

Supply in period t for product i can now be written as a function of $\tilde{\mathbf{z}}^t$: $\mathrm{a}_i^t(\tilde{\mathbf{z}}^t) = \mathrm{a}_i^{t,0} + \sum_{k \in \kappa_t} a_i^{tk} \tilde{\mathrm{z}}^t$, for $i \in M, t \in T$. Making use of this definition of supply, a factor based optimization model can include correlation of supply for different products over different periods.

5.2 Stochastic robust optimization model

For the determination of storage and retrieval locations of pallets a model is desired which can take adjustability into account as information unfolds over time. To realize this, we introduce the vector $\tilde{\mathbf{z}}^t$, which contains all realized supply of period t. We repeat the following sequence of events for every period t: At the start of period t, the information $(\tilde{\mathbf{z}}^{t-1})$ is known. The storage locations of arriving pallets in period t are determined based on this information. The realized supply in period t is known at the end of period t and $\tilde{\mathbf{z}}^t$ becomes available, then the retrieval of pallets can take place. We can repeat this procedure every time period, using a rolling horizon.

Since the actual supply distribution is not known, we can consider multiple supply scenarios in the following robust optimization model.

$$Z_R = \min \quad \max_{\mathcal{P} \in \mathcal{U}} E_{\mathcal{P}} \quad \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} (s_j v_{ij}^t(\tilde{z}^t) + r_j w_{ij}^t(\tilde{z}^t)) \tag{10}$$

$$\text{s.t.} \sum_{j \in \mathcal{N}} v_{ij}^t(\tilde{z}^t) = a_i^t \quad \forall i \in \mathcal{M}, t \in \mathcal{T} \tag{11}$$

$$\sum_{j \in \mathcal{N}} w_{ij}^t(\tilde{z}^t) = d_i^t(\tilde{z}^t) \quad \forall i \in \mathcal{M}, t \in \mathcal{T} \tag{12}$$

$$x_{ij}^{t+1}(\tilde{z}^t) = x_{ij}^t(\tilde{z}^{t-1}) + v_{ij}^t(\tilde{z}^{t-1}) - w_{ij}^t(\tilde{z}^t) \quad \forall i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T} \tag{13}$$

$$x_{ij}^1(\tilde{z}^0) = 0 \quad \forall i \in \mathcal{M}, j \in \mathcal{N} \tag{14}$$

$$\sum_{i \in \mathcal{M}} (x_{ij}^t(\tilde{z}^{t-1}) + v_{ij}^t(\tilde{z}^{t-1})) \leq c_j \quad \forall j \in \mathcal{N}^-, t \in \mathcal{T} \tag{15}$$

$$x_{ij}^t(\tilde{z}^{t-1}) \geq 0 \quad \forall i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T}^+ \tag{16}$$

$$v_{ij}^t(\tilde{z}^{t-1}), w_{ij}^t(\tilde{z}^t) \geq 0 \quad \forall i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T} \tag{17}$$

5.3 2-D linear storage and retrieval model

When z_k is defined as the difference between the expected demand and the realization in scenario k. For example, if the expected demand of product i is 20 and scenario k describes a demand of 18, then $z_k = -2$. In this way all expected demand scenarios can be expressed in the terms of z_k. Taking in account all scenarios k, the linear program described below will optimize the storage and retrieval policy based on a worst case scenario. In other words, the model will assign a pallet location for the exceeding supply.

$$Z_{LR} = \min \quad g^0 + \max_{z \in \tilde{W}} \quad \sum_{k \in \mathcal{K}_T} g^k z_k \tag{18}$$

$$\text{s.t.} \quad g^k = \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} (s_j v_{ij}^{t,k} + r_j w_{ij}^{t,k}), \quad \forall k \in \mathcal{K}_T^0 \tag{19}$$

$$\sum_{j \in \mathcal{N}} v_{ij}^{t,k} = \begin{cases} a_i^t & \text{if } k = 0 \\ 0 & otherwise \end{cases} \quad \forall i \in \mathcal{M}, k \in \mathcal{K}_T^0, t \in \mathcal{T} \tag{20}$$

$$\sum_{j \in \mathcal{N}} w_{ij}^{t,k} = \begin{cases} d_i^{t,k} & \text{if } k \in \mathcal{K}_t^0 \\ 0 & otherwise \end{cases} \quad \forall i \in \mathcal{M}, k \in \mathcal{K}_T^0, t \in \mathcal{T} \tag{21}$$

$$x_{ij}^{t+1,k} = x_{ij}^{t,k} + v_{ij}^{t,k} - w_{ij}^{t,k} \quad \forall i \in \mathcal{M}, j \in \mathcal{N}, k \in \mathcal{K}_T^0, t \in \mathcal{T} \tag{22}$$

$$x_{ij}^{1,0} = 0 \quad \forall i \in \mathcal{M}, j \in \mathcal{N} \tag{23}$$

$$h_j^{t,k} = \sum_{i \in \mathcal{M}} (x_{ij}^{t,k} + v_{ij}^{t,k}) \quad \forall j \in \mathcal{N}, k \in \mathcal{K}_T^0, t \in \mathcal{T} \tag{24}$$

$$h_j^{t,0} + \sum_{k \in \mathcal{K}_T} h_j^{t,k} z_k \leq c_j \quad \forall z \in W, j \in \mathcal{N}^-, t \in \mathcal{T} \tag{25}$$

$$v_{ij}^{t,0} + \sum_{k \in \mathcal{K}_T} v_{ij}^{t,k} z_k \geq 0 \quad \forall z \in W, i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T} \tag{26}$$

$$w_{ij}^{t,0} + \sum_{k \in \mathcal{K}_T} w_{ij}^{t,k} z_k \geq 0 \quad \forall z \in W, i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T} \tag{27}$$

$$x_{ij}^{t,0} + \sum_{k \in \mathcal{K}_T} x_{ij}^{t,k} z_k \geq 0 \quad \forall z \in W, i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T}^+ \tag{28}$$

$$w_{ij}^{t,k} = 0 \quad \forall i \in \mathcal{M}, j \in \mathcal{N}, k \in \mathcal{K}_T \setminus \mathcal{K}_t, t \in \mathcal{T} \tag{29}$$

$$v_{ij}^{t,k} = x_{ij}^{t,k} = h_{ij}^{t,k} = 0 \quad \forall i \in \mathcal{M}, j \in \mathcal{N}, k \in \mathcal{K}_T \setminus \mathcal{K}_{t-1}, t \in \mathcal{T} \tag{30}$$

All previous models described in this paper are based on a warehouse with single deep lanes. These type of models can also be used in a multi-deep warehouse when all pallets for one lane are stored or retrieved in the same time period. This way the costs of storing or retrieving the lane are equal to the costs of storing or retrieving a pallet to the dedicated class $\times$ the capacity of the lane. Next we will describe a pallet allocation model that can be applied on a multi-deep warehouse where pallets are not necessarily allocated and retrieved per full lane.

6 3-Dimensional model description

6.1 Multi pallet lanes

In this section the assumption of multi-pallet lanes is introduced. In contrary to the models presented in previous sections, multiple pallets of the same product can be stored in one lane. By expanding the model with this characteristic, it is not necessary to buffer pallets in the BOX warehouse to the point one full lane can be filled. To avoid excavating movements, we assume only pallets of the same product can be stored in the same lane. The second assumption we make is that all pallets of the same product are exchangeable. By doing this, there will never be the need to excavate a pallet. In practice, it might be more efficient to store pallets of the same product in the same lane. Products with the latest expiration date will then have to be stored in the back of the lane. This way excavating will not take place, but you may not have sufficient storage space.

Constraint (24) and (25) are not applicable when single product lanes are assumed. As the capacity of a class is now defined by the number of lanes it contains and not the total number of pallets. Therefore capacity constraint (25) can be rewritten as:

$$\sum_{i \in \mathcal{M}} \sum_{k \in \mathcal{K}} (NrInventoryLanes_{i,j}^{t,k} + NrInfeedLanes_{i,j}^{t,k}) \leq Capacity_j \quad \forall j \in \mathcal{N}, t \in \mathcal{T} \tag{31}$$

Where the number of inventory lanes can be expressed as:

$$0 \leq \quad (LaneDepth \times NrInventoryLanes_{i,j}^{t,k}) - x_{i,j}^{t,k} \quad \leq \quad LaneDepth - 1 \tag{32}$$
$$\forall i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T}, k \in \mathcal{K}$$

$$NrInventoryLanes_{i,j}^{t,k} \in \mathbb{Z} \quad \forall i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T}, k \in \mathcal{K} \tag{33}$$

$NrInventoryLanes_{i,j}^{t,k}$ will be forced to be the minimal number of inventory lanes required by constraint (32). Constraint (33) is a necessary constraint to make sure the number of lanes is an integer.

When determining the number of infeed lanes it is assumed that the minimum number of lanes is used for storage within a class. This can be accomplished by keeping track of the rest inventory for period t in every class. $RestInventory_{i,j}^{t,k}$ is the number of open pallet places in a lane where product i is present at the beginning of period t in class j. As it is assumed that all not fully stored lanes will be filled before a new lane is used, we can use the $RestInventory$ to determine how many infeed pallets are needed to fill this lane. We can subtract the $RestInventory$ from the number of allocated pallets of product i in class j in period t to know how many lanes we need to reserve to cover the infeed.

$$RestInventory_{i,j}^{t,k} = (NrInventoryLanes_{i,j}^{t,k} \times LaneDepth) - x_{i,j}^{t,k} \tag{34}$$
$$\forall i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T}, k \in \mathcal{K}$$

Where

$$0 \leq \quad (LaneDepth \times NrInfeedLanes_{i,j}^{t,k}) - (v_{i,j}^{t,k} - RestInventory_{i,j}^{t,k}) \quad \leq \quad LaneDepth - 1 \tag{35}$$
$$\forall i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T}, k \in \mathcal{K}$$

6.2 Practical additions

The objective of the current model is to minimize the total travel costs over the time horizon. This may not be the best objective for operational purposes as there are more factors to be considered for the allocation schedule to be optimal. In the following sections a few aspects are further investigated.

6.2.1 Demand Leveling

The model described above follows the *Just In Time* retrieval policy. The pallets that are selected for an outfeed order are retrieved the time period the outfeed order is issued. This strategy makes sure that the SAT warehouse is used as a storage facility and the BOX as a buffer. However, this strategy can cause peak activity at different moments of the day. In a warehouse a constant level of activity over the day is desired. A method to level the activity over the different time periods is to assign a penalty when the total activity in the warehouse in period i exceeds the average activity over the day. As we assume the supply flow cannot be shifted over the day, the outfeed orders can be prepared in reserved lanes in the BOX warehouse. However, the total number of pallets retrieved at period t should be at least the cumulative demand at period t. Equation [21] can now be altered to:

$$\sum_{j \in \mathcal{N}} Cumw_{ij}^{t,k} \geq \begin{cases} Cumd_i^{t,k} & \text{if } k \in \mathcal{K}_t^0 \\ 0 & otherwise \end{cases} \quad \forall i \in \mathcal{M}, k \in \mathcal{K}_T^0, t \in \mathcal{T} \tag{36}$$

Where

$$Cumd_i^{t,k} = \sum_{r=1}^{t} d_i^{t,k} \quad \forall t \in \mathcal{T}, k \in \mathcal{K} \tag{37}$$

$$Cumw_{ij}^{t,k} = \sum_{r=1}^{t} \sum_{j \in \mathcal{N}} w_{ij}^{t,k} \quad \forall t \in \mathcal{T}, k \in \mathcal{K} \tag{38}$$

The equations stated below describe for every period the number of pallets that exceeds the average activity. If leveling over the periods is desired, the exceeding number of pallets can be penalized. By defining a CrossTime variable this way, it is implicitly assumed the warehouse can handle the average activity. If this is not the case, the warehouse is not used or designed properly. However, this topic falls outside the scope of this thesis.

$$AverageActivity = \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{M}} a_i^t + d_i^{t,0}/T \tag{39}$$

$$CrossTime_t \geq \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} (w_{ij}^{t,0} + v_{ij}^{t,0}) - AverageActivity \tag{40}$$

$$CrossTime_t \geq 0 \tag{41}$$

6.2.2 Class Leveling

The type of warehouse considered in this thesis consists of different levels. Each level contains a shuttle with capacity one to transport pallets to the class(es) located on that level. The different levels are connected with one or multiple VTs. In a warehouse with this structure, it is not desirable to have all pallets allocated on one level. If the activity within one time period is not spread over the different floors, the utilization of the shuttle capacity is not optimal. One of the consequences of allocating all pallet activity to one level is that the maximum shuttle capacity is exceeded and not all pallet movements can be executed within the desired time period. To level the activity over the different floors the following equations can be used:

$$ClassAverage_t = \sum_{j \in \mathcal{N}} \sum_{i \in \mathcal{M}} (V_{i,j}^{t,0} + W_{i,j}^{t,0})/|\mathcal{N}| \quad \forall t \in \mathcal{T} \tag{42}$$

$$\sum_{i \in \mathcal{M}} W_{i,j}^{t,0} + V_{i,j}^{t,0} - LevelAverage_t \leq CrossClass_{j,t} \quad \forall t \in \mathcal{T}, j \in \mathcal{N} \tag{43}$$

$$CrossClassl_{j,t} \leq 0 \tag{44}$$

In equations (42) - (44) the $CrossLevel_{f,t}$ variable is defined. $CrossLevel_{f,t}$ notes the number of pallets that exceed the average level activity in class j in period t. In practice it can occur that only a selected group of classes is considered for leveling, because otherwise, for example, the preferable classes are not filled up if they get leveled. This situation is probably not desirable.

6.2.3 Daily Infeed Rate

In the model described so far, there is no distinction made between the different product pallets that are not retrieved within the time horizon. In practice it might be desirable to make a distinction between products with a high turnover rate and pallets with a low turnover rate. In traditional ABC-allocation models, products with a high turnover rate are placed in the front and products with a low turnover rate are placed in the back of the warehouse. By including the daily infeed rate/turnover rate, the slow moving products can be placed deeper in the warehouse than high turnover rate products. The turnover rate can be based on historical data and future data if it is available. By making a division by turnover rates, the travelling costs can be reduced over a period beyond the planning horizon of the proposed model.

The turnover rate can be incorporated in the objective function by assigning storage costs proportional to the turnover rate. This can be done in several ways. One can add the costs to the original objective function or the turnover rate can be incorporated by multiplying the storage costs with the corresponding turnover rate. When using the latter, the proportion of the storage costs to the retrieval and potential penalty costs will be completely different and can cause undesired allocation decisions. Therefore we have chosen to incorporate the turnover rate by adding the values as shown in equation (46).

6.2.4 Planning horizon

By using the daily infeed rate to determine the allocation of pallets, it is useful to model with a rolling horizon. The length of the horizon has to be at least the maximum of (batch size of product i / daily demand). By taking this value for the time horizon, at least one batch of every product has entered the warehouse. The result of this characteristic is that the most efficient storage location for every product is achieved. The rolling horizon ensures that this property is always valid.

6.2.5 Objective function

To integrate demand leveling, class leveling and daily infeed rates over the periods, then the objective function (18) can be altered. For the integration of $CrossTime_t$ and $CrossClass_{j,t}$, these values can be penalized in the objective function. Equation (18) and (19) can now be written as:

$$Z_{LR} = \min \quad g^0 + \max_{z \in \tilde{W}} \quad \sum_{k \in \mathcal{K}_T} g^k z_k \tag{45}$$

$$\text{s.t.} \quad g^k = \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{M}}\sum_{j\in\mathcal{N}}(s_j v_{ij}^{t,k} + r_j w_{ij}^{t,k}) + \sum_{t\in\mathcal{T}}(TimePenalty \times CrossTime_t)$$
$$+ \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{N}}(ClassPenalty_j \times CrossClass_{j,t}) + \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{M}}\sum_{j\in\mathcal{N}}(TurnoverRate \times s_j v_{ij}^{t,k}),$$
$$\forall k \in \mathcal{K}_T^0 \tag{46}$$

6.3 Model specification

The models that can be applied on a BOX or SAT-warehouse are described in section 6.2. All different versions of the allocation model have the property of multi pallet deep lanes. The different versions of the models are declared as follows:

1. The model presented in section 6.1 where the objective is to minimize the total travel distance with variable supply
2. The model presented in section 6.3.1 where demand leveling is included
3. The model presented in section 6.3.2 where class leveling is also included, for j $\geq$ 7
4. The model presented in section 6.2 where the daily infeed rate is also included

7 Validation

Throughout this thesis different additions have been made to the model presented by [1] and shown in section 5. In section 6 these additions are presented. In this section we will use a small case to demonstrate the allocation model presented in section 6.2.5. The dataset will consists out of 3 products, 2 time periods, 4 classes and 2 floors.

7.1 Example

To illustrate the model shown in section 5 this section will be used to show a small example. This example uses 4 classes, 2 floors and 3 products. The lane depth is 2 pallets and $z_k = 2$ for all k. It is easy to verify in table 2 and 3 that there is enough inventory to meet demand for all $\mathbf{z}$ in each period.

	t=1		t=2	
i	a_i^1	$d_i^1(\tilde{\mathbf{z}}^1)$	a_i^2	$d_i^2(\tilde{\mathbf{z}}^2)$
1	10	$2+z_1$	5	$2+z_4$
2	10	$5+z_2$	5	$5+z_5$
3	10	$0+z_3$	5	$7+z_6$

Table 2: Number of arrivals and demand for each product in each period

Class	Floor	Storage Costs	Retrieval Costs	Capacity
1	1	10	10	5
2	1	20	20	10
3	2	15	15	7
4	2	30	30	100

Table 3: Warehouse Layout

Let $\mathbf{v}^{t,k}$ and $\mathbf{w}^{t,k}$ be 3x4 matrices with $v_{i,j}^{t,k}$ and $w_{i,j}^{t,k}$ as their (i,j) entries respectively. Solving model 1 gives

$$v^{1,0} = \begin{pmatrix} 2 & 6 & 2 \\ 8 & 3 & 0 \\ 0 & 1 & 8 \\ 0 & 0 & 0 \end{pmatrix} \quad v^{1,1} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad v^{1,2} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad v^{1,3} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{v}^{2,k} = \mathbf{0}, \quad k = 4, ..., 6$$

The matrix $v^{1,0}$ shows the allocation of product i for class j in period 1. In this example 2, 8, 0 and 0 pallets of product 1 are assigned to class 1 to 4 in the first period. The matrices $v^{1,1}$ to $v^{1,3}$ show that the variable supply of product 1 and 2 are allocated to class 3 and the extra supply of product 2 may be stored in class 2.

It is easy to see that this allocation complies with the capacity constraint. If we look at class 2 for example, we see that 8 pallets of product 1 and 3 pallets of product 2 are assigned. As the capacity is expressed in number of lanes and the lane depth equals two, there are six lanes occupied by the initially. We can see in matrix $v^{1,2}$ that the extra supply of product 2 is allocated to class two as well, which fills up the capacity of 7 lanes exactly.

Given the coefficients of $\mathbf{v}^{t,k}$, once supply in period 1 is realized, we can determine the allocation decisions for period 1 according to $v^1_{ij}(\tilde{\mathbf{z}}^1) = v^{1,0} + v^{1,1}_{i,j}\tilde{z}_1 + v^{1,2}_{i,j}\tilde{z}_2 + v^{1,3}_{i,j}\tilde{z}_3$. Supposed the realized supply is 3, 5 and 2 pallets for product 1-3 resp. Using product 1 for illustration we have

$$\begin{pmatrix} v^1_{1,1}(\tilde{\mathbf{z}}^1) \\ v^1_{1,2}(\tilde{\mathbf{z}}^1) \\ v^1_{1,3}(\tilde{\mathbf{z}}^1) \\ v^1_{1,4}(\tilde{\mathbf{z}}^1) \end{pmatrix} = \begin{pmatrix} 2 \\ 8 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}(1) + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}(0) + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}(2) = \begin{pmatrix} 2 \\ 8 \\ 1 \\ 0 \end{pmatrix}$$

The solution of model 1 shows the coefficients $w^{1,k}_{i,j}$ as follows

$$w^{1,0} = \begin{pmatrix} 2 & 6 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{w}^{1,k} = \mathbf{0}, \quad k = 1, ..., 6$$

As we allow retrieval in prior periods, we see that in this example 1 pallet of product 2 and 3 pallets of product 3 are retrieved earlier than necessary. This creates storage space for period 2, where the allocation is as follows:

$$v^{2,0} = \begin{pmatrix} 2 & 4 & 4 \\ 0 & 0 & 0 \\ 3 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad v^{2,4} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad v^{2,5} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad v^{2,6} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{v}^{2,k} = \mathbf{0}, \quad k = 1, ..., 3$$

It may appear odd that in the first period only one pallet of product two is placed in class 3. This allocation is explained by the allocation in the second period. No pallet is retrieved from the third class and the lane is filled up with 1 pallet in the second period.

Due to the symmetry and littleness of this example this solution is not unique, but it is one of the optimal solutions which is found.

Finally, the retrieval coefficients of the second period are:

$$w^{2,0} = \begin{pmatrix} 2 & 4 & 4 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{w}^{2,k} = \mathbf{0}, \quad k = 1, ..., 6$$

After drawing of this example we can summarize the procedure of the policy as follows: We first obtain coefficients $v_{i,j}^{t,k}$ and $w_{i,j}^{t,k}$ by solving a linear optimization problem. After supply is realized in each period t, we derive the factors $\tilde{\mathbf{z}}^t$. Finally we determine the operational storage and retrieval decisions according to $v_{ij}^t(\tilde{\mathbf{z}}^t) = v^{t,0} + \sum_{k=0}^{\mathcal{K}^t} v_{i,j}^{t,k} \tilde{z}_k$.

8 Case studies

8.1 Obedos case

In this section we will apply the models developed in section 5 and 6. We will look at the effect of the additions in the models on the pallet allocation. Based on the results we hope to give a founded performance review and a business recommendation for practical usage of the model. All results in this study are obtained while using the CPLEX 12.5 solver of the AIMMS 3.13 package on a computer with a Core i5-3380M, 2.90 GHz processor with 8,00 GB RAM.

8.1.1 The Company Obedos

Obedos is one of Ortecs customers where the OptilogX is in operation. Obedos is a French company which produces instant meals and sandwiches. These products can be found at almost every gas station in France and parts of Belgium. Thirty-five years have passed since Obedos began producing and marketing products that are fresh, tasty and innovative. Today, they are the leader in France in the fresh deli products segment. This independent, family-owned company employs more than 2,000 staff members.

8.1.2 Data

For this study we have inventory, infeed and outfeed data available over the period 17 April 2011 to 22 April 2011. In this period 438 different products were moved into and/or out of the warehouse. Almost all activity in the Obedos warehouse takes place from Monday until Friday. Figure 3 shows the infeed and outfeed activities over this period.

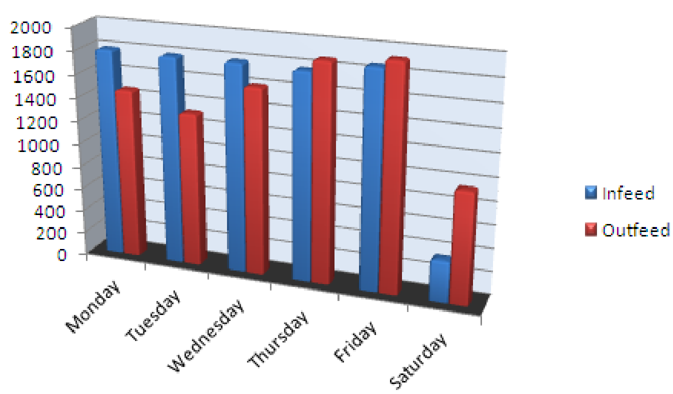


Figure 3: Infeed and Outfeed activities at Obedos

As can be seen in figure 3, the infeed quantity over the weekdays in the studied period is nearly constant. The outfeed quantity seems to increase over the weekdays. The activity on the Saturday is not even half the activity on the calmest weekday. However, it is noticeable that the outfeed on the Saturday is almost three times higher as the infeed.

When we look at the data, we can apply a form of ABC classification policy. If the volume of production is set against number of products, we can define an ABC policy for the pallet storage in a warehouse. Such a policy categorizes products in three groups, A, B and C. Products in category A are called fast moving products and products in category C are slow movers.

- A-items are goods which annual consumption value is the highest. The top 70-80% of the annual production typically accounts for only 10-20% of the total products.
- B-items are the interclass items, with a medium consumption value. Those 15-25% of annual production typically accounts for 30% of the total products.
- C-items are, on the contrary, items with the lowest consumption value. The lower 5% of the annual production typically accounts for 50% of the total products.

The ABC classification of Obedos is shown in figure 4. Class A items exist out of 20% of the products which cover 64% of the weekly infeed. Items in the B-class cover almost 25% of the infeed and contains 30% of the items. The rest of the products (50%) the remaining infeed (almost 13%).

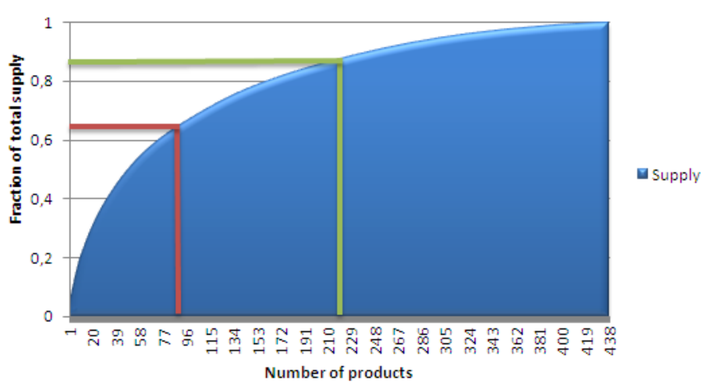


Figure 4: ABC Supply Scheme

For test purposes we will only consider the 163 products most commonly produced. This product group represents 80% of the total weekly storage volume. The rest of the products are not taken in consideration, because only a few pallets per week are requested. Including them in the model would make the model very large. These products have little turnaround, so they could be stored according to a classic ABC storage classification model and do not need to follow the single-product lane restriction if necessary as they have so little demand. To reduce the size of the storage allocation problem, we will only consider variable supply in the first period. This is also acceptable when the model is used in practice.

8.1.3 Initial Inventory

The initial inventory in the warehouse per product is known for the Obedos case. To incorporate this information in the model we have to determine a class in which the inventory is stored initially. To do this we apply a simple uniform distribution (random number) to allocate a class to the inventory. When doing this we assume that all pallets of the same product are stored in the same class.

Class	1	2	3	4	5	6	7	8	9	10	11	12
Nr of Pallets	135	107	92	169	127	154	149	155	136	88	231	167
Nr of Lanes	27	25	20	32	25	29	30	29	28	19	44	35

Table 4: Example Initial Inventory Obedos Case

This procedure is repeated 10 times, so we create 10 different start inventory situations to test the robustness of the model with respect to the starting inventory. The starting inventory can be expressed in pallet or lane quantity. An example of this allocation is shown in table 4. Figure 5 shows the minimum and maximum number of pallets of initial inventory per class over 10 different simulated start inventories.

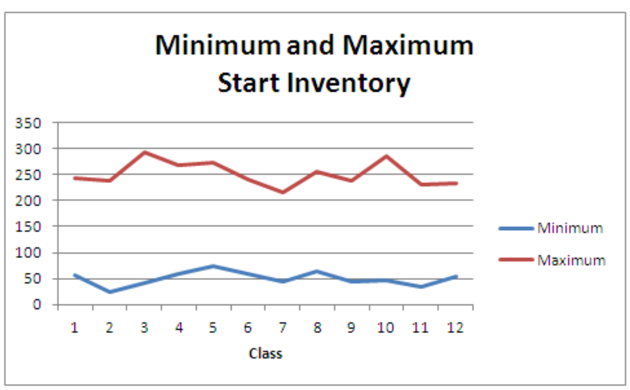


Figure 5: The min and max number of pallets in the start inventory for each class

If a model can find a feasible or optimal solution is dependent on the way the MIP solution tree is searched. Therefore it is important to test the different models and cases with multiple start inventory instances.

8.1.4 Capacity

The capacity used in this simulation is divided in two groups. For the first class on every floor in the warehouse a capacity of 52 lanes is defined. For the classes in the back of the warehouse a capacity of 100 lanes is used. The lane depth is set as 5 pallets deep. This means that the classes in the front of the warehouse have a capacity of 312 pallets and the classes in the back of the warehouse have capacity 600 pallets.

8.2 Case Specification

Data of five days of supply and demand is available. The simulation will have a time horizon of five periods, where each period represents one day. Every simulation is terminated if the gap between the lower bound and the best solution is smaller than a certain percentage. From preliminary experiments the value 5% appeared to be a good choice for the Obedos case.

8.2.1 Case 1: Stochastic Supply

In the first case a situation is simulated where supply is not deterministic. For every product an extra supply of two pallets is incorporated. These pallets are allocated to a position in the warehouse, but not included in the inventory of the next periods. Only stochastic demand is considered in the first period, because it has no influence on the operational execution of the upcoming period to incorporate the stochastic supply of the periods beyond the first period.

8.2.2 Case 2: Increased Deterministic Supply

In the second case the supply is defined deterministic, but we now incorporate the extra potential supply by adding the z_k values to the expected supply $Supply_i^t$. If we define the original supply variable as $^*Supply_i^t$, the new input variable supply now becomes $Supply_i^t = ^*Supply_i^t + z_k$ if $t = 1$. If $t > 1$ the supply does not change. The disadvantage of this method is that the extra supply incorporated stays present in the inventory of the warehouse for all periods. So when this new supply is not completely met, it still has an effect on the allocation of pallets in the warehouse. This effect is probably not very large, due to the fact that every period the simulation can be performed with an accurate inventory for the beginning of period 1. The advantage however, is that the model is smaller than in case 1. Especially the number of constrains is reduced significantly.

Case 1 and 2 are described below in terms of variable and constraint size for the first two models. The difference in size is not very large, due to the fact that for many variables the value is always equal to zero. For that reason these variables are not considered in the problem formulation. For model type 3 and 4, the model size will increase by the same absolute number of variables and constraints as these variables are not dependent on the index k.

	Model	Nr of Variables	Nr of Integer Variables	Nr of Constraints
Case 1	1,2	57187	23328	37717
Case 2	1,2	52489	21384	34741

Table 5: Cases Obedos

The objective function is the same in both cases. The penalty for every pallet that exceeds the average activity in a time period is equal to 1. This means that the only difference between the cases is the fact that in case 1 supply is split in 2 categories and in case 2 there is only 1 category. Case 1 does not consider the extra potential infeed expressed in values of z. Therefore the difference in the objective value is created by two causes:

1. The extra expected supply is not considered in the inventory of the next period (period 2).

2. The extra expected supply is not combined with the normal supply, therefore both types of supply will not share the same lane in the warehouse. So even if there is still enough rest inventory, the extra expected supply will have a lane allocated separately.

8.3 Results Obedos Case 1

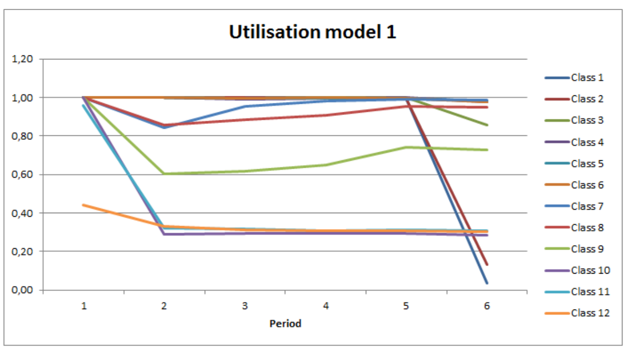
In the first Obedos case, stochastic supply is assumed and anticipated on. The average performance values of the different models are given in table 6. A more detailed representation of the results is given in the appendix. When solving the models, there are 3 different possible termination procedures. First, we can have a gap smaller than 5% between the best found solution and the best lower bound. Second, it can happen that the solver terminates the procedure when no feasible solution has been found. Last, a user interrupt has taken place, due to a long computational time and a small marginal improvement on the gap. We define a solved instance if a feasible solution is found, even if the gap is larger than 5%.

Tabel 6 shows the results of the different model types, using stochastic supply. Each model has been solved with 10 different starting inventories. The tabel shows the number of instances where a feasible solution was found, the average gap for which these solutions were found, the average solving time and the average number of iterations. The tabel also shows the minimum and maximum gap and solving time for the 10 different starting inventories.

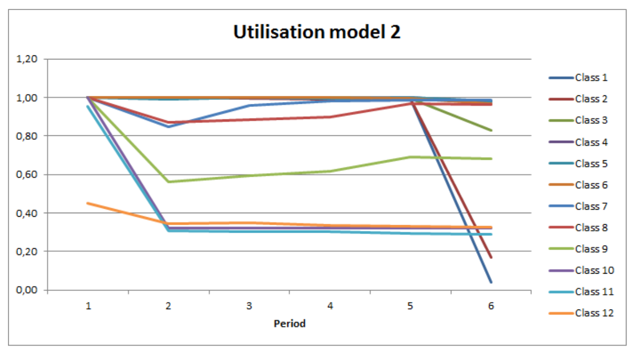
	Nr Solved instances	Obj.	Gap	Min gap	Max gap	Solving Time (s)	Min Time	Max Time	Number of Iterations
Model 1	10	120530	1,96	1,1	4,47	610	303	942	329998
Model 2	8	113683	1,95	1,2	5,26	673	310	3216	340774
Model 3	9	119594	6,23	2,22	17,33	851	92	3553	442989
Model 4	10	116836	3,39	2,02	5,26	770	235	1459	385867

Table 6: Obedos Results with stochastic supply, average values

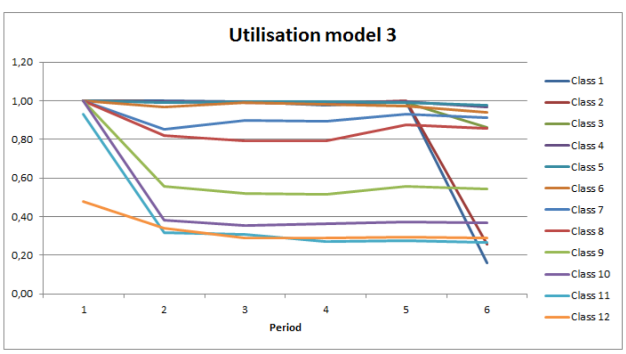
In table 6 we can see that there are two instances of model 2 where no solution is found, and one instance of model 3. The average gap of model 3 is higher than 5%, due to the fact that 2 instances have been terminated by a user interruption. This happened when the solutions were at a gap of 16,3% and 17,2% and explains the high average gap value.



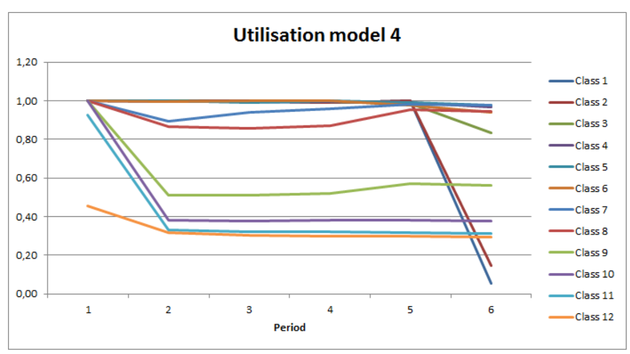
(a) Model 1



(b) Model 2



(c) Model 3



(d) Model 4

Figure 6: Average utilisation for case 1

It is remarkable to see that model 4 performs the best. As the objective value is small, it would be rather difficult for the solver to find a solution with a small gap between the solution and the lower bound. Apparently this has not been the case. We can also see that model four has found a solution with a gap smaller than 5% for all instances, this opposed to model 2 and 3.

The utilization of the four different models is stated in figure 6. We can see that the classes with the lowest costs have the highest utilization in all four models, which is expected. For model 1-3, the utilization of classes 1 and 2 drop significantly in period 6. This happens, because it is assumed retrieval takes place after allocation of pallets. So the desired situation is that all pallets are retrieved in the cheapest class in the last period, because these have the smallest travel times. Therefore we can say that the results show the expected scenario.

When we compare the different utilization charts, we see that there is not much difference between the models. We do see that from model 1 to model 4, the utilization is somewhat more stable over the time periods. However, this is not a very large change.

8.4 Results Obedos Case 2

In the second case we increase the deterministic supply and include these values in the inventory of the next periods. We expect the total utility of the first period to be smaller than in the first case. This is because less lanes are reserved for infeed, as the infeed lanes are filled up first, instead of reserving a separate lane for the extra supply.

	Nr Solved instances	Obj.	Gap	Min gap	Max gap	Solving Time (s)	Min Time	Max Time	Number of Iterations
Model 1	10	134464	10,12	2,90	18,07	2077	369	3350	1002422
Model 2	10	124464	7,86	2,01	21,97	1303	103	2912	639475
Model 3	10	120045	4,41	1,81	18,12	1109	225	2852	545723
Model 4	10	136531	7,13	1,90	16,47	1458	502	3507	715299

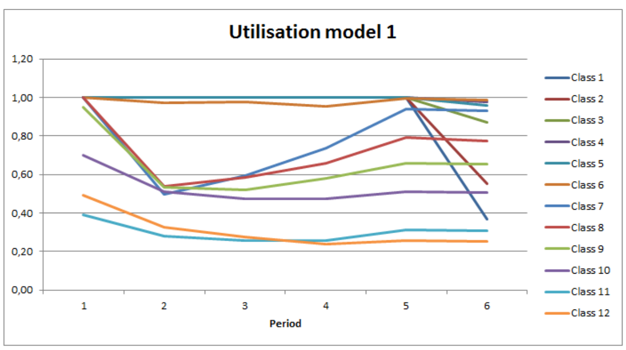
Table 7: Obedos Results with deterministic supply, average values

When we compare the results shown in table 6 and 7, we see that the model based on stochastic supply is not always able to result in a feasible solution. The second and third model have 1 and 2 instances respectively where the model types are not able to find a feasible solution for the given start inventories. As the models which use deterministic demand have less unfeasible solutions, the solutions that are found have a higher gap with the lower bound on average. The cause of this lies in the fact that there are 10 instances where the marginal improvement is very little and the solving procedure is terminated before a gap of 5% is reached.

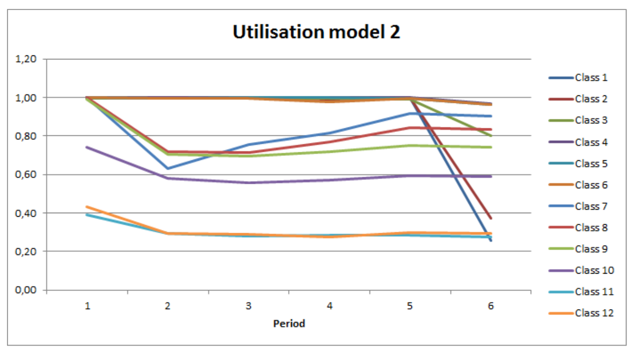
It is interesting to note that if an optimal solution is found, the largest computation time is 1459 seconds with an exception of 1 case. When no optimal solution is found, the minimal computation time is 2718 seconds. These numbers indicate that a large computation time can be set to be a termination condition.

We can see in table 7 that the first model has the most difficulty to find an optimal solution using deterministic supply. In 50% of the instances a solution with a gap larger than 5% is found. However, model 2-4 give a total of 4 suboptimal feasible solutions out of the 30 instances.

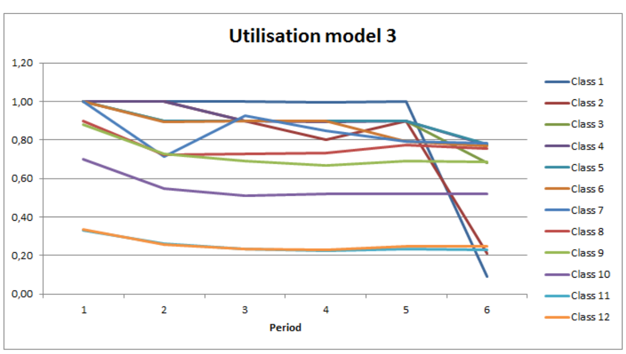
When we look at the course of the utilization values in figure 7a - 7d, we see that the changes within the model types have a larger effect on the utilization than in the previous case. When we look at the utilization of the first model, we see that after the first period, the utilization decreases by 40% for class 6 and higher. This behavior can be caused by the increase of supply in these types of models. This can cause a large decrease in the second period. After period 1, the utilization values increase over time. We see that every addition to the models smooth out the utilization over the periods, where there is almost no variation of the utilization in each class in the fourth model.



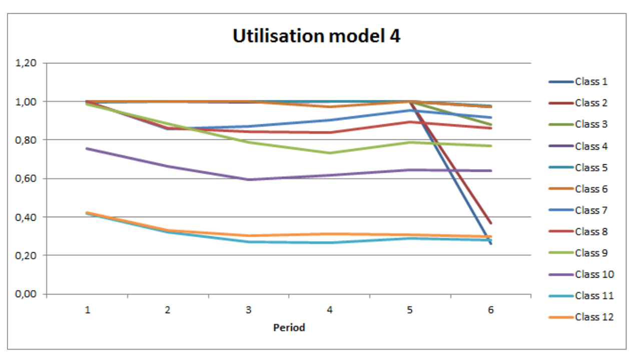
(a) Model 1



(b) Model 2



(c) Model 3



(d) Model 4

Figure 7: Average utilization for case 2

We have now performed an overall analysis of the different models and cases. In the next section we will analyze the impact of the different model types. We will look at one start inventory instance when comparing the different model types.

8.5 Detailed analysis on the different model types

To see what effect the different versions of the objective function described in section 6.3 have, we will devote this section to look at the different models in more detail. This analysis will be based on the start inventory set of instance 1 where stochastic supply is used.

8.5.1 Smoothing of periodic activity

When we add the penalty when the activity in a period exceeds the average activity, we observe that this creates a shift of 0,88% in the allocation of pallets over time. The highest allocation and retrieval activity takes place in the first period. We can explain this by looking at the number of early retrieval actions in this period. Pallets can be retrieved in an earlier time period as they are required for outfeed. This property is described in equation (36) and can cause early retrieval actions. However this can cause activity smoothing over the time periods, we see that when the penalty of timecrossing is not high enough, the penalty costs of timecrossing do not exceed the benefits of early retrieval. In practice this might not be the desired outcome.

To prevent this effect the penalty costs for timecrossing can be increased, which forces the model to reduce early retrieval activity in the first period. When the penalty costs are doubled, we observe a reduction in early retrievals of 0.93%. Remarkable is that when penalty costs are increased tenfold, the number of early retrieved pallets increase by 10%. This may not be the desired outcome, so we see that tuning of the penalty costs is very important.

8.5.2 Smoothing of level activity

When creating smoothing of activity over levels, we have assumed that only smoothing of the more expensive classes is wanted. In the Obedos case, this means only classes 7-12 have to deal with CrossClass penalties. When testing the influence of the CrossClass penalties, we will compare the activities in the considered classes of model 2 and 3. The results are shown in figure 8 - 10. Only the first period exceeds the CrossClass average and this can be explained by the assumption made on the start inventory. Activity is forced in the first period, as it is assumed all pallets of a product are stored in the same class when the start inventory is determined. In practice, this will not be the case and a representative image is the activity in periods 2-5, where inventory is not forced in the more expensive classes.

However, we can still see that the penalty values influence the activity on an overall level, as the total activity in these classes are reduced especially in the first period. Further, we can see that an increase of the penalty values has a positive effect on the smoothing between classes.

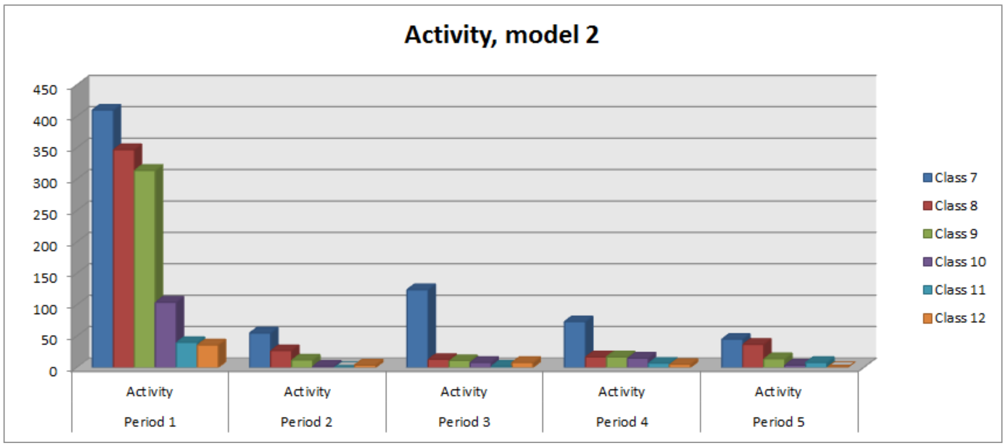


Figure 8: Activity model 2, class 7-12

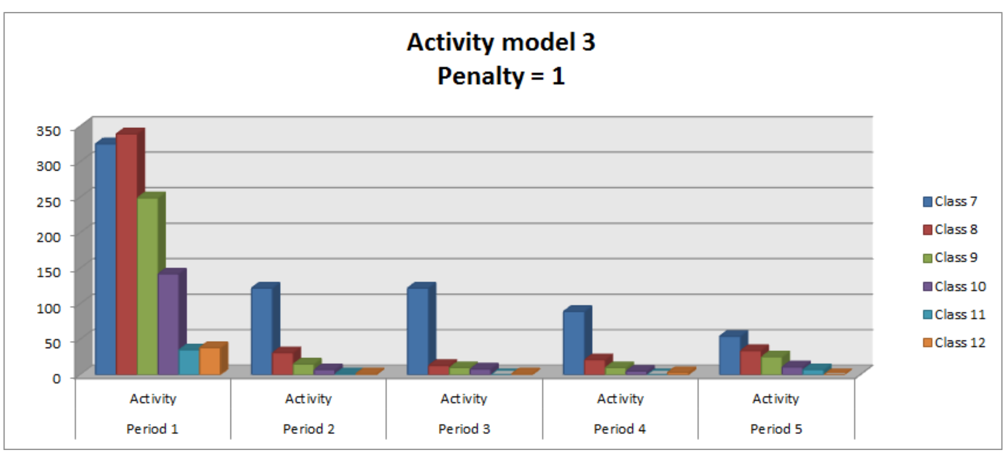


Figure 9: Activity model 3, Penalty = 1, class 7-12

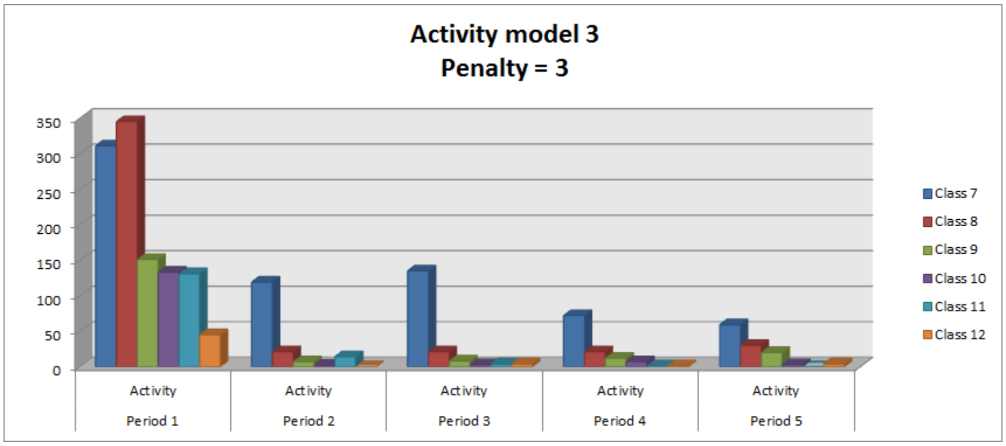


Figure 10: Activity model 2, Penalty = 3, class 7-12

8.5.3 Including turnover rate

To influence the placement of pallets that are not retrieved within the time horizon, we will include the turnover rate of each product in the objective function. The desired effect is that products with a high turnover rate are placed in a less expensive class than a low turnover rate product. We can measure this by taking the average class the pallets are allocated at infeed. This number can tell us if high turnover pallets are actually placed more in front of the warehouse, as we wish to accomplish. To create a more representative result, we only consider the infeed allocation over all periods.

In figures 11 - 13, the average class the pallets are placed are set out per product. The products are sorted on turnover, where product 1 has the highest turnover and product 162 has the lowest. We have determined the trend and the coefficient of determination, R^2.

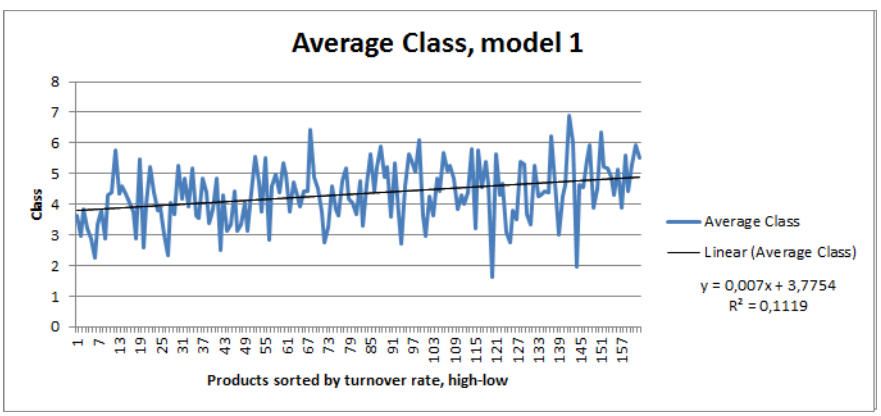


Figure 11: Average Class Allocation model 1

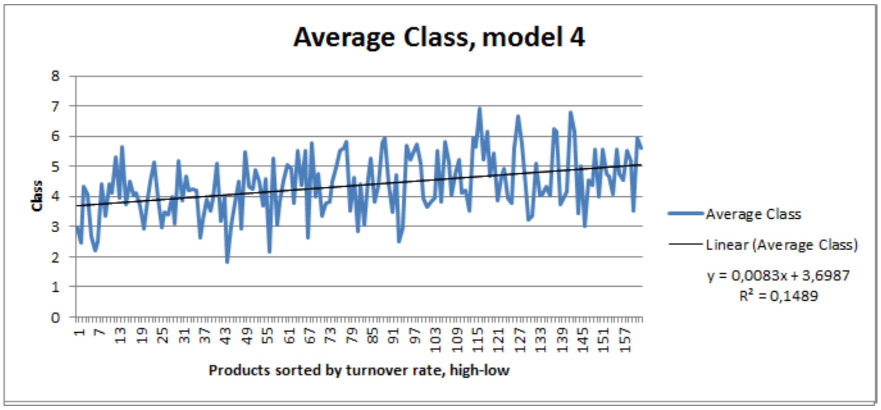


Figure 12: Average Class Allocation model 4, Penalty = 1

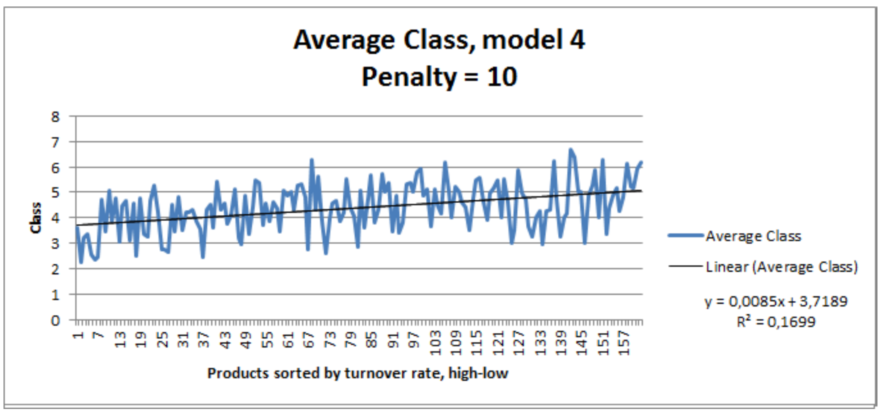


Figure 13: Average Class Allocation model 4, Penalty = 10

We see that the trend increases slightly when we incorporate the turnover rate, from 0,007 to 0,0085. However, this is not a significant increase and we can therefore conclude that the addition of the usage of turnover rates does not have the desired effect. Including the turnover rate in the model does have an effect on the R^2-value. When the penalty increases, the error decreases. This is the desired result.

8.6 Bound Convergence

When solving a mixed integer problem (MIP), the optimal solution is found by using a search tree. The path of this procedure is shown in figure 14 and figure 15. The path of the lower bound shown in these figures is typical behavior for these models. Both cases have a similar order of execution time and both cases seem to have one point in the process where the decrease stagnates. Nonetheless, case 1 appears to converge faster to the lower bound than case 2. So in the case a time constraint on the execution time is applied, the model with stochastic supply would have a higher probability to give a solution with a smaller gap than the model using deterministic supply.

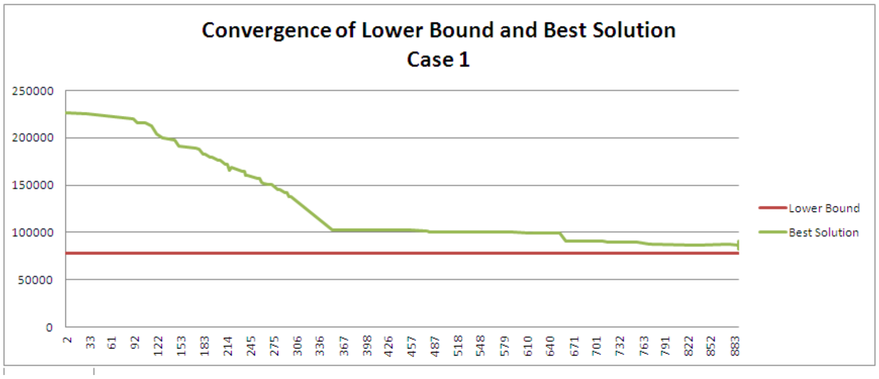


Figure 14: Convergence of bounds model 1, case 1

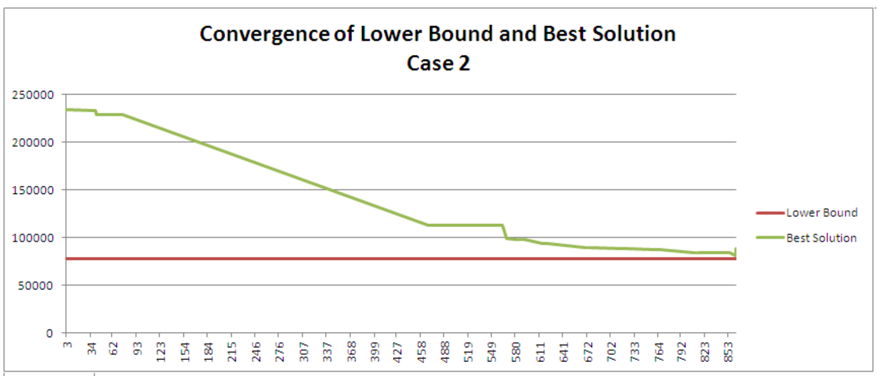


Figure 15: Convergence of bounds model 1, case 2

8.7 Model evaluation

Now that we have applied the model to a practical case, we can evaluate the results. We have split the models up in two cases: stochastic and deterministic supply. In the first case, where supply is stochastic, we use factors to incorporate the uncertain supply. In the second case we add the expected uncertain supply to the normal expected supply.

We have seen that the model based on stochastic supply is not always able to result in a feasible solution. However the model which uses deterministic supply has found a feasible solution in all 40 instances, but it takes much longer on average to find a solution with a gap smaller than 5%. One way to avoid this property is to use a termination criterion based on solving time and marginal improvement of the objective function.

When we look at the influence of the different model types, we see that the additions made to the objective function have more effect in the cases with deterministic supply. This can be explained by the fact that the models which use stochastic supply have reserved an excessive number of lanes. This creates robustness, but also inefficient use of the warehouse.

There are two main causes which lead to the differences in results between the two cases. First, the extra uncertain supply is not considered in the inventory of the next period in the first case. Second, this extra supply is also not combined with the normal expected supply. This means that in the first case more lanes are reserved than may be necessary. This leads to inefficient use of the warehouse, especially for products with a low supply.

9 Discussion

This section discusses some of the choices and assumptions that are made in this study. The purpose of this section is to investigate which assumptions are made and whether the proposed models are applicable to other cases as well.

9.1 Assumptions

First we will discuss some assumptions that are made in this study.

During the allocation of pallets, we assume single-product lanes. This means that only pallets with the same product type are stored in the same lane. In practice, this causes lanes to not be filled up to capacity, which can lead to inefficient use of the warehouse. As the models presented are used for a tactical purpose, the operational software can correct for this assumption. Pallets which are expected to leave the warehouse earlier than other pallets can be stored in front of each other without the need to excavate at the moment of retrieval. Another consequence of this assumption is that the overall capacity of the warehouse can be exceeded. Due to the inefficient use of space, especially when the factor-based models are used, this can lead to infeasible instances.

When the allocation of pallets is determined in the model, it assumes that all lanes within a class are exchangeable. In other words, it assigns equal travel times/costs to each lane in the class. In practice this assumption is acceptable when the sizes of the classes are relative small. If not, the travel distance within a class can differ very much, which makes this a nonrealistic assumption.

By allowing the possibility of early retrieval in constraint (36), we assume that the model is applied on a forward and reserve warehouse. The retrieved pallets can then temporarily be stored in the forward section of the warehouse. If this is not the case, this constraint should be strict like stated in constraint (21). This will limit the solution space of the models and can lead to longer solution times.

One other characteristic of the warehouses these models can be applied on, are the property of fixed and equal lane depths. We have assumed all lanes within the warehouse have equal depth. In many warehouses this does not have to be the case. It could be that the depth of the lanes variate in different areas of the warehouse and even the lane depth does not have to be fixed over time. For instance at the Obedos warehouse, the lanes in the middle of the warehouse can be entered from both sides. In the case studies we have separated these lanes by using a fixed lane depth and both sides of this lane are even assigned to different classes. If the lanedepth is variable, it means the only restriction on these lanes is the total length of the lane.

The basis of all models presented in this study is the availability of supply and demand information. If this information is not available of incorrect, this will cause the model to perform badly with respect to the real demand and supply. For the application of all model types presented in this study it is important that the data is available and corresponds with reality.

9.2 Penalty weights

The additions made to the basic model, introduced in section 6, have corresponding penalty weights. The values of these weights will determine the behavior of the pallet allocation model. This can be an advantage for the user, as he can increase the penalty on the aspects he finds of importance. In this study, the assignments of penalties to the CrossLevel and CrossTime variable have been set to different values. It has shown that an increase of the penalty value has the desired effect, see section 8.5.1. So when the models which include penalty costs are applied in practice, it is important to thoroughly test the models before setting a fixed penalty value.

9.3 Generality of the model

All conclusions made in section 8 are based on the Obedos warehouse and data. However, the models used can easily be altered to warehouses with a different lay-out or product range. The lay-out can simply be altered to the desired form by changing the lanedepth and the capacity of the classes. The models could even be applied on single-unit storage facilities, for which the lanedepth would be set to 1. These attributes make the models widely applicable.

10 Conclusion

The allocation of pallets while minimizing travel in a unit-load warehouse is a complex and nontrivial problem. The problem becomes more complex by the fact that supply and demand are determined by sources outside the warehouse, like production plants. It is therefore challenging to find an efficient storage-retrieval policy for the warehouse.

In this study we have taken a closer look at the storage and retrieval model used by Ang et al. (2012). We have ellaborated on the factor based storage and allocation model which can handle uncertain supply and variable demand. This model is compared to the storage-retrieval model based on deterministic supply.

The results of both models show some essential differences in outcome. One important aspect is the solvability of the models. In the first case, there are some instances which could not be solved, which limits the practical usage of the model. If a feasible solution was found in the first case, in almost all cases a solution with a gap smaller than 5% was found in reasonable time.

For every model which can handle deterministic supply, we have found an feasible solution within a few seconds (gap is less than 50%). We can also determine a maximal solving time, for a solution within a 5% gap per model. So a computation time based termination criteria can be constructed based on test cases. There is a window of computation time, where we can terminate the procedure knowing that no better solution will be found soon. This causes a tradeoff between the hope for a better solution, and saving of computation time.

The effect of the additions of the different models can be tuned by adjusting the associated penalty costs. The more developed the model, the less clear the effects of increased penalty costs get. A higher penalty cost can have an undesired effect as can be seen by the increased penalty of timecrossing.

10.1 Business recommendation

After these results, we can conclude that the models where deterministic supply is used are recommended to Ortec. It is expected that this model will perform better than a rule based heuristic used to allocate pallets. It has also shown in previous research that this model shows better results than the classic ABC-storage model. These types of models have shown to always be able to find a feasible solution. In practice this is a desired attribute for a model to have, even if the solution found is not optimal. It depends on the characteristics of the products and the design of the warehouse which specific model type is recommended.

As a tactical allocation tool it is also important to have a supporting operational system. Some assumptions made, like single product lanes, can be ignored by the operational software of the warehouse. This way, the warehouse can be used even more efficient than the allocation model shows.

References

[1] Lim Y.F. Sim M. Ang, M. Robust storage assignment in unit-load warehouses. *Management Science*, 58(11):2114–2130, 2012.

[2] White J.A. Bozer, A.Y. Travel-time models for automated storage/retrieval systems. *IIE Trans*, 16(4):329–338, 1984.

[3] Schwarze S. Vo S. Caserta, M. A mathematical formulation and complexity considerations for the blocks relocation problem. *European Journal Of Operational Research*, 219(1):96–104, 2012.

[4] Vo S. Sniedovich M. Caserta, M. Applying the corridor method to a blocks relocation problem. *OR Spectrum*, 33(4):915–929, 2009.

[5] Gunn A.G. Cormier, G. A review of warehouse models. *European Journal of Operational Research*, 58:3–13, 1992.

[6] Leduc T. Roodbergen K. De koster, R. Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, 182(2):481–501, 2007.

[7] Van Der Poort E. De Koster, R. Routing orderpickers in a warehouse: a comparison between optimal and heuristic solutions. *IIE Transactions*, 30(5):469–480, 1998.

[8] P.J. Egbelu. Framework for dynamic positioning of storage/retrieval machines in an automated/retrieval system. *International Journal of Production Research*, 29:17–37, 1991.

[9] Rosenblatt M.J. Eynan, A. Establishing zones in single-commandclass-basedrectangular as/rs. *IIE Transactions*, 26(1):38–46, 1994.

[10] Ratldff H.D. Goetschalckx, M. Optimal lane depths for single and multiple products in block stacking storage systems. *IIE Transactions*, 23(3):245–258, 1991.

[11] Goetschalckx M. Gu, J. and L. M. McGinnis. Solving the forward-reserve allocation problem in warehouse order picking systems. *Journal of the Operational Research*, 61(6):1013–1021, 2010.

[12] Goetschalckx M. McGinnis L.F. Gu, J. Research on warehouse operation: A comprehensive review. *European Journal Of Operational Research*, 177(1):1–21, 2007.

[13] Lu Z.L.Z. Xi L.X.L. Han, X.H.X. A proactive approach for simultaneous berth and quay crane scheduling problem with stochastic handling time. *International Conference on Computers Industrial Engineering*, 207(3):1327–1340, 2009.

[14] Schwarz L.B. Graves S.C. Hausman, W.H. Optimal storage assignment in automatic warehousing systems. *Management Science*, 22(6):629–638, 1976.

[15] Kim S.W. Kim K.H. Jang, D. The optimization of mixed block stacking requiring relocations. *Int. J Production Economics*, 2012.

[16] Ramirez R.G. Kanet, J.J. Optimal stock picking decisions in automatic storage characteristics machines and retrieval. *OMEGA Int. J. Mgmt. Sci.*, 14(3):239–244, 1986.

[17] Hong G.P. Kim, K. H. A heuristic rule for relocating blocks. *Computers Operations Research*, 33(4):940–954, 2006.

[18] Choe R. Kim Y.H. Ryu K.R. Park, T. Dynamic adjustment of container stacking policy in an automated container terminal. *International Journal of Production Economics*, 133(1):385–392, 2011.

[19] Eynan A. Rosenblatt, M.J. Deriving the optimal boundaries for class-basedautomatedstorage/retrieval systems. *Management Science*, 35(12):1519–24, 1989.

[20] Babu P.S. Sarker, B.R. Travel time models in automated storage/retrieva systemsl: A critical review. *Int. J Production Economics*, 40:173–184, 1995.

[21] Cardin O. Castagna P. Sari Z. Meghelli N. Taylor, P. Performance evaluation of in-deep class storage for flow-rack as / rs. *International Journal of Production Research*, 50(23):6775–6791, 2013.

[22] Sharp G.P.G.A.J.R.N. Pochet Y. Van den Berg, J.P. Forwardreserve allocation in a warehouse with unit-load replenishments. *European Journal of Operational Research*, 111:98113, 1998.

[23] Kim K.H. yang, J. H. A grouped storage method for minimizing relocations in block stacking systems. *Journal of Intelligent Manufacturing*, 17(4):453–463, 2006.

[24] Sharp G.P. Yoon, C.S. A structured procedure for analysis and design of order pick systems. *IIE Transactions*, 28:379–389, 1996.

[25] De Koster R.B.M. Yu, Y. Sequencing heuristics for storing and retrieving unit loads in 3d compact automated warehousing systems. *IIE Transactions*, 44(2):69–87, 2012.

[26] de Koster R.B.M. Yu, Y. On the suboptimality of full turnover-based storage. *International Journal of Production Research*, 51(6):1635–1647, 2013.

[27] Liu J. Wan Y. Murty K.G. Linn R.J. Zhang, C. Storage space allocation in container terminals. *Transportation research*, 37:883–903, 2003.

11 Appendix

11.1 Result tables

The following tables show the results of the Obedos case with all model types described in section 6.3.

	Objective value	Gap	Solving Time	Iterations	Notes
1	118542	1,1	357	171548	
2	120362	1,45	452	249196	
3	118802	1,16	303	157558	
4	121349	1,72	499	280381	
5	123966	4,47	942	527070	
6	122027	2,56	788	413211	
7	120776	1,75	556	291317	
8	121011	2,63	879	488626	
9	118691	1,26	916	491270	
10	119777	1,53	408	229804	
Average	120530	1,96	610	329998	

Table 8: Model 1 with stochastic supply

	Objective value	Gap	Solving Time	Iterations	Notes
1	111710	1,2	373	187820	
2	113489	1,4	508	266682	
3	LB: 110579		3216	1107893	solver termination
4	114688	1,81	1192	570212	
5	LB: 111621		2371	825910	solver termination
6	118405	5,26	1216	585525	
7	113614	1,44	498	267262	
8	112389	1,22	310	159738	
9	112131	1,64	861	447002	
10	113034	1,69	427	241953	
Average	113683	1,95	673	340774	

Table 9: Model 2 with stochastic supply

Zonder k en alleen storage and retrieval costs

	Objective value	Gap	Solving Time	Iterations	Notes
1	112974	2,22	92	57370	
2	116133	3,23	222	137210	
3	134201	17,33	3553	1830951	User Interrupt
4	119387	5,29	934	454505	
5	114106	1,77	202	109648	
6	LB: 112743		2799	1087525	Solver termination
7	134225	16,16	1950	1013938	User Interrupt
8	115487	3,46	186	111190	
9	114669	3,48	261	129410	
10	115168	3,14	256	142680	
Average	119594	6,23	851	442989	

Table 10: Model 3 with stochastic supply

	Objective value	Gap	Solving Time	Iterations	Notes
1	114228	2,02	949	470455	
2	116640	3,86	1166	611558	
3	114462	2,01	1459	748878	
4	116829	2,16	818	396271	
5	119590	5,26	674	361678	
6	118768	4,12	235	137090	
7	117223	3,02	1253	584031	
8	117281	3,94	560	235853	
9	115651	3,29	238	139436	
10	117690	4,21	347	173420	
Average	116836	3,39	770	385867	

Table 11: Model 4 with stochastic supply

	Objective value	Gap	Solving Time	Iterations	Notes
1	122720	2,9	369	173971	
2	145301	16,9	2718	1356320	User Interrupt
3	125884	4,99	2104	1057222	
4	127436	4,78	1036	499207	
5	126826	4,98	896	473536	
6	142267	14,83	3119	1481921	User Interrupt
7	122728	1,48	1031	484190	
8	145911	17,93	3172	1668004	User Interrupt
9	139180	14,34	3350	1452059	User Interrupt
10	146386	18,07	2970	1377792	User Interrupt
Average	134464	10,12	2077	1002422	

Table 12: Model 1 with deterministic supply

	Objective value	Gap	Solving Time	Iterations	Notes
1	116293	3,17	504	240082	
2	116504	2,12	103	62630	
3	116649	3,07	787	409836	
4	119397	3,73	687	353939	
5	145982	21,97	2912	1441835	User Interrupt
6	138779	17,31	2589	1277219	
7	143852	20,4	2082	1045773	
8	116307	2,61	209	111866	
9	115149	2,21	253	126648	
10	115725	2,01	2905	1324924	
Average	124464	7,86	1303	639475	

Table 13: Model 2 with deterministic supply

	Objective value	Gap	Solving Time	Iterations	Notes
1	115278	1,81	234	126782	
2	117775	2,41	836	429241	
3	115768	1,84	558	266816	
4	122013	5,19	832	407211	
5	117525	2,42	2137	968211	
6	141081	18,12	2852	1480216	User Interrupt
7	118861	3,08	1253	574502	
8	116483	2,18	225	125268	
9	119485	5,24	1058	517687	
10	116184	1,85	1100	561299	
Average	120045	4,41	1109	545723	

Table 14: Model 3 with deterministic supply

	Objective value	Gap	Solving Time	Iterations	
1	127705	2,2	502	262777	
2	151601	16,47	2900	1521056	User interrupt
3	131488	4,69	813	389598	
4	156840	18,8	2624	1238000	
5	128722	1,83	546	262803	
6	149662	15,02	3507	1710221	User interrupt
7	133936	5,24	831	398412	
8	129865	3,27	997	512046	
9	127316	1,9	1302	570666	
10	128172	1,92	557	287406	
Average	136531	7,13	1458	715299	

Table 15: Model 4 with deterministic supply