Master Thesis
Industrial Engineering and Management

Accuracy Improvement of Warehouse Capacity Calculation using the Bin Packing Problem

Yorick Beekman

First Supervisor: Breno Alves Beirigo
Second Supervisor: Dennis Prak
Company Supervisor: Logistics Specialist

November 10, 2023

University of Twente
Faculty: Behavioural, Management and Social Sciences
Specialization: Production and Logistics Management
Orientation: Supply Chain and Transportation Management

UNIVERSITY OF TWENTE.

Acknowledgements

Dear reader,

This document represents my master's thesis, symbolizing the end of my academic journey at the University of Twente. Prior to delving into the content, I wish to express my gratitude to numerous individuals whose invaluable assistance contributed to the development of this thesis and supported me throughout my five-year time at the university.

Foremost, I extend my appreciation to my family for their unwavering support. From my years in primary school to the finishing of my academic journey through this master's thesis, their encouragement and motivation have been essential, propelling me to maximize my efforts. Additionally, I extend my gratitude to my friends who have provided a comforting place in the midst of the challenges and demands of my academic journey. Our shared memories have been crucial in preserving my sanity (sort of) and rendered moments of joy that would have otherwise remained unrealized.

Moreover, I would like to thank the Warehouse Operations team of Company X for their warm welcome within the team from the first day on. In addition to this, I express my gratitude for their invaluable provision of essential insights and constructive feedback, as well as their consistent readiness to answer all my questions.

In addition, I extend my gratitude to my academic supervisors at the University of Twente, Breno and Dennis. Their invaluable feedback has been crucial in enhancing the quality of this report. Moreover, their guidance throughout the developmental process was of great significance.

Finally, I wish to express my appreciation to all individuals who have contributed to my research, in whatever way this may be. Their collective involvement has undoubtedly enriched this thesis.

I am really looking forward to the next steps I will take within the company.

Yorick Beekman
Utrecht
November 10, 2023

Management Summary

This research is undertaken at Company X, specifically, the shelving area within a single warehouse out of the company's eight facilities. The particular warehouse in question serves as a distribution centre for items within size groups ranging from 3XS to L. On average, the warehouse distributes approximately 1.35 million items per month.

Problem Context

The Warehouse Operations team have elucidated that the warehouse frequently experiences elevated fill rates and occupancy rates, thereby giving rise to some challenges. These challenges encompass additional operational costs resulting from inefficient put-away operators as they have difficulties finding an open storage location. In order to mitigate the consequences of these challenges, the Warehouse Operations team has introduced the Traffic Light Model. This model delineates a set of strategic measures to act upon once specific threshold values for the fill rate (the degree to which a storage location is filled) and occupancy rate (number of occupied locations, as a ratio to the total number of locations) are attained. Some examples of these measures are the refusal of deliveries to the warehouse or internal item relocations. Through the implementation of this Traffic Light Model, the Warehouse Operations team averted the occurrence of more severe scenarios.

As communicated by the Warehouse Operations team, the strategies outlined in the Traffic Light Model have been employed with a notable frequency. During the years 2021 and 2022, the company achieved the preferred green scenario only 30.3% of the time. Conversely, for the remaining duration, the company found itself situated in undesired scenarios denoted as orange, red, or even black. These scenarios signify that the warehouse surpassed the threshold values for the occupancy rate and fill rate that have been set in order to have efficient warehouse operations.

Root Cause

It swiftly became evident that the computation of the stockmax value was inaccurate. For clarification, the stockmax denotes the maximum quantity of items that the company seeks to maintain within the warehouse, this value is deliberately set lower than the warehouse's actual capacity, as the company accounts for a fill rate of 35% and an occupancy rate of 85%. This is done to maintain efficient warehouse operations. For example, with an occupancy rate that is almost 100%, it is hard to find an empty storage location. The put-away operator then has to search for an empty location, which takes more times, and thus is less efficient.

To determine the precise disparities between the calculated stockmax prediction and the actual stockmax, we examine the instances when the company transitioned from a green scenario to an orange scenario or conversely, from orange to green. We assume that the stock levels on these specific dates serve as indicators of the actual stockmax values.

The calculated stockmax of the Warehouse Operations team exceeded the actual stockmax by 876,641 items on average on the specified dates. This miscalculation ensured a misperception that the warehouse possessed a greater capacity for accommodating items than it genuinely did. Consequently, an excessive number of items were stored within the warehouse, thereby exceeding the threshold value for the fill rate or occupancy rate, necessitating recourse to the actions delineated in the Traffic Light Model. Thus, to minimize the duration spent in undesirable scenarios as outlined by the Traffic Light Model, we need to refine the calculation method for determining the stockmax value.

Literature Review
A review of the literature reveals a congruence between the problem under investigation and the widely recognized Bin Packing Problem. In the Bin Packing Problem, the goal is to assign a given set of items to a minimal number of bins. In our context, we have a given number of bins in the warehouse (accounting for an occupancy rate of 85%) with a given capacity (accounting for a fill rate of 35%), and we determine the maximum number of items that can be assigned to these bins. In light of this insight, we search for viable solutions to solve our problem. Due to the NP-hard nature of the problem at hand, we cannot achieve an optimal solution and thus need heuristics. From the literature, we select several viable Bin Packing Problem heuristics for the purpose of the assignment of items to bins.

Model for Company X
To evaluate the selected heuristics, a computer model has been constructed, customized to align with the specific operational requirements of Company X. Within this computer model, we initially replicate the current situation within the warehouse by employing a bin assignment strategy that mimics the current item to shelf allocation in the warehouse. This configuration serves as the initial state for the application of our heuristic algorithms, wherein we assign additional items to the bins until there is one item that cannot be added to any bin. The execution of these heuristics necessitates the specification of various parameters, including but not limited to sizemix, item volumes, fill rate, and occupancy rate.

In our experiments, we test a total of five heuristic approaches, both with and without the incorporation of the operators' put-away logic as observed in the warehouse. When implementing this logic, a specific shelf level is chosen with a certain probability, with the likelihood of selection dependent on the current allocation of size groups to distinct shelf levels. In practical terms, this implies that if 60% of a given size group is currently situated on shelf three in the real warehouse, then the probability of selecting shelf three as the destination for item assignment in the model also stands at 60%. In addition to the five heuristics, a randomized assignment approach has been included for assessment. Under the random assignment method, a shelf level is randomly selected, and subsequently, a random bin is selected for item addition.

Solution Choice
One hard constraint is imposed on the heuristics; the calculation procedure must be executed within one hour. Regrettably, two of the employed heuristics, namely, *Best Fit* and *Worst Fit*, substantially exceeded this temporal threshold, with computational times surpassing six hours. Consequently, these heuristics were omitted from the selection as the optimal solution. Conversely, the residual heuristics demonstrated computational times well within the stipulated one-hour timeframe.

In light of our experiments, consisting of 10 iterations, we extend our assessment to an examination of the standard deviations in the stockmax values. This analysis reveals that, on average, the standard deviations hover approximately between 2,000 and 4,000 items across the heuristics. To gain a better understanding of these standard deviation values, we introduce an additional layer of analysis through the application of the coefficient of variation (CoV), a metric expressing the standard deviation relative to the mean. It is widely acknowledged that a CoV value below 10% is considered indicative of exceptional performance. In our investigation, the highest CoV value observed among the heuristics was a mere 0.15%, significantly below the 10% threshold. Given the uniformly exceptional performance demonstrated by all heuristics in terms of standard deviation, this particular metric has been omitted from further consideration within our decision-making process.

To determine the best solution, we employ the Multi-Criteria Decision-Making methodology. This approach leads us to the conclusion that the *Next-3 Fit with Put-Away Logic* heuristic exhibits superior performance when compared to its counterparts. Notably, the mean discrepancy between the stockmax values computed by the computer model and the actual stockmax is only 4,570 items on average. Furthermore, the heuristic demonstrates an average absolute deviation of 154,797 items between the computed and actual stockmax values.

Comparison with Current Method

When comparing these findings with the current approach employed by the Warehouse Operations team, a significant improvement becomes evident. The Warehouse Operations team consistently exhibited an overestimation of the stockmax, with an average excess of 876,641 items. Consequently, the average absolute deviation aligns with this figure at 876,641 items as well. In both cases, it is unmistakably apparent that the computer model significantly outperforms the accuracy of the current procedures.

Conclusion

The computational model provides the Warehouse Operations team with the possibility to compute a more accurate stockmax value. With this improved stockmax estimate derived from the computer model, the Warehouse Operations team anticipates a decrease in the actions taken as described by the Traffic Light Model. This, in turn, enhances the operational efficiency within the warehouse and leads to cost reductions.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations and Jargon

BPP

Bin Packing Problem. A combinatorial optimization problem where heterogeneous-sized items need to be placed into a limited number of fixed-capacity bins, with the objective of minimizing the total number of bins required. 35, 41

Bunker

A fireproof storage location, dedicated to flammable and/or explosive items. 2

CMC

A machine that automatically packs mono-orders in boxes. 8

EAN

European Article Number. A standardized 13-digit barcode used for identifying products. 5

Fill Rate

The degree to which utilized storage locations are filled, as a ratio to the maximum storage volume of the storage location. 5, 9, 20, 22, 23, 29, 33, 48, 49, 51, 58, 69

KP

Knapsack Problem. A combinatorial optimization problem involving items which are characterized by a weight and a value, where the goal is to select items for inclusion in the knapsack such that the total value of the selected items is maximized while their combined weight does not exceed the limit of the knapsack. 38, 41

MILP

Mixed Integer Linear Programming. A mathematical optimization technique that deals with linear mathematical models containing both continuous and discrete decision variables, aiming to find the best solution while allowing some variables to take only integer values. 36

Mono-Order

Orders consisting of only one item. 7

Multi-Order

Orders consisting of more than one item. These can be different items, or more than one of the same item. 7

Multi-SKU

Number of unique SKUs in one storage location. Currently, at most two different SKUs can be in one location, known as multi-SKU2. 4

NP-hard

Non-deterministic Polynomial-time hard. A classification in computational complexity theory for problems at least as difficult to solve as the hardest problems in NP (Non-deterministic Polynomial-time). These problems do not necessarily have efficient algorithms for finding their solutions, and solving them may require exponential time. 37, 40, 41

Occupancy Rate

The number of storage locations occupied by at least one item, as a ratio to the total number of storage locations in the warehouse. 4, 5, 9, 20, 22, 23, 29, 33, 48, 49, 58, 69

Partner

A company or person that uses the platform and/or logistics services of Company X to sell items. 1, 16, 22, 29

Pick Batch

One pick batch consists of a group of customer orders, which are picked by one order-picker in one zone. A pick batch is limited to the size of one tote. Each tote is filled with a unique pick batch. 7

Picking Route

The route that an order-picker walks to collect all items from one pick batch. The WMS determines the route by calculating the shortest path. 7

SAM

Stock Allocation Model. Running this model, Capacity Steering determines which supplier delivers its products to which warehouse. Using this information, the size mix can be determined. 17, 25, 29

Size Mix

The percentages of items of a specific size group that is expected to be delivered to the warehouse in the coming periods. For the warehouse in this research, the percentages of size groups 3XS to L should add up to 100%. 16, 17, 20, 22, 25, 29, 47, 50

SKU

Stock Keeping Unit. A number assigned to products for inventory management. 4, 14

SmartMailer

A machine that automatically packs small items in an envelope. 9

Spur

Incoming or outgoing station at the zones in the warehouse. At the spur, totes arrive with items that need to be stored, or totes leave with items that have been picked. 4, 7

Stingray

Temporary storage tower. Totes with multi-orders will be stored in the Stingray until all totes containing the products for that multi-order have arrived. After that, the totes travel to the outbound lines at the same time. 7

Stockmax

The maximum number of items the company wants to have in the warehouse. This is lower than the true capacity of the warehouse as the company accounts for a specific threshold for the fill rate and occupancy rate in order to improve efficiency and decrease costs. 9, 16, 18, 22, 25, 48, 57, 69

T-Car

Station where operators place items from a pallet into a tote to be transported to the incoming spurs. 3

Tote

Big blue box to place items in. Used to transport items through the warehouse using the conveyor belt. 3

Warehouse Share

The percentage of items assigned to a specific warehouse. 27

WIP

Work In Progress. A location where items are stored that need to go to the T-Car stations. 3

WMS

Warehouse Management System. Keeps track of all placed orders, picked orders and inventory. 5, 6, 15, 17

Zone

A subset of the picking area in the warehouse. All items from one pick batch should be picked from a single zone. In total, there are 57 zones in the warehouse, divided over 5 fireproof bunkers and 52 regular storage zones. 2

1 Introduction

Optimal logistic performance holds significant importance within the operations of a company. The logistics network of the company in question entails the delivery of products from suppliers to a warehouse, which in turn facilitates distribution to customers. When this network operates efficiently, product availability is enhanced, leading to increased customer satisfaction and ultimately higher revenue, which is the overarching objective of the company in question. However, achieving this goal becomes challenging without a well-performing logistics network.

The primary focus of this paper is to enhance a specific segment of the company's logistics network, namely the shelving area within the warehouse. This particular area serves as the designated space where individual products are stored in storage locations. Section 1.1 presents an introductory overview, providing insight into the company. Section 1.2 elucidates the relevant logistics processes. This section is crucial for developing an understanding of the research environment. Section 1.3 systematically identifies the challenges and issues prevalent within the company, aiming to identify the core problem. Following that, Section 1.4 defines the research problem and delineates the scope of the study. Lastly, in Section 1.5, the problem-solving approach is delineated alongside the formulation of specific research questions.

1.1 An Introduction to Company X

Company X has experienced remarkable growth, transitioning from a modest entity operating out of a construction shed to emerging as the dominant market player in the Netherlands and Belgium. In the preceding year, Company X successfully served a vast customer base of more than 13 million individuals across the two countries. Notably, the company's expansive product portfolio has increased over the past few decades, consisting of a total of over 36 million items. Such substantial growth and diversification in products would have been unattainable without the contributions of tens of thousands of *partners* who leverage Company X's platform to market and distribute their products. Some achievements are visually represented in Figure 1, which highlights the significant milestones accompanying the success of Company X.

Company X operates from a network of eight warehouses, each characterized by its distinct operational attributes. Seven of these warehouses are dedicated to the distribution of items, while the remaining warehouse serves as a central facility for collecting returned items. Each warehouse is tailored to accommodate specific categories of products. For instance, one warehouse specializes in handling one particular item, while another focuses on medium-sized products. Additionally, two warehouses are specifically designed to manage extra-large items. This research centres its investigation on one of the warehouses involved in distributing items falling within the size range of 3XS to L. These could be any sort of items, from clothing to phones and from board games to monitors.

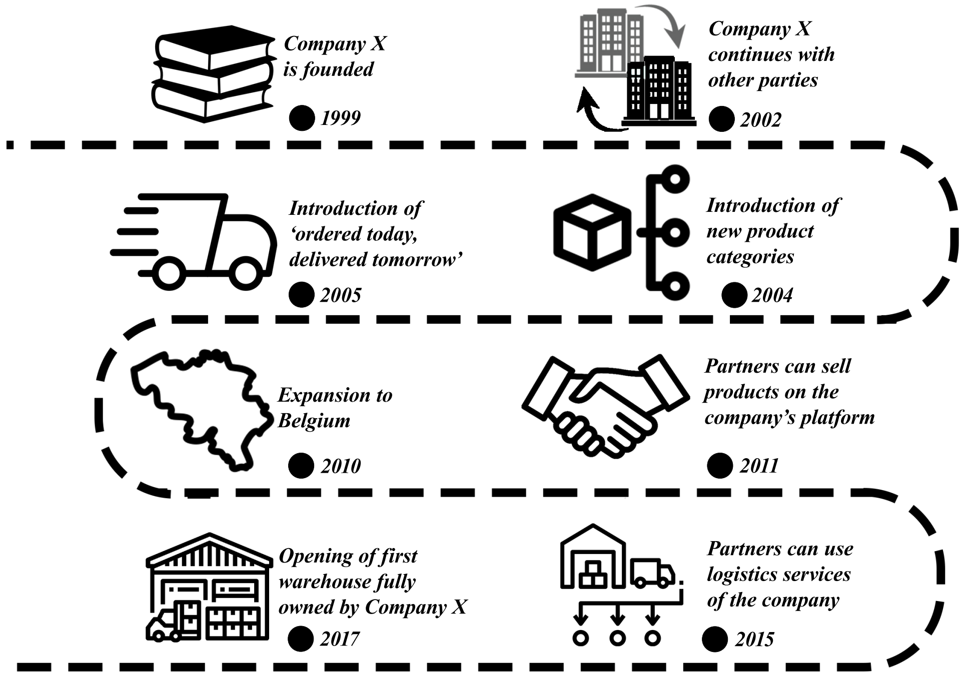


Figure 1: Timeline of Company X

1.2 An Introduction to the Logistics Processes

The warehouse under consideration is equipped with an extensive storage infrastructure, consisting of over 1.5 million storage locations and utilizing a total of 57 storage *zones*. Among these zones, five specialized areas, known as *bunkers*, are specifically designated for storing flammable items such as perfumes or deodorants. The remaining 52 zones function as standard storage locations for all other products. Each zone consists of 12 aisles, each approximately 40 meters in length. To facilitate efficient movement within the warehouse, cross aisles are strategically positioned at the half-way point of each aisle, allowing pickers and stockers to swiftly transition between aisles. Within each aisle, there are six shelves designated for product storage. It is assumed that storage locations on the same shelf possess identical storage volumes. Table 1 provides an overview of the storage volumes per storage location per shelf. Notably, shelf three is divided by distribution boxes, resulting in an increased number of storage locations. These distribution boxes are specifically designed to accommodate smaller items. Figure 2 offers a visual representation of a distribution box.

Figure 2: A distribution box, extracted from shelf level three

Table 1: Volume per storage location per shelf

Shelf	Volume (L)
5 (top shelf)	123
4	55
3	5
2	38
1	57
0 (bottom shelf)	151

In the initial two months of 2023, the warehouse achieved an average monthly distribution of 1.35 million items. Timely delivery of these items to customers necessitates the implementation of several logistics processes. To get a better understanding of these processes, a concise description of them is provided.

1.2.1 Inbound

The processes start with the unloading of items from the truck, a step carried out at the inbound section of the warehouse. At this stage, pallets containing the items are collected from the truck and subsequently enter the warehouse. Upon arrival, the pallets are directed to either the bulk storage area or the work in progress (*WIP*) area, depending on their characteristics. Only mono-pallets, pallets that exclusively consist of a single type of item, can be allocated to the bulk storage area if the items are not required in the shelving area within the upcoming seven days. Conversely, multi-pallets, comprising multiple distinct items, are always assigned to the WIP area.

When an item needs to be replenished, a pallet is transferred from the WIP area to a designated workstation located at one of the 12 *T-Car* stations (Figure 3). At these stations, an operator retrieves the items from the pallets and proceeds to place them inside a *tote*. The tote, a large blue bin, serves as a practical container for transporting items throughout the warehouse via the conveyor belt system. Its dimensions are measured at 75 x 55 x 41 centimetres. Notably, a 13$^{\text{th}}$ T-Car station is exclusively designated for handling items requiring special attention. Such items may include those without a designated code or items that have not yet been integrated into the system and need measurement procedures.

Figure 3: One of the 10 workstations at one of the 12 standard T-Car stations

During the process of placing items into totes at the T-Car station, it is recommended that operators distribute the same Stock Keeping Unit (*SKU*) across different totes. This strategic approach ensures that items are spread throughout the entire warehouse rather than being concentrated in specific areas. This is because totes are sent to different zones. By adhering to this practice, the company believes that the picking process becomes more efficient, as the items are accessible from multiple zones within the warehouse.

Once the operator has filled a tote with items, the system may prompt the operator to push the tote onto the conveyor belt when it has reached its capacity. This capacity can be determined by factors such as weight or volume, ensuring that the tote does not exceed its maximum limit (e.g., items should not protrude from the top of the tote). Alternatively, an operator has the option to manually select the *Full Tote* option if the system fails to recognize that the tote is full. Once placed on the conveyor belt, the tote begins its journey towards a specific zone within the warehouse. The determination of the destination zone for the tote is guided by an algorithm that considers three criteria in sequential order of importance:

1. If the number of totes at the incoming spur of a particular zone has reached its capacity, that zone is not considered as an option for the tote's placement.
2. From the remaining zones, the algorithm selects the zone with the least stock quantity of the products contained in the tote.
3. If multiple zones have the least stock of the items in the tote, the algorithm prioritizes the zone with the lowest *occupancy rate*.

Once the algorithm identifies a specific zone as the destination for the tote, the tote proceeds to arrive at the incoming *spur* of that designated zone. As depicted in Figure 4, the incoming spur represents a branch of the conveyor belt system where a stocker is stationed to retrieve the tote. The stocker's responsibility involves unloading the items from the tote and subsequently placing them onto the shelves within the assigned zone. This process ensures that the items are appropriately stored and made accessible for subsequent picking and order fulfilment operations within the warehouse.

Figure 4: Incoming spur with blue totes (front) and outgoing spur (back)

1.2.2 Stock

Regarding the put-away process, the operators are not bound by strict constraints regarding the placement of items, except for the *multi-SKU* constraint. Currently, the team adheres to the multi-SKU-2 constraint, allowing for a maximum of two distinct European Article Numbers (*EANs*) to be stored within a single storage location. In addition to this constraint, managers provide guidance to the operators, offering recommendations based on various factors. For instance, liquids are advised to be placed on the lowest shelf, heavy items should not be positioned at excessive heights, and operators are encouraged to utilize the smallest available storage location whenever possible. However, it should be noted that these recommendations are merely advisory, and there are no penalties imposed when items are stored in locations that deviate from the suggested guidelines.

As a consequence, stockers tend to prioritize proximity to the incoming spur during the put-away process, as it minimizes the distance they need to walk. This preference is reflected in Figure 5, which illustrates a higher *occupancy rate* in storage locations near the incoming spurs. The occupancy rate indicates the percentage of storage locations that hold at least one item. Furthermore, the *fill rate*, which represents the volume of the items in the bin in relation to the maximum storage volume of the bin, is also higher in proximity to the incoming spurs. Conversely, storage locations situated farther away from the incoming spurs exhibit considerably lower occupancy and fill rates, as depicted in Figure 6.

Within the warehouse, an experimental initiative is being carried out on one of its four floors, involving the implementation of put-away advice for stockers. This guidance is provided by the Warehouse Management System (*WMS*) and is rooted in an ABC classification methodology. The aim of this classification is to ensure that items with high sales frequency are positioned closer to the incoming and outgoing spurs. However, it should be noted that the advice provided by the WMS is limited to specifying the aisle where the product should be placed, rather than providing precise location instructions. The three remaining floors of the warehouse do not employ any form of put-away advice. As a result, the distribution of items across storage locations on these floors occurs in a random manner.

Figure 5: Storage locations near the incoming spur, with high occupancy and fill rate

Figure 6: Storage locations far from the incoming spur, with low occupancy and fill rate

1.2.3 Picking

Upon the arrival of a customer order, the subsequent step involves picking the required items from the shelves. However, prior to initiating the picking process, the order undergoes optimization within the Warehouse Management System (*WMS*). This optimization aims to enhance the efficiency of the picking process by following a series of predefined steps.

Initially, the orders are subjected to filtering based on specific characteristics, such as the nature of the delivery (e.g., same-day delivery) or the type of sorting and packing (e.g., manual or automatic). Following the filtering stage, the orders are sorted according to their priority, which is determined by the planning team. Only orders matching the designated priority level are considered for further processing. Subsequently, the cut-off time assumes prominence in the sorting procedure. Orders with the earliest cut-off times are accorded higher priority. Once the orders are sorted based on the cut-off time, the software further arranges them based on the delivery date. Orders with the earliest delivery dates receive the highest priority in this step. Lastly, the difficulty of picking is taken into account during the sorting process. Items that are deemed more challenging or complex to pick are prioritized for handling first. This consideration ensures that the most difficult items are addressed early in the picking process. By undertaking this filtering and sorting process, the output is a sorted list of items that are ready to be picked for order fulfilment.

Subsequent to the sorting and prioritization process, the software proceeds to allocate the identified items to specific pools and zones within the warehouse. Notably, the orders that were deemed most challenging during the previous sorting process are assigned to a pool and zone first. In the allocation of orders to zones, a ranking system based on stock availability is employed. This entails selecting the zone with the highest stock level as the first choice. Once the orders are appropriately assigned to their respective zones, the outcome of this process comprises two lists: a list of pools containing a collection of zones, and a list of zones containing the corresponding orders that need to be picked within each specific zone.

Subsequently, the software proceeds to enhance the pools by incorporating the precise locations of the items within each pool. The determination of these locations is based on the application of two primary principles: First Expired, First Out (FEFO) and First In, First Out (FIFO). Initially, the FEFO and FIFO principles are applied to identify the appropriate locations for item retrieval. This ensures that items nearing their expiration dates or those that were first received are selected first for picking. Once the FEFO and FIFO rules have been applied, the software prioritizes locations that involve multiple picks over those that only require a single pick. This prioritization is based on the premise that consolidating multiple picks at a single location maximizes picking efficiency. The outcome of this process is a list of orders that includes the precise locations of the items to be picked.

After the aforementioned processes, the software creates *pick batches*. During this phase, the physical constraints of a tote (weight and volume limitations), are taken into account. Each pick batch is designated to be assigned to a single tote, ensuring efficient utilization of space within the tote. In situations where the number of picks assigned to a particular pool and zone exceeds the tote's capacity, a sorting mechanism is employed to accommodate the constraints. Picks within the pool and zone are sorted in alphabetical order based on their locations, and subsequently assigned to a tote. On average, a pick batch comprises 11.2 items, all of which are consolidated within the same tote.

Following the determination of item allocations to specific pick batches, an algorithm is employed to determine the most efficient *picking route*. This algorithm calculates the shortest path that an order picker should traverse to collect all the assigned items. By identifying the path with the minimal distance, the optimal picking route is established. A predefined time budget is set for picking each item, allowing approximately 31.3 seconds per item. In other words, an order picker is expected to complete the picking of 115 items per hour based on this time constraint. However, in practice, the actual time required for picking an item is approximately 26.8 seconds, enabling the picking of around 134 items per hour.

Upon the completion of item collection for a specific pick batch, the order-picker proceeds to place the filled tote onto the outgoing *spur*, directing it towards the packing area. As illustrated in Figure 4, the outgoing spur is positioned at the rear end, adjacent to the conveyor belt where totes arrive from the floor above. Subsequently, the order-picker retrieves a new empty tote, signalling the start of a fresh pick batch. While there may be multiple order-pickers operating within a single zone, only one order-picker is assigned to pick items for a particular pick batch at any given time. Furthermore, it is important to note that an order-picker remains confined to their assigned zone throughout the picking process, ensuring efficiency and focus within their designated area.

Upon placing the tote on the conveyor belt, there is a possibility that the tote will be temporarily stored within a buffer tower known as the *stingray* (depicted in Figure 7). This scenario often arises in the case of *multi-orders*, which encompass multiple items. Given that the items for a single multi-order may be located in different zones within the warehouse, and thus assigned to different pick batches, they may end up being collected in separate totes. In such instances, the stingray serves to collect and temporarily hold all the totes associated with the same multi-order. This temporary storage of totes ensures that they are released simultaneously towards the packing area. This approach prevents a single tote from obstructing other totes within the packing area, thereby maintaining an organized and efficient workflow. However, it should be noted that if a tote solely consists of *mono-orders*, which entail orders containing only one item, it may bypass the stingray altogether. In the case of mono-orders, there is no need to await the collection of all items within the order, as there is only one item to be retrieved.

Following the passage of a tote through the stingray, it proceeds to travel towards a specific packing area within the warehouse. The determination of this specific location is indirectly influenced by the Control Room, which assesses the need to activate certain machines. This decision is based on ensuring that a minimum number of orders are present, guaranteeing the efficient operation of the machines. Utilizing this information, the software subsequently determines the appropriate packing and/or sorting procedures, including the specific locations and methods for packaging and sorting packages.

Figure 7: The Stingray, a tote buffer

1.2.4 Sorting & Packing

In the case of a mono-order requiring manual packing (Mono-Manual), the tote containing the mono-orders is transported to the designated packing stations, where operators undertake the task of packing the items. The software provides guidance regarding the appropriate box in which each item should be packed. In an effort to reduce material waste, items that are already enclosed within a robust box may not require an additional packaging box. These items are referred to as "no-packs."

On the other hand, if the mono-orders are to be packed by an automated machine, the tote carrying the orders is conveyed to the *CMC* (automatic packaging machine). At the CMC, the items are placed on the conveyor belt just before entering the machine. This specialized machine efficiently carries out the packaging process by precisely cutting the appropriate amount of packing material, ensuring optimal use of resources.

In situations where an order consists of multiple items, a sorting process is necessary to ensure that all items are consolidated within the same package. This sorting process is carried out through two distinct methods: manual sorting and automatic sorting. In the case of manual sorting, totes containing the items of a multi-item order are directed to one of the designated manual sorting stations (Multi-Manual). An operator extracts the items from the tote and places them in specific locations on a shelf. Once all the items belonging to a particular order have been placed in their respective location, another operator retrieves these items and proceeds to pack them together in a single box. Also in this manual sorting and packing process, the software provides instructions to the operator regarding the appropriate box to be used for packaging.

In contrast, the automatic sorting method involves an operator extracting the items from the totes and placing them on a conveyor belt. The conveyor belt then passes through multiple packing stations. As the items move along the conveyor belt, they reach the designated packing station corresponding to their respective order, where they drop into a chute. Once all the items belonging to a particular order have been collected in the same chute, an operator manually packs them in a box.

Finally, the packing process includes the option of utilizing the *SmartMailer* machine, specialized in packing small-sized items. In this process, an operator places the small item into a compact carton tray. The tray then proceeds through the machine where it ends up in an envelope. The machine automatically closes and seals the envelope. Additionally, the SmartMailer machine automatically prints and includes the appropriate address on the envelope.

Following the packing process, regardless of whether it was performed by an operator or a machine, the completed package is conveyed to another location within the warehouse, where Company Y operates. Company Y carries the responsibility of managing the delivery of these packages to the respective customers.

1.3 The Problem Identification

In Section 1.3.1, we present the problem context within the Warehouse Operations team of Company X. Building upon the problem context, Section 1.3.2 outlines a problem cluster, which systematically captures all the problems encountered within the company that are associated with the problem context. Leveraging the insights from the problem cluster, Section 1.3.3 focuses on the identification of the core problem. This involves a thorough analysis and evaluation of the problems highlighted within the problem cluster.

1.3.1 Problem Context

Every six weeks, the Warehouse Operations team determines the *stockmax*, this is the maximum number of items that can be stored within the warehouse, while still being efficient. Thus, this is lower than the true capacity of the warehouse since the team takes a predefined value for the *fill rate* and *occupancy rate* into account to make sure the warehouse is not filled to the roof. Upon comparing the stockmax determined by the Warehouse Operations team and the actual stock levels (as depicted in Figure 8), it becomes evident that by the end of 2022, the stock level had surpassed the stockmax. This discrepancy arose due to a lower-than-expected number of outbounded items, coupled with an unchanged number of inbounded items. Consequently, an excess of items accumulated within the warehouse, surpassing its intended capacity.

Although in the other periods, the operations may appear to run smoothly, the Warehouse Operations team highlights the frequent occurrence of problems arising from high occupancy and fill rates that exceed the threshold values. Currently, the threshold value for the fill rate is set at 35% due to the increasing item size, making it difficult to attain a higher fill rate since fewer items fit the storage location. For the occupancy rate, the threshold value is set at 85%. Although it would sound more efficient to have a higher occupancy rate, the following is stated by Sutterer (2010): "facilities exceeding 80-85% utilization are ready for expansion". Issues following from exceeding the threshold values contribute to reduced warehouse efficiency, requiring additional time and effort from the Warehouse Operations team, and incurring increased costs. Such problems emerge when the communicated stockmax proves to be inaccurate.

If the communicated stockmax is too low, indicating that the warehouse could store more items than communicated, it leads to two main problems. Firstly, lost sales may occur due to an inability to accommodate items. Secondly, the workforce within the warehouse may become inefficient as workers have fewer items to process. In response to such situations, steps can be taken, such as arranging additional deliveries or adjusting the workforce by requesting flexible workers to stay at home. These actions aim to mitigate

Figure 8: Actual stock level compared to estimated stock capacity

the negative impact caused by an inaccurate stockmax and optimize operational efficiency within the warehouse.

When Supply Chain Excellence (SCE) acquires more products than the warehouse's stockmax allows, the Logistics Service Provider (LSP) adjusts the put-away ratios once the occupancy rate, excluding shelf level three, reaches a specified threshold. Shelf level three is excluded from this calculation because it only comprises storage locations suitable for small products. The inclusion of these locations would bias the occupancy rate to be much lower (approximately 60% of all storage locations are situated on shelf level three, with an average occupancy rate of 47% since January 1, 2021).

In case the occupancy rate remains below 90%, an operator is expected to put away approximately 145 items per hour. However, when the warehouse's occupancy rate is high, operators encounter challenges in finding available storage locations, thus requiring more time for the put-away process. In cases where the occupancy rate exceeds 90%, the Logistics Service Provider adjusts the put-away ratio to 141 items per hour and modifies their planning accordingly. Furthermore, if the occupancy rate surpasses 93%, the put-away ratio is further reduced to 138 items per hour. These adjustments incur financial consequences as workers' efficiency decreases, necessitating additional time for the put-away process.

Shifting the put-away ratio to 141 items per hour incurs a cost of €539 per working day, whereas reducing it to 138 items per hour results in costs of €964 per working day. It is worth noting that the operators of the Logistics Service Provider work from Monday to Friday. Based on these cost considerations, an estimation of the expenses associated with lowering the put-away ratios in 2021 and 2022 indicates a total amount of €172,627.

Additionally, in situations where the occupancy rate and fill rate within the warehouse exceed the thresholds, the Warehouse Operations team (WO) employs the Traffic Light Model as a strategic approach. Developed by the team, this model serves as a framework outlining a set of inventory control guidelines to be adhered to by various teams operating within the warehouse. An overview of these guidelines can be found in Figure 9, illustrating the key components and principles of the Traffic Light Model.

According to the Warehouse Operations team, the utilization of the actions stated in the Traffic Light Model occurs with notable frequency, particularly during peak periods like Black Friday and Christmas. This emphasizes the significance of employing the model as a strategic tool to manage inventory. Figure 10 provides an overview of the specific time periods during which Company X operated within different scenarios of the Traffic Light Model, categorized as green, orange, red, or black. Concurrently, the figure displays the corresponding stockmax values as determined by the Warehouse Operations team and stock levels during those periods.

It is important to highlight that the Warehouse Operations team implemented a modification to the Traffic Light Model on September 23rd of 2022, whereby the threshold value for the fill rate was lowered by 5%. This adjustment was made in response to the increasing size of items in the inventory. By revising the threshold, the team aimed to account for the evolving nature of the products being stored and to ensure that the model remained relevant and effective in managing inventory control.

Traffic Light Model		
Scenario	Metrics	Action to be taken
Black	Occupancy rate >90% AND Fill rate >35%	Refuse deliveries
Red	Occupancy rate >85% AND Fill rate >30%	Daily calls with CR-OC Send suppliers to other warehouse if possible Decrease inbound volume Send mail with weekly update to other teams Set up 'rejectlist'
Orange	Occupancy rate >85% OR Fill rate >30%	Validation with logistics provider about operational situation Analysis to determine the reason Relocate/consolidate to make up free space Check with replenishment (workload) Distribute roll containers to zones with lower occupancy rate Plan weekly meetings about ratio's, option to use multisku and map impact Meeting about the prediction of shelving with CM + inbound FC Check possibilities to send suppliers to other warehouse Check possibilities to place roll containters at the end of an aisle Daily update in the stand-up
Green	Occupancy rate < 85% AND Fill rate <30%	No actions needed

Figure 9: Traffic Light Model

Table 2 presents the percentage of time that Company X operated within each scenario from 1/1/2021 until 31/12/2022. These percentages reveal that the company frequently found itself in scenarios other than the desired green state. However, through the implementation of the actions outlined in the Traffic Light Model (Figure 9), the teams successfully averted more severe scenarios from occurring. Nonetheless, it is important to note that even the orange scenario is considered undesirable. Thus, the overarching objective is to minimize the duration for which Company X operates in any scenario other than the green state.

Table 2: Percentage of time in each scenario between 1/1/2021 and 31/12/2022, based on the Traffic Light Model

Scenario	**Percentage of time**
Green	30.3%
Orange	59.8%
Red	3.3%
Black	6.6%

In order to gain a better understanding of the magnitude of the stockmax inaccuracies, an examination of stock levels at the precise moments when the company transitioned into or out of an orange scenario is undertaken. It can be assumed that these stock levels at these moments represent the actual stockmax values. Adherence to these values would have ensured the maintenance of a green scenario. As delineated in Figure 10, there exist nine discrete instances on the timeline where transitions into or out of an orange scenario occur. The stock levels at these specific dates, the stockmax as determined by the team, together with the difference between these values, are presented in Table 3. This data clearly shows the inaccuracy of the current stockmax calculation.

Table 3: Warehouse Operations stockmax and actual stockmax

Date	**Stockmax WO**	**Actual Stockmax**	**Difference**
28/01/2021	3,900,000	2,996,871	903,129
16/08/2021	3,900,000	2,964,663	935,337
15/10/2021	4,300,000	3,145,350	1,154,650
05/12/2021	4,300,000	2,929,828	1,370,172
27/12/2021	4,300,000	2,842,127	1,457,973
26/01/2022	3,500,000	2,932,930	567,070
13/06/2022	3,000,000	2,339,683	660,317
17/09/2022	2,600,000	2,274,383	325,617
24/12/2022	2,600,000	2,084,491	515,509
		Average Difference	**876,641**

Section 1.3.2 delves into further detail regarding the action problems outlined in the Traffic Light Model, as well as the underlying problems that necessitate the adoption of these actions.

Figure 10: Scenario timeline since 1/1/2021, including stock levels and stock capacities. The colours in the graph correspond to the colours of the Traffic Light Model. Discrepancies between calculated and actual capacity are evident, as indicated by the graph's colours. Despite remaining below capacity, the company frequently operates in orange, red, or black scenarios.

1.3.2 Problem Cluster

With a clear understanding of the problem context, we now construct a problem cluster to identify the interdependencies among various issues. The problem cluster diagram illustrates the relationships between different problems, where an arrow from problem X to problem Y signifies that problem X influences problem Y. On the right side of the diagram, we observe the action problems. According to Heerkens and van Winden (2017), an action problem refers to a disparity between the norm and the perceived reality as acknowledged by the problem owner. In our case, the Warehouse Operations team assumes the role of the problem owner as they are directly responsible for addressing the action problems. The white blocks within the cluster represent problems that ultimately impact the action problems. The left and top sides of the problem cluster illustrate potential core problems that may underlie the identified issues. The problem cluster presented in Figure 11 highlights five potential core problems.

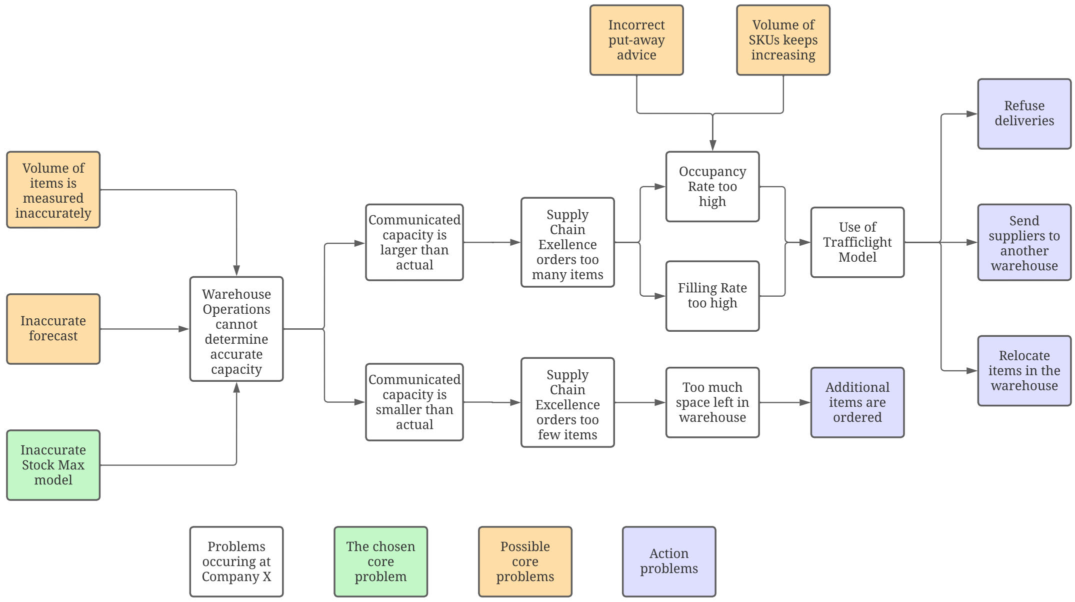


Figure 11: Problem cluster of Company X

Volume of SKUs keeps increasing

The first core problem pertains to the *Volume of SKUs keeps increasing*. Conversations and analysis of data (see Figure 12) reveal that the item volume remained relatively stable at just over four litres throughout 2021. However, since the beginning of 2022, the item size has increased to approximately seven litres. This upward trend poses challenges for the Warehouse Operations team as larger items may not always fit within existing storage locations alongside other items, necessitating the allocation of empty storage locations. Consequently, this results in a higher occupancy rate. Additionally, given that larger items occupy more space and the total storage capacity of the warehouse remains constant, a reduced number of items can be accommodated. The Logistics and Strategy team has proposed several potential approaches to address this issue, including options such as increasing the size of distribution boxes or eliminating the use of these boxes altogether to create more space for larger storage locations.

Figure 12: Increase in item volume since 2021

Incorrect put-away advice

An *Incorrect put-away advice* emerges as a potential core problem. Put-away, defined as the physical assignment of incoming unit loads to storage locations (Baruffaldi et al., 2020), plays a crucial role in ensuring operational efficiency. As highlighted by Park (2011), effective put-away practices can significantly enhance overall efficiency as well as order-picking efficiency. However, the selection of appropriate storage locations requires careful consideration. According to Chan and Chan (2011), multiple factors influence the placement of products within a warehouse, including order picking methods, storage system size and layout, material handling systems, product characteristics, demand trends, turnover rates, and space requirements. In the case of Company X, its current distributed storage policy fails to account for several of the abovementioned factors, such as storage system size, layout, and space requirements. To address this issue, one of the four floors in the warehouse is currently experimenting with a put-away advice system. This system utilizes an ABC classification, whereby the Warehouse Management System (*WMS*) advises the stocker on which aisle to place the item. However, the WMS does not provide precise location guidance. On the remaining three floors, no specific put-away advice is given. Instead, general guidelines are provided, such as placing liquids on the lowest shelf, avoiding the placement of heavy items on high shelves, and opting for the smallest possible storage location. In practice, operators often choose empty storage locations near the incoming spur, even for items with low sales frequency. Consequently, frequently sold items may end up being placed farther away from the incoming spur, while unnecessarily large storage locations may be utilized for smaller items. This is evident in Figure 13, which clearly demonstrates that numerous small items occupy storage locations with significant capacity.

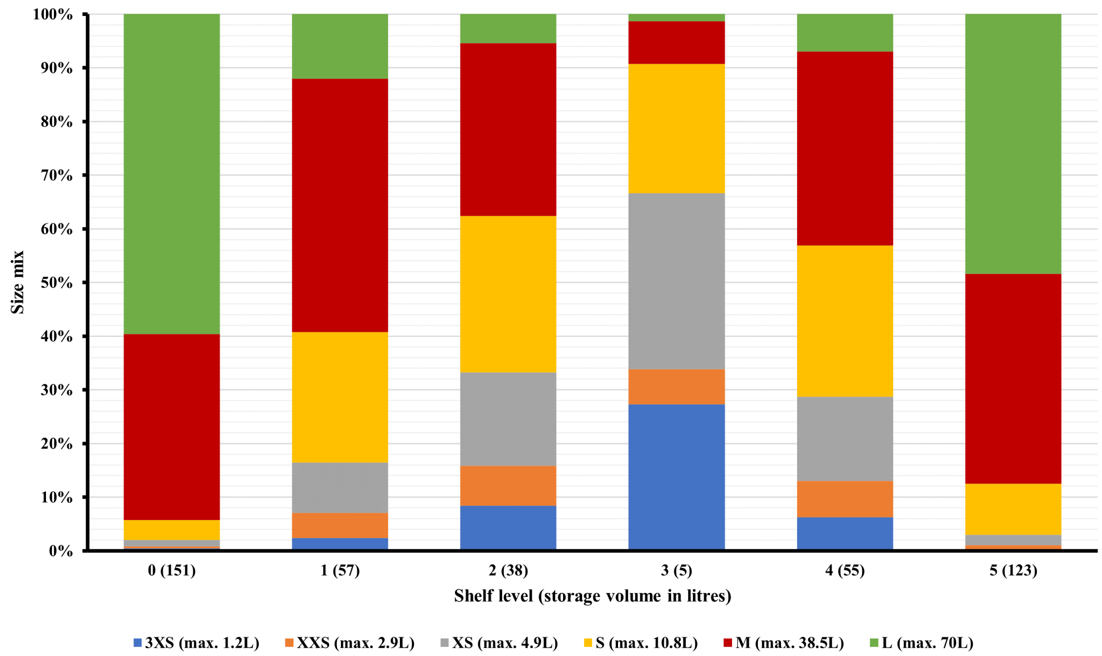


Figure 13: Size mix per shelf

Volume of items is measured incorrectly

Another potential core problem is that *Volume of items is measured inaccurately*. Upon the first arrival at the warehouse, items undergo volume measurement at the 13^{th} T-Car station using a machine. In cases where items possess a squishy nature, manual re-measurement by an operator might be necessary. However, if either the machine or the operator incorrectly measures the volume of these newly arrived items, it leads to erroneous output when utilizing this volume in subsequent calculations due to inaccurate inputs, thereby affecting the accuracy of their assessments. For instance, consider a scenario where the true volume of an item is one litre, but the machine erroneously measures it as two litres. In this case, the item's contribution to the fill rate calculation will be twice as much as its actual volume, skewing the calculation and potentially leading to misinterpretations of warehouse capacity utilization.

Inaccurate Stockmax model

Another potential core problem is the *Inaccurate Stockmax model*, which is utilized to determine the warehouse's desired storage capacity, such that the operations still run smoothly, in upcoming periods (*stockmax*). Employing this Stockmax model, the company aims to determine the number of items that fit the warehouse while remaining under the threshold values for the fill rate and occupancy rate. This model considers the projected outbound *size mix* of items to calculate the total number of items that can be accommodated at once in the warehouse while maintaining the desired occupancy rate and fill rate thresholds. Capacity Management utilizes this calculated capacity to allocate storage space for Company X's own products and the products of Company X's *partners*. Based on the allocated capacity for Company X's own products, Supply Chain Excellence makes purchasing decisions. However, if the calculations within the Stockmax model are inaccurate, it will result in an erroneous estimation of the stockmax. Consequently, an incorrect quantity of items may be purchased, leading to either overstocking or understocking situations. The current model is criticized for relying on numerous assumptions, being overly generalized, and employing multiple averages, which could compromise the accuracy of the capacity estimation. To address this possible core problem, it is crucial to refine and enhance the Stockmax model. This entails improving the accuracy of the assumptions used, incorporating more specific data, and reducing reliance on general averages. By enhancing the model's precision, the Warehouse Operations team can obtain more reliable information about the stockmax, enabling better-informed purchasing decisions and more efficient management of inventory levels.

Inaccurate forecast

Another potential core problem is the presence of an *Inaccurate Forecast*. To determine the allocation of suppliers to specific warehouses, Capacity Steering employs the *SAM*-model. By considering the delivery patterns of various suppliers, which may involve predominantly large or small packages, the team can estimate the expected shares of different size groups to be outbounded in a period, known as the *size mix*.

1.3.3 Identification of the Core Problem

To determine the core problem among the potential options, the method proposed by Heerkens and van Winden (2017) is utilized. Using this method, four criteria should be considered:

1. The core problem must be a genuine issue observed within the company.
2. There cannot be a direct cause for the core problem.
3. The core problem must be susceptible to influence.
4. If multiple options remain, prioritize the problem with the highest impact at the lowest costs.

Volume of SKUs keeps increasing

Upon analyzing the first potential core problem discussed in Section 1.3.2, which pertains to the increasing volume of SKUs, it becomes apparent that Company X has limited influence over this issue at its core. While the company is taking measures such as removing distribution boxes to address the problem, it is important to recognize that the underlying issue itself cannot be directly influenced by Company X. As a result, this problem cannot be considered the core problem.

Incorrect put-away advice

The following potential core problem is the issue of incorrect put-away advice. Within the warehouse environment, having accurate put-away advice is vital for optimizing efficiency. Concerns have been raised by certain employees at Company X regarding the current put-away advice, prompting the belief that improvements in this area could yield cost savings and enhance overall efficiency. Currently, a dedicated team is conducting tests where the Warehouse Management System (*WMS*) provides put-away advice based on an ABC classification, and future experiments will explore additional criteria. These criteria include assigning operators to different aisles to prevent congestion and time wastage, as well as considering the occupancy rates of the aisles. For instance, if aisle A has an 80% occupancy rate and aisle B has a 50% occupancy rate, the advice would be to place the item in aisle B. While this may require operators to walk a greater distance, the advantage lies in avoiding the need to search for an available storage location. The ongoing research conducted by the specialized team suggests that implementing comprehensive put-away advice across all floors could result in a 5% cost reduction. Given the existing efforts dedicated to addressing this problem, it is not selected as the core problem.

Volume of items is measured incorrectly
The third potential core problem pertains to the inaccurate measurement of item volumes. Generally, the machine used for measuring item volumes is deemed accurate enough for most cases. The need for remeasuring only arises in situations involving squishy items, and even then, it occurs infrequently. Considering the feedback from the warehouse teams, who indicate that the machine's measurements are sufficiently accurate, and the limited occurrence of remeasuring, it can be concluded that this is not a prevailing issue at Company X. Therefore, the problem of inaccurate measurement of item volumes is not selected as a core problem.

Inaccurate forecast
The fourth potential core problem revolves around inaccurate forecasts, specifically the outbound size mix as determined by Capacity Steering. This aspect is controlled by a team separate from the Warehouse Operations team, rendering the Warehouse Operations team unable to directly influence this problem. Additionally, it is widely acknowledged that achieving complete accuracy in forecasting is exceedingly challenging. Therefore, inaccurate forecasts are not deemed the core problem in this context.

Inaccurate Stockmax model
The core problem that remains is the inaccurate Stockmax Model. As can be seen in Figure 10, the stock levels clearly remained below the calculated *stockmax*. However, although the stock levels remained under the calculated stockmax, there were still too many items in the warehouse. This led to extraordinary values for the fill rate and occupancy rate, and thus inefficient warehouse operations. It is believed that this model, when calculating the stockmax accurately, can significantly impact warehouse operations. With an accurate Stockmax Model, the teams can better decide how many items can be delivered to the warehouse in order to remain under the threshold values for the fill rate and occupancy rate. With an accurate number for the stockmax value, less time is spent in a scenario other than the green one since we better know the limit of the warehouse. Moreover, as the Warehouse Operations team developed and implemented the model, they have the ability to influence and address this problem. Therefore, the core problem is:

An Inaccurate Stockmax Model.

1.4 The Research Problem

In Section 1.3.3 we identified the core problem. From this core problem, the following main research question arises:

How can the stockmax of the shelving area in the warehouse be accurately calculated while ensuring compliance with the predefined threshold values for the fill rate and occupancy rate?

In order to address the main research question, we impose some limitations on the research scope. We limit the focus of our study to the following specific domains.

Shelving area
It is hypothesized that implementing an accurate capacity calculation methodology for the shelving area of the warehouse will enable the company to maintain occupancy and fill rates below the predefined thresholds of 85% and 35%, respectively. The main research question focuses specifically on the shelving area, as identified by the Warehouse Operations team as the area with the greatest potential for improvement. This emphasis is justified by the observation that the bulk area and bunkers typically experience lower levels of utilization and, consequently, encounter fewer operational challenges. Therefore, by addressing capacity management in the shelving area, it is anticipated that significant improvements in warehouse efficiency and problem mitigation can be achieved.

Size groups
As stated in Section 1.1, the warehouse under investigation is limited to handling items within the size range of 3XS to L. Consequently, items belonging to size groups outside of this specified range or those lacking a specified size group are not considered relevant to our research objectives. Therefore, such items are omitted from our analysis. Additionally, it is worth noting that the presence of these excluded items raises concerns about the quality and reliability of their associated data. Subsequent examination of the data has revealed the presence of outliers, which are summarized in Table 4.

Table 4: The percentage of undesired size groups among the total number of items in the shelving area

Size Group	Percentage
XL	0.2180%
XXL	0.0014%
3XL	0.0023%
Blank	1.5780%
Unknown	3.2361%

Software
Regarding the selection of software for the new capacity calculation model, there is flexibility in choosing the appropriate software. While Microsoft Excel is currently used by Company X, alternative software options can be considered for this purpose. The primary criteria for the model are usability by the Warehouse Operations team and the ability to complete the calculation within a maximum timeframe of one hour.

Objective
The objective of this research is to minimize the duration of Company X's presence in orange, red, or black scenarios as defined by the Traffic Light Model, by improving the accuracy of the stockmax calculation. Ideally, Company X aims to operate predominantly in a green scenario, maintaining a state of optimal efficiency while ensuring an adequate stock level to mitigate lost sales. This approach strikes a balance between normal warehouse operations and sufficient product availability.

1.5 The Problem Approach

Section 1.3 and 1.4 of this study successfully completed the first step of the *Managerial Problem-Solving Method (MPSM)* as proposed by Heerkens and van Winden (2017); defining the problem. In Section 1.5, we delve into the second step of the MPSM, which outlines the problem-solving approach and strategy for addressing the core problem. Moving forward, step three entails a more detailed analysis of the problem, including a comprehensive examination of the current Stockmax Model. Subsequently, steps four and five involve formulating potential solutions, from which the most optimal one will be selected based on predefined metrics. Once the best solution is chosen, it will be implemented in step six, followed by an evaluation in step seven to assess the impact of the new solution.

1.5.1 Analysing the Problem

During the problem analysis phase, a comprehensive understanding of the problem and its context is essential. This involves examining the current approach used by the Warehouse Operations team to calculate the stockmax. Additionally, an investigation into the stockmax calculations of similar warehouses within Company X is conducted. Furthermore, the accuracy of the outbound *size mix* received from Capacity Steering is assessed by comparing it with the actual size mix of items in stock. Next to that, we explain the desired output of the model. This phase aims to address the following research questions as part of the problem-solving process:

1. What method is employed by the Warehouse Operations team for calculating the stockmax in the current context?

2. How is the calculation of the stockmax conducted in comparable warehouses belonging to Company X?

3. To what extent does the received outbound size mix provided by Capacity Steering align with the actual size mix of items in stock?

4. What is the desired output of the model?

1.5.2 Formulating Solutions

In the subsequent stage, a literature review is undertaken to identify potential models capable of calculating the stockmax of a warehouse characterized by dynamic item sizes. The stockmax of the warehouse is expressed as the total number of items that fit the shelving area at any point in time during a period while adhering to the *fill rate* and *occupancy rate*, where a period equals one month. After the literature study, we set up the crucial elements for a model tailored to the needs of Company X. Within the solution formulation phase, the following research questions are addressed:

1. What existing models can be identified within the literature that possess the capability to accurately compute the stockmax of a warehouse?

2. What specific inputs are required by the stockmax model to facilitate the calculation of stockmax?

3. What constraints should be incorporated into the stockmax model with respect to size mix, volumes, and put-away policies to ensure a comprehensive and realistic representation of the warehouse capacity?

4. Which assumptions should be taken into account pertaining to the characteristics of products, storage locations, inputs, and outputs when developing the stockmax model?

5. What does the model for Company X specifically look like?

1.5.3 Choosing a Solution

Upon identifying the relevant models from the literature, the next step involves employing historical data to estimate the stockmax of the shelving area within the warehouse. To assess the alignment between the calculated stockmax and the actual stockmax, specific metrics are utilized. Through the examination of historical data, the congruence between the output generated by the new method and the actual stockmax is assessed. The pursuit of an optimal solution necessitates addressing the following research inquiries:

1. How can the best solution be determined?

2. What are the experimental inputs?

3. What are the experimental outputs?

4. Which solution emerges as the optimal, based on the established criteria and assessment?

1.5.4 Implementing and Evaluating the Solution

The selected model from the previous step, although deemed the most favourable among the available options, does not automatically guarantee superiority over the current method. To assess its efficacy, a benchmarking analysis must be conducted, comparing the chosen solution to the existing approach. The objective is to determine whether the new solution demonstrates improved performance compared to the current method. If a positive outcome is achieved, considerations will be made regarding the implementation of the model within Company X's operations. Following the implementation, Company X should evaluate the outcomes during subsequent periods. This phase aims to address the following research question:

1. How does the selected solution compare to the current method?

1.6 Conclusion

In Section 1.3 it became evident what the challenges at hand are in the warehouse environment. Following an investigation, it has been determined that the root cause of the challenges arising can be attributed to the inaccurate nature of the stockmax model. The primary objective of this study is to address this issue, with the main research question stated in Section 1.4 serving as our focus. To methodically address this research question, a systematic framework has been set up, as explained in Section 1.5. Subsequent sections of this research will execute the various steps delineated within this framework.

2 Problem Analysis

Chapter 2 delves into a detailed exploration of the problem at hand, focusing on the current stockmax calculation. Section 2.1 provides an in-depth examination of the existing method employed by Company X for calculating the stockmax. Subsequently, in Section 2.2, we undertake an investigation into the accuracy of the *size mix*, a critical input parameter for the stockmax model. This section scrutinizes the reliability and precision of the size mix data, examining its alignment with the actual inventory composition. Section 2.3 delineates the desired output of our model, setting a clear objective for our research.

2.1 Warehouse Capacity Calculation

The calculation of the *stockmax* is a tactical decision; it is determined for the next six weeks, contributes to the strategic goals of the company and helps achieve operational efficiency if done correctly. In this section, we explain the methodology employed to determine the stockmax and shed light on the teams involved in this decision-making process. The primary objective is to ascertain the appropriate quantity of items that can be accommodated within the shelving area of the warehouse while adhering to the threshold values for the *fill rate* and *occupancy rate*. By correctly calculating the stockmax, the warehouse can operate at its maximum efficiency. We now proceed to provide an overview of the teams engaged in this process.

2.1.1 The Teams

The determination of the stockmax involves the collaborative efforts of six distinct teams, each assigned with specific responsibilities as outlined below:

- Warehouse Operations (WO): This team is responsible for computing the anticipated costs and establishing the stockmax for the relevant warehouse.
- Logistics Service Provider (LSP): As a third-party entity overseeing the logistical operations within the warehouse, the LSP plays a crucial role in budget allocation and determining the stockmax for each period and warehouse.
- Capacity Steering (CS): Tasked with utilizing the Stock Allocation Model, the CS team focuses on determining the proportionate allocation of items across different size groups (*size mix*).
- Commercial Development (CD): The CD team carries the responsibility of generating the commercial forecast.
- Capacity Management (CM): Responsible for determining the warehouse share and distribution between own capacity and *partner* capacity.
- Supply Chain Excellence (SCE): As the team responsible for procurement activities, the SCE team is tasked with strategically purchasing items to meet demand.

Figure 14 illustrates the collaborative workflow of the teams involved and the specific decision-making points during the stockmax calculation process. It is important to note that although the diagram depicts a start and endpoint, this process is iterative and continuously repeated. The ultimate objective is to procure the appropriate quantity of items that ensure the warehouse maintains a fill rate not exceeding 35% and an occupancy rate not surpassing 85%, while maintaining the highest possible product availability.

2.1.2 Starting the Process

To initiate the process, the Warehouse Operations team (WO) periodically inputs the warehouse's stockmax into a designated software, typically every quarter. This stockmax represents the maximum number of items that can be accommodated in the warehouse, considering predefined *fill rate* and *occupancy rate* thresholds. The detailed procedure for calculating this stockmax will be outlined later in this document.

Next to that, the Commercial Development team (CD) generates a commercial forecast every six weeks. This forecast encompasses the projected total sales volume of both the company's own products and those of its partners. It is important to note that this forecast is not warehouse-specific but encompasses the overall sales projections across all warehouses operated by the company.

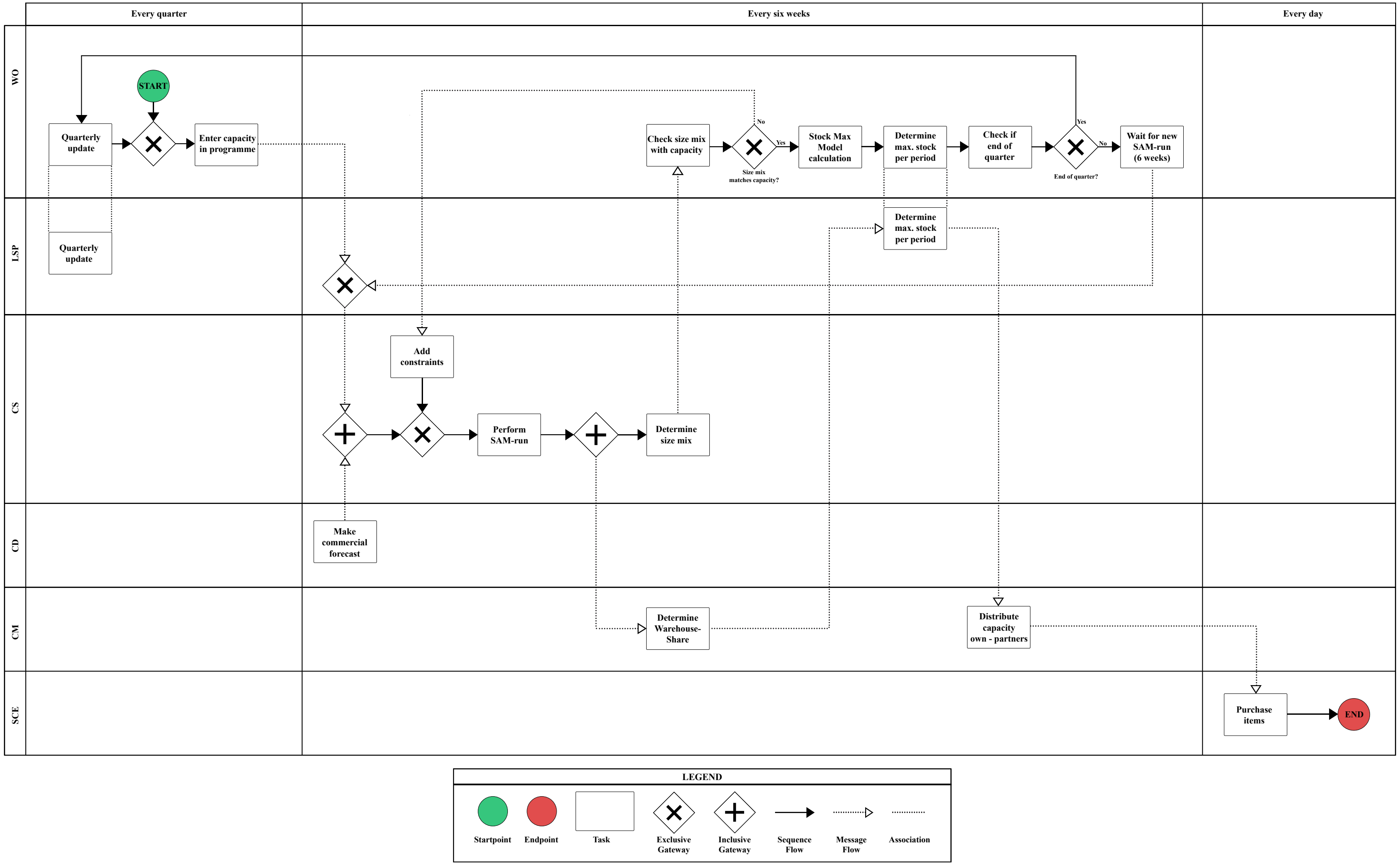


Figure 14: Continuous process to determine the stockmax with the goal of purchasing the correct number of items. WO = Warehouse Operations, LSP = Logistics Service Provider, CS = Capacity Steering, CD = Commercial Development, CM = Capacity Management, SCE = Supply Chain Excellence.

2.1.3 Stock Allocation Model

Once the stockmax is entered into the program and the commercial forecast is generated, the Capacity Steering team (CS) proceeds to execute the Stock Allocation Model (*SAM*). This model employs a comprehensive algorithm that utilizes these inputs, along with other inputs and constraints, to determine the optimal allocation of items from suppliers to warehouses. Subsequently, the SAM generates a warehouse-specific *size mix*, which delineates the expected proportion of items in each size group to be outbounded from the warehouse during a specific time period. In the context of this research, the size groups range from 3XS to L. It is essential to ensure that the cumulative percentages of size groups 3XS to L sum up to 100%, thereby covering the entirety of the expected item distribution.

The establishment of a size mix is feasible due to the fact that suppliers often specialize in specific size categories. However, a drawback of this model is the inherent uncertainty associated with the size mix. While the model determines the size mix based on supplier characteristics, there is no certainty that a supplier will exclusively provide items of designated size groups. It is plausible that a supplier that is anticipated to deliver predominantly smaller items, may also supply a substantial quantity of larger items. Consequently, the size mix introduces an element of uncertainty into the subsequent calculations of stockmax.

Subsequently, the outputs CS generates through the Stock Allocation Model are utilized by both WO and CM, with each team having distinct roles. WO's initial task involves assessing the alignment between the obtained size mix and the warehouse's stockmax. It is desirable to avoid a scenario where a significant portion of items is of size L since this poses challenges due to their incompatibility with smaller storage locations. To address this concern, CS introduces constraints within the Stock Allocation Model to limit the proportion of size L items, ensuring that, for instance, no more than 30% of the size mix pertains to size L within the warehouse. Consequently, the model generates a revised size mix, which WO subsequently evaluates for compatibility with the warehouse's stockmax. If the revised size mix seems correct, WO proceeds to employ the Stockmax Model.

2.1.4 Stockmax Model

WO has developed the Stockmax Model as a means to establish the *stockmax*, which signifies the upper limit for the number of items that the company aims to store in the warehouse at any point during a period. The stockmax is deliberately set lower than the warehouse's actual maximum capacity to enhance operational efficiency and thus minimize costs. By maintaining the stockmax at an optimal level, stockers can easily store items without encountering delays caused by the need to search for available storage locations. To facilitate the stockmax calculation, the model relies on specific inputs, including:

- Size mix per size group, as determined by CS (SM_{SG}).
- Average volume per size group in litres, retrieved from a dashboard $(AvgVol_{SG})$.
- Volume per storage location on shelf i in litres $(VpL_i\ (L))$.
- Number of storage locations on shelf i (LpS_i).
- Threshold value for the fill rate and occupancy rate.

The Stockmax Model operates through a step-by-step process, as exemplified in the following instance. Initially, WO acquires the average volumes for each size group *($AvgVol_{SG}$)* from a dashboard, denoting the average volume of the items within a specific size group currently present in the warehouse. Subsequently, WO employs Formula 1 to determine the anticipated weighted average volume *(WAV)* of an item. This involves multiplying the expected size mix percentage of a particular size group by the average volume of that corresponding size group, and then summing these products across all size groups to obtain the weighted average volume. Table 5 showcases an illustrative calculation applying this approach.

$$WAV = \sum_{SG\ in\ \{3XS,...,L\}} SM_{SG} \cdot AvgVol_{SG} \tag{1}$$

Table 5: Calculation of weighted average volume (WAV) per item

Size Group	Size Mix	Average Volume
3XS	14.47%	0.35
XXS	06.25%	1.13
XS	18.39%	1.17
S	23.45%	2.89
M	24.49%	7.60
L	12.95%	25.15
WAV	**100%**	**6.13**

Using the weighted average volume of an item, alongside other relevant inputs, WO proceeds to determine the quantity of items that can be accommodated within the warehouse. Table 6 presents a comprehensive illustration of this calculation. It is important to note that in this computation, WO has applied a maximum fill rate of 35% and a maximum occupancy rate of 85%.

The first column of Table 6 denotes the shelf level, where shelf level 0 corresponds to the bottom shelf and shelf level 5 corresponds to the top shelf. The subsequent column illustrates the number of individual storage locations available on each specific shelf level *(LpS_i)*. Additionally, the column adjacent to it displays the volume in litres, of each individual storage location on that shelf *(VpL_i (L))*. Utilizing Formula 2, the WO team calculates the total volume for each shelf level i *($TVpS_i$ (L))* in the fourth column.

$$TVpS_i = LpS_i \cdot VpL_i, \quad \forall\ 0 \leq i \leq 5 \tag{2}$$

By employing Formula 3, the Warehouse Operations (WO) team computes the aggregate volume across all shelves within the warehouse. This calculation is conducted based on the total volumes per shelf derived earlier.

$$TVpS_{total} = \sum_{i=0}^{5} TVpS_i \tag{3}$$

By utilizing the predefined threshold values for the fill rate (35%) and occupancy rate (85%), the Warehouse Operations (WO) team establishes the stockmax expressed in litres, denoted as *SML_i (L)*, for each shelf level. The incorporation of these threshold values aims to enhance operational efficiency and minimize costs, primarily due to the fact that stockers can easily locate an unoccupied storage location. Formula 4 outlines the computation of the stockmax in litres for each shelf level i.

$$SML_i = TVpS_i \cdot 0.35 \cdot 0.85, \quad \forall\, 0 \leq i \leq 5 \tag{4}$$

By aggregating the stockmax values across all shelves, as indicated by Formula 5, the total stockmax expressed in litres over all shelves can be determined.

$$SML_{total} = \sum_{i=0}^{5} SML_i \tag{5}$$

The WO team calculates the stockmax per shelf, referred to as the stockmax in items per shelf *(SMI$_i$ (Items))*, using the $\mathrm{SML_i}$ value obtained earlier. This is achieved by dividing the $\mathrm{SML_i}$ by the weighted average volume of an item, as presented in Table 5. The calculation of the stockmax per shelf level is represented by Formula 6

$$SMI_i = \frac{SML_i}{WAV}, \quad \forall\, 0 \leq i \leq 5 \tag{6}$$

By summing over the shelves, the stockmax of the shelving area of the warehouse can be determined. This summation is represented by Formula 7.

$$SMI_{total} = \sum_{i=0}^{5} SMI_i \tag{7}$$

In addition to the Warehouse Operations team, Capacity Management also utilizes the outputs of the SAM-run. CM's role is to determine the *warehouse share*, which represents the percentage of total items allocated to a specific warehouse. This information is then communicated to the LSP, who determines their own stockmax based on this warehouse share. WO and the LSP engage in discussions to align their respective stockmax values. These deliberations, often facilitated by expert opinions, result in the collaborative determination of a single stockmax for the warehouse.

Table 6: Current stockmax calculation

i	**$\mathrm{LpS_i}$**	**$\mathrm{VpL_i}$ (L)**	**$\mathrm{TVpS_i}$ (L)**	**FR**	**OR**	**$\mathrm{SML_i}$ (L)**	**$\mathrm{SMI_i}$ (Items)**
0	69,318	151	10,467,018	35%	85%	3,113,938	507,793
1	130,898	57	7,461,186	35%	85%	2,219,703	361,969
2	172,216	38	6,544,208	35%	85%	1,946,902	317,483
3	948,647	5	4,743,235	35%	85%	1,411,112	230,111
4	99,999	55	5,499,945	35%	85%	1,636,234	266,822
5	130,901	123	16,100,823	35%	85%	4,789,995	781,109
Total	**1,551,979**		**50,816,415**			**15,117,883**	**2,465,286**

Column Name	**Detailed Description**
i:	Shelf Level, with level 0 being the bottom row of the shelves and 5 the top row.
$\mathrm{LpS_i}$:	Locations per shelf (and total shelving area).
$\mathrm{VpL_i}$ (L):	Volume per location *L* in litres.
$\mathrm{TVpS_i}$ (L):	Total volume per shelf (and total shelving area) in litres.
FR:	Fill Rate.
OR:	Occupancy Rate.
$\mathrm{SML_i}$ (L):	Stockmax in litres, desired capacity per shelf (and total shelving area) in litres, accounting for the FR and OR.
$\mathrm{SMI_i}$ (Items):	Stockmax in items, desired capacity per shelf (and total shelving area) in items, accounting for the FR and OR.

2.1.5 Division of Stock Capacity and Item Purchasing

Following the determination of the stockmax by the LSP and WO, CM assumes the responsibility of allocating the capacity between the company's *partners* and the company itself. This allocation is based on the share of predicted outbound for each party. For instance, if the stockmax for the warehouse is 1,000,000 items and the company itself is projected to account for 80% of the outbound in a given period, while the partners are projected to account for the remaining 20%, the stockmax for the company would be set at 800,000 items, while the partners would have a stockmax of 200,000 items.

Using this allocated capacity, SCE undertakes the task of purchasing the appropriate quantity of items. With an accurate stockmax, we ensure that the warehouse's *fill rate* and *occupancy rate* remain below the predetermined threshold values of 35% and 85% respectively. In situations where the partners have insufficient stock capacity, while the company has available spare capacity, the company may allocate some of its unused capacity to the partners. By doing so, the partners can still utilize the warehouse for storing their items, while the company benefits from a higher occupancy rate. The same principle applies when the roles are reversed, and the partners have surplus capacity to offer.

2.1.6 Quarterly Update

The calculation of the stockmax is conducted on a six-week basis, while the input parameters of the Stock Allocation Model are updated quarterly. These input parameters encompass various factors such as costs associated with specific warehouse processes and the time required to complete them. WO is responsible for updating the warehouse's capacity, which serves as an input for the *SAM*-model.

Cost considerations hold significant importance within the Stock Allocation Model, as the algorithm tries to assign suppliers based on the most cost-effective warehouse. Once WO updates the input parameters in the designated program, the entire process restarts, initiating the cycle of stockmax calculation and allocation afresh.

2.1.7 Calculation at Similar Warehouses

Company X operates multiple warehouses, each with its own unique characteristics and activities. The warehouse in this research can be compared to two other warehouses within Company X in terms of product size and warehouse operations. From interviews, it became clear that the stockmax calculation employed by the Warehouse Operations teams in these two warehouses is found to be consistent with the approach used for the warehouse under investigation. While this does not yield new insights, it suggests that the proposed solution may be applicable to other warehouses within the company.

2.2 Size Mix Accuracy

The current model incorporates the outbound *size mix* as provided by Capacity Steering as one of its inputs. It is crucial to acknowledge that this size mix is a forecasted estimate and achieving complete precision is unattainable. Furthermore, it is important to note that the size mix is derived from outbound operations rather than the actual stock size mix. As a result, discrepancies may arise when comparing these two sets of data from the same periods. An interesting investigation is to assess whether the accuracy of the size mix can be improved by adjusting the timing, specifically by shifting the outbound size mix one or two periods forward to align with the period when these outbounded items were part of the inventory.

Furthermore, it is essential to emphasize that the test does not incorporate the definitive data for the size mixes. The company generates the size mix every six weeks, which is considered the most accurate and is utilized for determining the warehouse capacity. However, every six months, the company also establishes a size mix for the upcoming half-year. Unfortunately, the availability of the six-weekly size mixes is constrained to the past eight months. This means that for the preceding months, less precise size mixes from the half-yearly assessments need to be utilized.

A paired t-test was utilized to assess the presence of a statistically significant difference between the predicted value of the outbound size mix and the observed values of the stock size mix. The analysis was performed for each size group, considering three scenarios: no shift, a one-period shift (equivalent to 4 weeks), and a two-period shift (equivalent to 8 weeks). Despite the variation in scenarios, the procedural steps remained consistent throughout all tests. To determine the significance of any observed differences, a seven-step approach was followed.

1. Parameters:
    - $\mu_1,\ \mu_2$ = mean of sample 1 (size mix determined by CS) and sample 2 (actual size mix in stock) respectively.
    - $\bar{\mu}$ = mean of the differences between the observations.
    - σ = standard deviation of the differences.
    - $n_0 = 94$ = number of observations (pairs) with no shift.
    - $n_1 = 90$ = number of observations (pairs) with one period shift.
    - $n_2 = 86$ = number of observations (pairs) with two periods shift.
    - $\alpha = 0.05$ = significance level.
2. Hypothesis:
    - $H_0 : \mu_1 = \mu_2$
    - $H_1 : \mu_1 \neq \mu_2$
3. Test statistic:
    - $T = \frac{\bar{\mu}}{\sigma/\sqrt{n_i}}$
4. Distribution:
    - Our paired t-test utilizes a t-distribution with a significance level (α) of 0.05 and degrees of freedom equal to the sample size minus one ($n_i - 1$).
5. Rejection criteria:
    - In the case of the test without any shift, the sample size (n_0) is 94. Given a significance level of 0.05 and ($n_0 - 1$) degrees of freedom, the null hypothesis (H_0) is rejected if the absolute value of the test statistic exceeds the critical value of 1.9858, or in case the p-value falls below α.

- In the case of the test with a one-period shift, the sample size (n_1) is 90. Given a significance level of 0.05 and $(n_1 - 1)$ degrees of freedom, the null hypothesis (H_0) is rejected if the absolute value of the test statistic exceeds the critical value of 1.9870, or in case the p-value falls below α.
- In the case of the test with a two-period shift, the sample size (n_2) is 86. Given a significance level of 0.05 and $(n_2 - 1)$ degrees of freedom, the null hypothesis (H_0) is rejected if the absolute value of the test statistic exceeds the critical value of 1.9883, or in case the p-value falls below α.

6. Value of test statistic:
   - In this step, the test statistic value is computed individually for each size group, considering the three test scenarios: no-shift, one-period shift, and two-period shift.
7. Conclusion:
   - Here, we examine the existence of a statistically significant difference. As outlined in step 5, the null hypothesis is rejected when the absolute value of the test statistic surpasses the critical value specific to the conducted test, or in case the p-value falls below α

Table 7: Paired t-test to check for significant differences between predicted outbound size mix and actual stock size mix

Size Group	Shift	Test Statistic	P-Value	Accept/Reject
3XS	No Shift	-1.9417	0.0552	Accept
	1 period	-2.0168	0.0473	Reject
	2 periods	-2.1446	0.0348	Reject
XXS	No Shift	-4.6832	9.61×10^{-6}	Reject
	1 period	-4.7978	6.42×10^{-6}	Reject
	2 periods	-4.8009	6.70×10^{-6}	Reject
XS	No Shift	-1.5760	0.1184	Accept
	1 period	-1.9570	0.0535	Accept
	2 periods	-2.1994	0.0306	Reject
S	No Shift	-5.9660	4.37×10^{-8}	Reject
	1 period	-6.1902	1.80×10^{-8}	Reject
	2 periods	-6.2629	1.49×10^{-8}	Reject
M	No Shift	-0.0423	0.9662	Accept
	1 period	0.4400	0.6610	Accept
	2 periods	0.8448	0.4006	Accept
L	No Shift	15.7720	5.02×10^{-28}	Reject
	1 period	17.5499	1.21×10^{-30}	Reject
	2 periods	19.2050	1.15×10^{-32}	Reject

As depicted in Table 7, the results of the *no shift* test reveal that three out of six size groups exhibit no statistically significant difference. Notably, the *Large* size group displays a considerably high test statistic value. Similar patterns of substantial differences are observed in both the *one-period shift* and *two-period shift* scenarios. Moreover, it is worth mentioning that the size group *3XS* starts demonstrating a significant difference when the outbound size mix is shifted one period forward. Likewise, the size group *XS* exhibits a significant difference when the outbound size mix is shifted two periods forward.

Based on these test outcomes, we can infer that the size mix has the potential for improved accuracy. However, it should be acknowledged that we were unable to employ the most accurate size mixes. Instead, we had to resort to the half-yearly size mixes, which, in reality, have undergone multiple updates in an effort to enhance their accuracy. Despite observing significant differences between the predicted size mix and the actual size mix in half of the size groups, we still opt to utilize this size mix as an input for our model as it serves as a source of crucial information.

2.3 Desired Output of the Model

The computation of the stockmax, often perceived as a straightforward task, is accompanied by complexities that defy its apparent simplicity. As observed by Cosgrave (1997), the conventional notion of deriving warehouse capacity by dividing the overall cubic volume by the stowage factor falls short of capturing the true essence of the calculation. Indeed, the actual capacity of a warehouse is contingent not only upon its overall volume but also on factors such as spatial layout, the nature of stored goods, and operational characteristics (Cosgrave, 1997).

The complexity of calculating the stockmax for Company X arises from several factors, including the extensive size of the warehouse itself, the vast assortment of products that can be accommodated, the uncertainty of item volumes, and the implementation of a distributed storage policy. These complexities present considerable challenges in effectively managing warehouse operations in an efficient manner.

As underscored by Faber et al. (2013): "An important question, therefore, is how warehouse management, as a cluster of planning and control decisions and procedures, is organized in order to meet today's challenges." To optimize efficiency, control and planning of various processes within the warehouse are necessary. Ackerman (1997) and Frazelle (2001) elucidate key processes such as inbound flow handling, product-to-location assignment, product storage, order-to-stock location-allocation, order batching and release, order picking, packing, value-added logistics activities, and shipment. Notably, storage emerges as one of the most intricate and labour-intensive processes that wield significant influence over warehouse performance (Faber et al., 2013).

Given the criticality of storage operations and their associated processes, accurate stockmax calculation assumes significant importance. The determination of the stockmax is indispensable to avert lost sales, low picking and put-away ratios, and the eventual need to resort to mechanisms like the Traffic Light Model (Figure 9) to mitigate operational inefficiencies.

The proposed solution necessitates the development of a stockmax calculation model that strives for the utmost accuracy. This model should calculate a precise estimate of the warehouse's stockmax. By attaining the highest accuracy, the operational activities within the warehouse can be executed with optimal efficiency and effectiveness.

Additionally, to calculate the stockmax, the model should incorporate the put-away logic that is currently used in the warehouse. It would be insightful to compare and contrast the outcomes of three distinct calculation methods: a completely random approach, an optimal approach, and an approach in which the current put-away logic of operators is used (a logical assignment). Using the random approach, items are assigned to a random shelf level and a random bin on that shelf. With the optimal approach, we try to find a bin that suits the item best. This bin can be located on any shelf level. Lastly, when using the put-away logic of the operators, we essentially do the exact same as in the optimal methods, however, now we first determine the shelf level that an item, given its size group, should be located on. This can be done since we know what percentage of a size group is located on a specific shelf.

An analysis of these methods would enable us to discern whether one method outperforms the others in terms of efficiency and effectiveness. Notably, logical assignment, which accounts for the actual placement of size groups on shelves as currently observed in the warehouse, is anticipated to yield the most realistic results. Furthermore, an optimal calculation is of interest for future implementation, particularly in conjunction with a put-away advice that the company is currently developing. By incorporating the current state of the warehouse as an input, the accuracy of the capacity calculation can be enhanced.

2.4 Visualization of Example Models

To provide further clarity, we have generated a conceptual representation of an artificial warehouse consisting of three storage bins of varying sizes. Additionally, we have introduced products of three distinct sizes: S (orange), M (blue), and L (green). As previously mentioned, the size mix serves as one of the inputs to the model, denoting the percentage of S, M, and L items that need to be stored. For the purpose of this illustration, let us assume an equal distribution of 33.3% for each size group. This signifies that the number of items in each size group should be equivalent; thus, adding one item of size S necessitates the inclusion of one item of size M and one item of size L. It is important to note that, in the interest of simplicity, other constraints and assumptions have not been considered in this depiction. In reality, the number of size groups would be more extensive, the *fill rate* would limit the capacity of the bins and the *occupancy rate* would limit the number of available bins. Figure 15 depicts a visual representation of a potential model for calculating the stockmax while incorporating three distinct forms of assignment. In Figure 15a, the current state of the warehouse is presented, which serves as a fixed input for our model.

2.4.1 Random Assignment

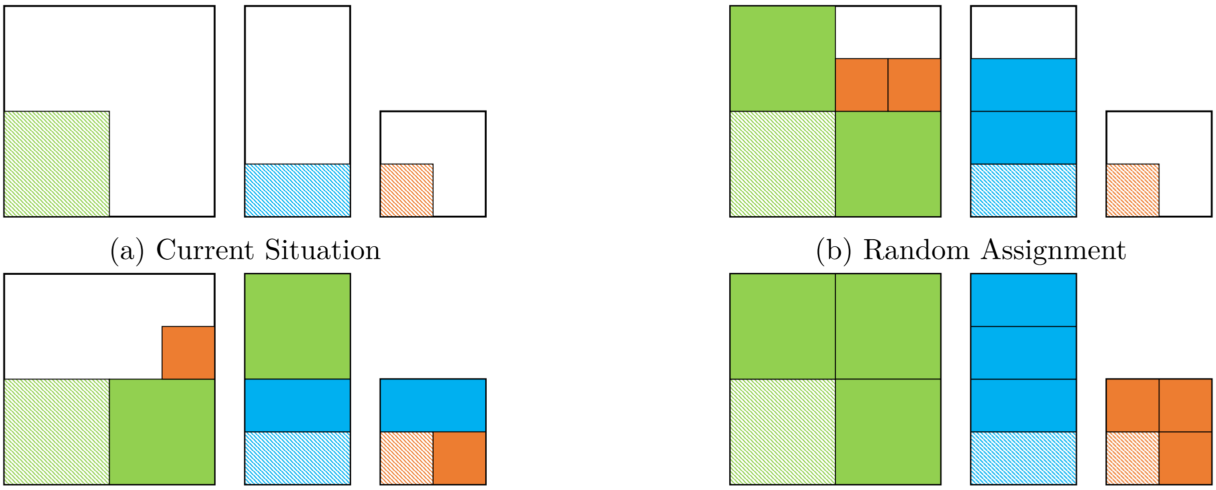
Figure 15b showcases a random allocation of items to storage bins. In this scenario, there is the possibility of adding additional S and M items. However, it is essential to note that including one item of size groups S and M necessitates the inclusion of another L item to maintain a size mix of 33.3% for all size groups. As depicted in the figure, this proves to be infeasible as we cannot include one more L item. In total, the warehouse can accommodate a maximum of three S, M, and L items.

2.4.2 Logical Assignment

Let us now examine the logical assignment depicted in Figure 15c, which incorporates a degree of logical reasoning. In this case, we observe that it is possible to include additional items in the large storage location. We could even include one item from each size group, thereby adhering to the size mix constraint. However, upon closer inspection of the specific location where these items would be stored, this allocation does not appear to be particularly sensible. We can add one more S, M, and L item to the large bin, but this would result in 50% of the S items being located in the largest storage bin and 50% in the smallest storage bin. This deviates from realistic expectations as operators in the warehouse would not store 50% of the S items in the largest storage bin, but rather use a smaller bin. Since we do not add an S item, we also do not add an M and L item to adhere to the size mix constraint. Thus, in a logical calculation, the capacity for each size group would be limited to three items.

2.4.3 Optimal Assignment

Finally, let us examine the optimal assignment illustrated in Figure 15d, where products are assigned to bins in an optimal manner. Regarding the size mix, we observe an allocation of four items for each product. Consequently, each size group accounts for 33.3% of the total item count, thereby satisfying the size mix constraint. This capacity allocation would be highly desirable. However, it is important to note that the warehouse currently lacks put-away advice. Next to that, we do not have knowledge about the exact item arrivals in the coming months. If such advice were in place, together with information about the item arrivals, our capacity could be expanded to accommodate four items for each size group.



(a) Current Situation (b) Random Assignment (c) Logical Assignment (d) Optimal Assignment

Figure 15: A visualization of assignment methods. Graph (a) shows the Current Situation of the warehouse, with one L item (green), one M item (blue) and one S item (orange). Graph (b) shows a possible outcome when assigning items to bins in a random manner. Stockmax = 12 items. Graph (c) showcases a possible outcome when the put-away logic of the operators is implemented. More items can be added, however, 50% of S items in the largest bin does not reflect how operators in the warehouse work. Stockmax = 12 items. Graph (d) reveals what an optimal assignment of items to bins would look like. Stockmax = 16 items.

3 Literature Study

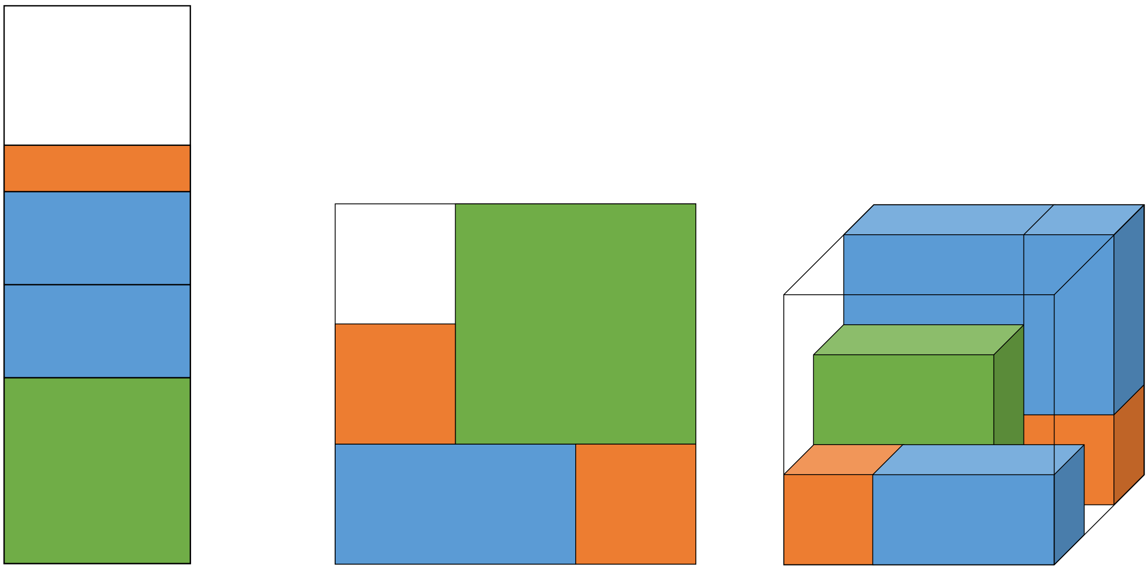
Chapter three serves as a resource for identifying models within the existing literature that can effectively compute the warehouse capacity in various ways. In Sections 3.1 and 3.2, we undertake a comprehensive evaluation of potential models available, considering their applicability and effectiveness. In Section 3.3, we investigate interesting heuristics that could be useful for the model of Company X.

3.1 Bin Packing Problem

When considering the assignment of items to storage locations, it is natural to draw connections with the renowned Bin Packing Problem (*BPP*). As articulated by Abdel-Basset et al. (2018), the classical BPP consists of a given set of items that need to be packed into one bin. Each item consists of a volume and each bin has a capacity. The main goal of the BPP is to use as few bins for item allocation. From this, we find that the Bin Packing Problem closely relates to our problem at hand as we also assign items to storage bins. For this reason, we delve into a comprehensive examination of this problem as it could give great insights into our solution approach.

3.1.1 Types of BPPs

The Bin Packing Problem encompasses multiple dimensions, specifically the first, second, and third dimensions, as depicted in Figure 16. In the 1-Dimensional BPP, items are arranged in a single line, allowing for vertical stacking only. Figure 16a provides a visual representation of this dimension. Figure 16b illustrates the concept of the 2-Dimensional BPP, where items can be placed both vertically and horizontally, enabling adjacent placement. Lastly, Figure 16c presents the 3-Dimensional BPP, which permits items to be positioned vertically, horizontally, and depth-wise. This 3D-BPP closely resembles the problem at hand in our warehouse setting. However, it is important to note that the primary distinction between the bin packing problem and our research problem lies in the given parameters. The BPP entails a fixed set of items that must be efficiently packed into the fewest number of bins possible, while our research problem involves a predetermined set of bins, and our objective is to determine the number of items that fit in these bins.



(a) 1-Dimensional BPP (b) 2-Dimensional BPP (c) 3-Dimensional BPP

Figure 16: Different types of Bin Packing Problems

3.1.2 Mathematical Model

The mathematical model for the 3D-Bin Packing Problem is described in the research paper by Hifi et al. (2010). In their study, each item i (i=1, 2, ..., n) is characterized by its width, w_i, height, h_i, and depth, d_i. The bins, on the other hand, have identical dimensions, with width W, height H, and depth D. The upper-bound of the bin label is represented as $\bar{\gamma}$, while the vector (x_i, y_i, z_i, γ_i) represents the geometric location of item i in bin γ_i. In this notation, $x_i, y_i, z_i \geq 0$ and $\gamma_i \geq 1$. The vector $(x_i, y_i, z_i) = (0, 0, 0)$ denotes the coordinates of the left-bottom-back corner of item i, and γ_i represents the label of the bin to which item i is assigned. The objective of the problem is to minimize the maximum label of the utilized bins, which is denoted as $\gamma = \max_{1 \,\leq\, \mathrm{i} \,\leq\, \mathrm{n}} \{\gamma_\mathrm{i}\}$. As per the formulation presented by Hifi et al. (2010), the Mixed Integer Linear Programming (*MILP*) model for the 3D-Bin Packing Problem is as follows:

$$\min \gamma \tag{8}$$

$$
\begin{aligned}
\text{s.t.}\quad & l_{ij} + l_{ji} + u_{ij} + u_{ji} + b_{ij} + b_{ji} + c_{ij} + c_{ji} = 1 && i \leq j = 1, ..., n && (9)\\
& x_i - x_j + W(l_{ij} - c_{ij} - c_{ji}) \leq W - w_i && i \neq j = 1, ..., n && (10)\\
& y_i - y_j + H(u_{ij} - c_{ij} - c_{ji}) \leq H - h_i && i \neq j = 1, ..., n && (11)\\
& z_i - z_j + D(b_{ij} - c_{ij} - c_{ji}) \leq H - d_i && i \neq j = 1, ..., n && (12)\\
& (\bar{\gamma} - 1)(l_{ij} + l_{ji} + u_{ij} + u_{ji} + b_{ij} + b_{ji}) && &&\\
& \; + \gamma_i - \gamma_j + \bar{\gamma} c_{ij} \leq \bar{\gamma} - 1 && i \neq j = 1, ..., n && (13)\\
& l_{ij}, u_{ij}, b_{ij}, c_{ij} \in \{0, 1\} && i \neq j = 1, ..., n && (14)\\
& 0 \leq x_i \leq W - w_i && i = 1, ..., n && (15)\\
& 0 \leq y_i \leq H - h_i && i = 1, ..., n && (16)\\
& 0 \leq z_i \leq D - d_i && i = 1, ..., n && (17)\\
& 0 \leq \gamma_i \leq \gamma \leq \bar{\gamma} && i = 1, ..., n && (18)
\end{aligned}
$$

The MILP model proposed by Hifi et al. (2010) incorporates several binary variables and constraints to address the 3D-Bin Packing Problem. Specifically, the formulation includes the following elements:

1. The binary variables l_{ij}, u_{ij}, b_{ij} and c_{ij} indicate the relative positions of items i and j with respect to left, under, behind, and the bin label, respectively. Where c_{ij} equals 1 if $\gamma_i < \gamma_j$.

2. Constraints 9, 10, 11 and 12 ensure the absence of overlap between packed items by enforcing mutual exclusivity in their positions.

3. Constraint 13 guarantees that items i and j are assigned to different bins if c_{ij} or c_{ji} equals 1. Conversely, if $l_{ij}, l_{ji}, u_{ij}, u_{ji}, b_{ij}$ or b_{ji} equals 1, it indicates that items i and j must share the same bin.

4. While not explicitly mentioned in the paper by Hifi et al. (2010), the binary nature of the variables can be inferred from constraint 14. Additionally, constraints 15, 16 and 17 impose dimensional constraints, ensuring that items fit within the specified bin dimensions.

5. Constraint 18 prevents items from being assigned to bins with higher labels than the bin with the highest label, while also restricting the highest bin label from exceeding the specified upper-bound, denoted as $\bar{\gamma}$.

Together, these constraints and variables form the mathematical foundation of the MILP model proposed by Hifi et al. (2010) for tackling the 3D-Bin Packing Problem.

3.1.3 Solvability

Solving the 3D-BPP poses significant challenges due to its inherent difficulty. The bin packing problem is widely recognized as *NP-hard* (Non-deterministic Polynomial-time hard). As acknowledged by Maarouf et al. (2008), this classification implies that finding an exact solution to this problem in polynomial time is infeasible, presenting a formidable challenge in devising efficient approaches. The efficiency of an algorithm is closely tied to its computational complexity. As elucidated by Mann (2017), an algorithm can be deemed efficient if it operates in polynomial time, wherein the number of steps required for inputs of size N does not exceed $c_1 \cdot N^{c_2}$, with appropriate constants c_1 and c_2. Regrettably, this criterion is not met by the bin packing problem. Despite extensive efforts by mathematicians and computer scientists spanning decades, no algorithm has been discovered that can compute an optimal solution within a reasonable timeframe. Nevertheless, it has not been definitively proven that achieving optimality is impossible, as noted by Sweep (2003). While it is theoretically feasible to solve the BPP optimally, the computational requirements are extensive. As Sweep (2003) stated: "At this time, no optimal solutions to the bin-packing problem may be derived by a computer without deriving nearly every possible solution. In other words, finding a perfect solution to one non-trivial instance of the bin-packing problem with even the most powerful computer may take months or years."

When addressing the bin packing problem, Lewis (1983) states that this problem is NP-hard in the strong sense, so there is little hope of finding even a pseudo-polynomial time optimization algorithm for it. However, there are several simple algorithms for it that are worth considering. In Section 3.3, we will delve into compelling heuristics that can be employed to tackle the bin packing problem.

3.2 Knapsack Problem

The Knapsack Problem (*KP*) presents another challenge that involves the allocation of specific products to a storage location, meaning that the idea of the Knapsack Problem corresponds with our problem at hand since we also allocate items to storage bins. Assi and Haraty (2018) provide a concise description of the Knapsack Problem as follows: "The Knapsack Problem is composed of n items, each with its associated weight and profit. The objective is to select a subset of the n items that maximizes the profit and that at the same time fits the knapsack." Various variants of this problem have been identified and studied. In the subsequent section, we provide a brief overview of the most renowned variants. Although the overarching objective remains consistent across these variants, each Knapsack Problem exhibits unique characteristics. Our elucidation encompasses a thorough examination of these distinguishing attributes, accompanied by a presentation of the corresponding mathematical models.

3.2.1 0-1 Knapsack Problem

We commence our discussion with the 0-1 Knapsack Problem (KP), alternatively referred to as the Binary Knapsack Problem. As indicated by Assi and Haraty (2018), the 0-1 KP represents the most widely recognized and extensively studied variant due to its manifold real-world applications. Ali et al. (2021) enumerate several domains where this problem finds practical utility, encompassing areas such as project selection, decision-making, water resource engineering, flood management, as well as resource distribution and allocation. In the context of the 0-1 KP, each item under consideration can be either chosen for inclusion within the knapsack or entirely disregarded. Within this formulation, the problem entails n distinct items, indexed by j, each possessing a corresponding weight denoted as w_j and a value represented by v_j. Additionally, we introduce the binary variable x_j, which assumes the value 1 if the item is selected for placement within the knapsack and 0 if it is omitted. To account for the weight constraint, we stipulate a maximum weight threshold of the knapsack denoted by W. Consequently, the mathematical model for the 0-1 KP is expressed as follows:

$$\max \sum_{j=1}^{n} v_j x_j \tag{19}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} w_j x_j \leq W \tag{20}$$

$$x_j \in \{0,1\} \qquad j = 1, ..., n \tag{21}$$

3.2.2 Bounded Knapsack Problem

The Bounded Knapsack Problem (BKP) closely resembles the Binary Knapsack Problem (KP) in many aspects. While the KP involves selecting items at most once, the BKP, as articulated by Assi and Haraty (2018), permits the inclusion of each item multiple times. As in the KP, we now also consider a set of n items denoted by index j, wherein each item possesses a weight represented by w_j and a value indicated by v_j. Our decision variable, denoted as x_j, remains central to the formulation; however, in the BKP, x_j assumes integer values instead of binary values. To account for the multiple inclusion of items, we introduce the parameter m_j, representing the maximum number of occurrences for item type j. Pisinger and Toth (1998) succinctly describe the problem as follows: "The problem is to select a number x_j $(0 \leq x_j \leq m_j)$ of each item type j such that the profit sum of the included items is maximized without the weight sum exceeding c." Analogous to the weight constraint in the KP, we employ W to denote the weight constraint of the knapsack. Incorporating these parameters, we can formally define the mathematical model for the Bounded Knapsack Problem as follows:

$$\max \sum_{j=1}^{n} v_j x_j \tag{22}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} w_j x_j \leq W \tag{23}$$

$$x_j \in \{0, 1, ..., m_j\} \qquad j = 1, ..., n \tag{24}$$

3.2.3 Unbounded Knapsack Problem

In addition to the Bounded Knapsack Problem (BKP), another prominent variant is the Unbounded Knapsack Problem (UKP). Pisinger and Toth (1998) describe the UKP as a special case of the BKP in which there exists an unlimited supply of each item type. In essence, the BKP and the UKP are nearly indistinguishable, with the exception that the UKP allows an unbounded quantity of items to be selected. However, as emphasized by Assi and Haraty (2018), the number of items somehow remains constrained by the capacity of the knapsack. Leveraging this information, we proceed to construct the mathematical model for the Unbounded Knapsack Problem.

$$\max \sum_{j=1}^{n} v_j x_j \tag{25}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} w_j x_j \leq W \tag{26}$$

$$x_j \geq 0, \ \textit{integer} \qquad j = 1, ..., n \tag{27}$$

3.2.4 Multiple-Choice Knapsack Problem

The Multiple-Choice Knapsack Problem (MCKP) presents another notable variant of the Knapsack Problem (KP). As highlighted by Pisinger and Toth (1998), the MCKP involves the consideration of multiple classes, denoted as $N_i, ..., N_k$, consisting of items to be packed into a knapsack. Each item $j \in N_i$ possesses its own weight w_{ij} and value v_{ij}, while the knapsack has a limited capacity denoted as W. The objective of the MCKP is to optimally select one item from each class while respecting the knapsack's capacity, making it applicable in real-life scenarios such as production scheduling (Assi and Haraty, 2018). To capture these decisions, the MCKP employs the decision variable x_{ij}, which takes the value of 1 if item j is chosen from class N_i, and 0 otherwise. By leveraging the insights presented by Pisinger and Toth (1998), we establish the mathematical model for the Multiple-Choice Knapsack Problem.

$$\max \sum_{i=1}^{k} \sum_{j \in N_i} v_{ij} x_{ij} \tag{28}$$

$$\text{s.t.} \quad \sum_{i=1}^{k} w_j x_j \leq W \tag{29}$$

$$\sum_{j \in N_i} x_{ij} = 1 \qquad i = 1, ..., k \tag{30}$$

$$x_{ij} \in \{0, 1\} \qquad i = 1, ..., k,\ j \in N_i \tag{31}$$

3.2.5 Multiple Knapsack Problem

Lastly, we delve into the Multiple Knapsack Problem (MKP), which finds practical applications in various domains such as multiple processor scheduling, task allocation among autonomous agents, vehicle/container loading, loading liquids to tanks, and continuous double-call auctions (Assi and Haraty, 2018; Eilon and Christofides, 1971; Kalagnanam et al., 2001; Martello and Toth, 1990; Pisinger and Toth, 1998). In the MKP, we encounter a scenario where multiple knapsacks, denoted as m, are involved instead of a single one. Pisinger and Toth (1998) explicate that the main objective of the MKP is to effectively distribute n given items across the m knapsacks, each possessing distinct capacities denoted as W_i for $i = 1, ..., m$. Analogous to previous variants, each item j is characterized by its weight w_j and value v_j. Moreover, to address the selection process, we introduce a binary decision variable x_{ij}, which signifies whether item j is included in knapsack i or not. Assi and Haraty (2018) emphasize that the decision is not only whether a specific item should be included or not, but also to which bag it should be added to. Equipped with this information, we construct the mathematical model for the Multiple Knapsack Problem (MKP).

$$\max \sum_{i=1}^{m} \sum_{j=1}^{n} v_j x_{ij} \tag{32}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} w_j x_{ij} \leq W_i \qquad i = 1, ..., m \tag{33}$$

$$\sum_{i=1}^{m} x_{ij} \leq 1 \qquad j = 1, ..., n \tag{34}$$

$$x_{ij} \in \{0, 1\} \qquad i = 1, ..., m,\ j = 1, ..., n \tag{35}$$

3.2.6 Solvability

Similar to the Bin Packing Problem, the Knapsack Problem is widely recognized as an *NP-hard* (Non-deterministic Polynomial-time hard) problem, implying that an optimal solution cannot be obtained within polynomial time. Assi and Haraty (2018) present an illustrative example that highlights the inherent challenge of this problem. They explain that a straightforward approach to the problem involves exhaustively enumerating all possible combinations in the search space. However, such an enumeration would entail evaluating 2^n potential subsets, where n represents the total number of items. Consequently, if $n = 60$ and assuming that a computer can process one billion vectors in one second, the computation of 2^n vectors would require more than 30 years to complete.

Furthermore, the computational time grows exponentially with the addition of each item, making the execution of a brute-force algorithm highly impractical. Assuming that the number of items to be assigned to the bins corresponds to the current stock level currently present in the warehouse, a scenario with $n = 1.48$ million items would emerge. In this case, employing the same computational power as in the $n = 60$ example, solving the problem would necessitate approximately 7.8×10^{445507} (a 7.8 with 445,507 zeros) years. To provide a sense of scale, the age of the universe is estimated to be 13.8×10^9 (a 13.8 with *only* nine zeros) years. Thus, the optimal solution to our problem is clearly unattainable within a feasible timeframe.

3.3 Heuristics

Evans and Zanakis (1981) provide several justifications for the utilization of heuristics, highlighting circumstances in which they offer desirable and advantageous outcomes. One such circumstance arises when an exact method is available for problem-solving; however, the computational demands of this method make it impractical. As discussed in Sections 3.1 and 3.2, our problem falls into this category due to its classification as *NP-hard*. Given the computational complexity inherent in our problem, we turn our attention to exploring noteworthy heuristics employed in the Bin Packing Problem. We choose to look into heuristics of the *BPP* since the problem at hand relates closest to the bin packing problem as we also assign items to bins. However, it should be noted that some heuristics can be used for both the BPP and the *KP*. These heuristics serve as alternative approaches to solutions, offering practicality and efficiency in scenarios where exact solutions are unattainable within reasonable computational bounds.

3.3.1 Online versus Offline Algorithms

Algorithms can be classified into two distinct categories: online and offline. In the case of an online algorithm, we receive a sequence of requests and we perform an immediate action in response to each request (Hopper and Turton, 2001). Consequently, items are packed as soon as they arrive, without prior knowledge of future items. In contrast, offline algorithms receive the entire sequence of requests in advance, enabling them to make decisions based on the complete set of requests. Hopper and Turton (2001) elaborate that although an offline algorithm must respond to each request individually, it can consider the entire sequence to determine the optimal action for each request.

In the offline scenario, the exact set of items to be packed is known in advance. This knowledge allows for various optimization techniques, such as sorting the items based on specific characteristics. However, in the online case, where the arrival of future items remains uncertain, such sorting strategies are infeasible. In our problem domain, we lack a predefined set of items. Therefore, it is reasonable to consider employing an online algorithm. However, while we may not have precise information regarding the specific items, we do possess predictions about the size distribution of the incoming items. Consequently, one could argue that incorporating aspects of an offline algorithm becomes viable. For example, we have the option of generating items one by one, employing an online algorithm, or creating a list of items with known characteristics, which would allow for the utilization of offline algorithms.

3.3.2 First Fit

One of the prominent heuristics employed for addressing the Bin Packing Problem is the First Fit algorithm. To facilitate the understanding of this algorithm, we introduce two variables: a_i, representing the size of item i, and β_j, denoting the total size of all items packed in bin j. It is assumed that the capacity of all bins is standardized to 1. Xia and Tan (2010) provide a concise description of the First Fit algorithm, stating that "When we are packing a_i, we place it in the lowest indexed bin whose current content does not exceed 1 - a_i. Otherwise, we start a new bin with a_i as its first item."

Thus, by examining the first bin and evaluating the condition $\beta_j + a_i \leq 1$, we determine whether the item can be accommodated within the bin. If the sum exceeds 1 ($\beta_j + a_i > 1$), it indicates that placing the item in the current bin would exceed its capacity. In such cases, the item is placed in the subsequent bin to ensure adherence to the bin's capacity limit. If there are no bins available with sufficient capacity, a new bin is created. Consequently, the algorithm always selects the first bin that can accommodate the volume of the current item.

To illustrate the First Fit algorithm, we refer to Rajagopal (2016) and present Figure 17 as an example. The item sizes arrive in the following order: 0.3, 0.6, 0.2, 0.1, 0.5, 0.7, 0.2, 0.4 and 0.1. The numbers that are not in brackets indicate the order of assignment, while the numbers in brackets show the item volume.

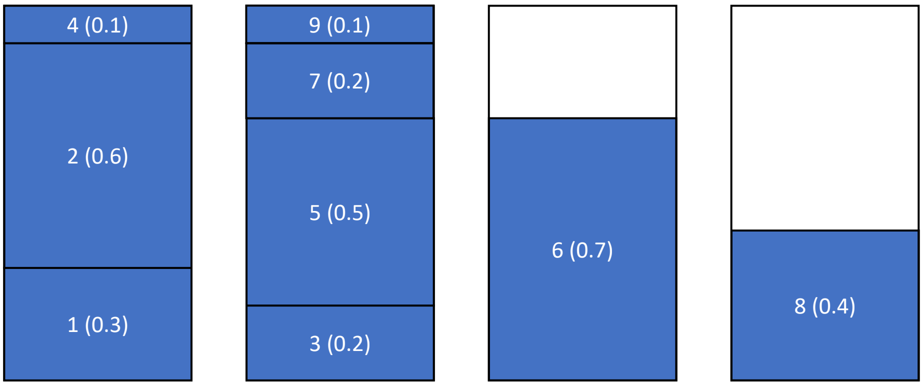


Figure 17: First Fit algorithm visualized. Item labels: packing order (item volume)

3.3.3 Next Fit

Another prominent algorithm in the context of the Bin Packing Problem is the Next Fit algorithm. It can be considered the most intuitive approach, as it aims to maximize the number of items packed in the current bin before initiating a new bin. Gourgand et al. (2014) provide a description of the Next Fit algorithm: "If the available space in the current bin is smaller than the size of the considered item, this bin is closed and a new bin is opened and becomes the new current bin".

The Next Fit algorithm shares some similarities with the First Fit algorithm as it also examines the condition $\beta_j + a_i \leq 1$. If this condition holds, the item is added to the current bin, similar to the First Fit algorithm. However, when $\beta_j + a_i > 1$, indicating that the current bin lacks sufficient space to accommodate the item, the algorithm proceeds to close the current bin and opens a new bin. Unlike the First Fit algorithm, items cannot be added to the previously closed bin at a later stage. The newly opened bin then assumes the role of the current bin.

Figure 18 serves as an illustrative example of the Next Fit algorithm. In this case, the item sizes arrive in the same order as previously: 0.3, 0.6, 0.2, 0.1, 0.5, 0.7, 0.2, 0.4 and 0.1.

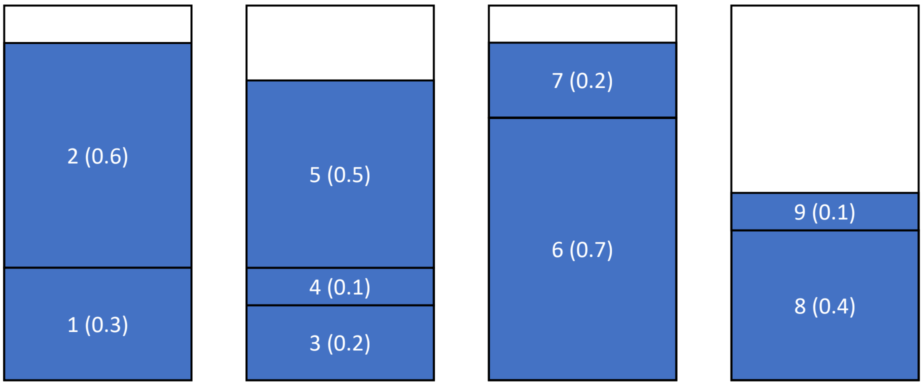


Figure 18: Next Fit algorithm visualized. Item labels: packing order (item volume)

3.3.4 Next-k Fit

The Next-k Fit algorithm is closely related to the Next Fit algorithm, with a slight modification to its approach. In this algorithm, instead of considering only the current bin, we now utilize a set of k possible bins to determine where to add items. Mao (1993) sheds light on the Next-k Fit algorithm, which operates as follows: starting with item a_1, we allocate it to bin B_1. When attempting to pack item a_i, we examine the last k non-empty bins. If a_i cannot fit into any of these bins, we open a new bin and assign a_i to it. Conversely, if there is sufficient space in any of the last k bins, a_i will be placed in the bin with the lowest index among these k possible bins.

It is worth noting that the Next-k Fit algorithm also bears a resemblance to the First Fit algorithm. However, a distinction lies in the fact that the Next-k Fit algorithm considers only a subset of k bins, while the First Fit algorithm encompasses the entire set of bins. For our Next-k Fit algorithm example, we employ the same set of items. In Figure 19, which serves as our illustrative example, we set $k = 2$, indicating that we solely focus on the last two active bins during the packing process.

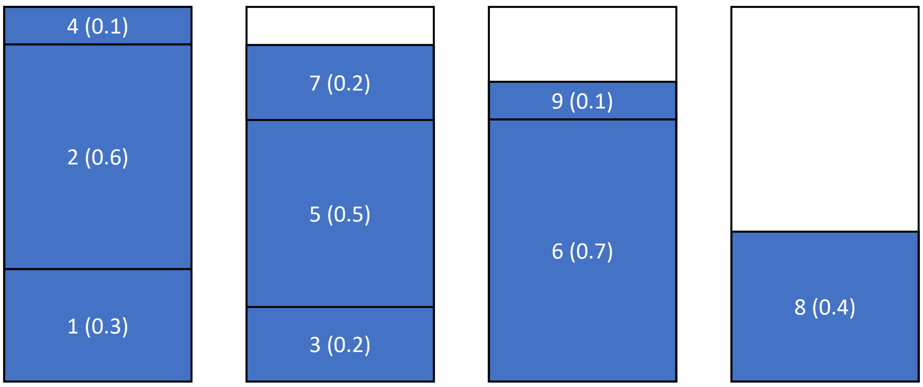


Figure 19: Next-2 Fit algorithm visualized. Item labels: packing order (item volume)

3.3.5 Best Fit

The Best Fit algorithm aims to minimize the remaining volume in a bin. Garey et al. (1972) provide an explanation of the algorithm as follows: "To consider a_i, find the bin B_j such that B_j is filled to a level $\beta_j \leq 1 - a_i$, and $\beta_j + a_i$ is as large as possible." This formulation indicates that the item should be placed in a bin where the sum of its current contents, β_j, together with the item's size, a_i, is maximized. Consequently, the algorithm strives to minimize the remaining space within the bin.

The Best Fit algorithm is widely recognized and renowned, possibly owing to its notable performance. Kenyon (1996) state that the Best Fit algorithm achieves results within a few percentage points of the optimal solution. Moreover, it is believed to be the most effective algorithm for the online Bin Packing Problem. Figure 20, inspired from the paper of Rajagopal (2016), provides an example illustrating the Best Fit algorithm. The item sizes are again received in the following order: 0.3, 0.6, 0.2, 0.1, 0.5, 0.7, 0.2, 0.4 and 0.1.

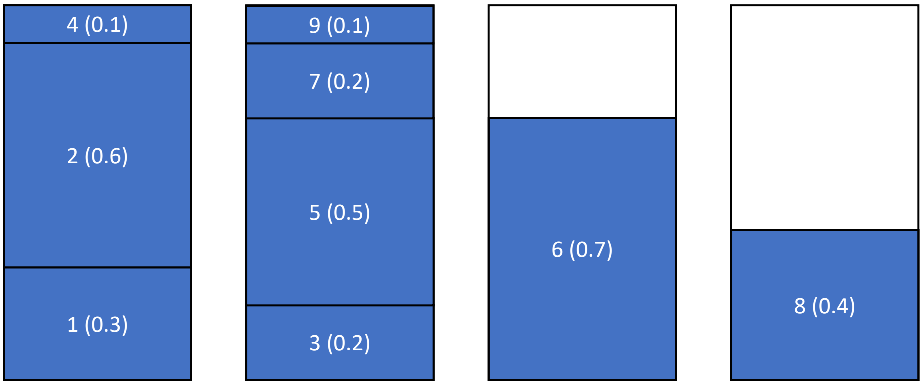


Figure 20: Best Fit algorithm visualized. Item labels: packing order (item volume)

3.3.6 Worst Fit

In contrast to the Best Fit algorithm, the Worst Fit algorithm operates by selecting the bin with the largest amount of free space, rather than the least remaining volume. Rajagopal (2016) outlines the functioning of the Worst Fit algorithm according to the following rule: "While trying to pack item a_i, the worst fit algorithm assigns the item to the bin whose empty space is maximum. If the item a_i is unable to fit in any of the opened bins, then a new bin is opened to pack that item a_i." Figure 21 presents an illustrative example of the Worst Fit algorithm, where the items are received in the same order as in the other examples.

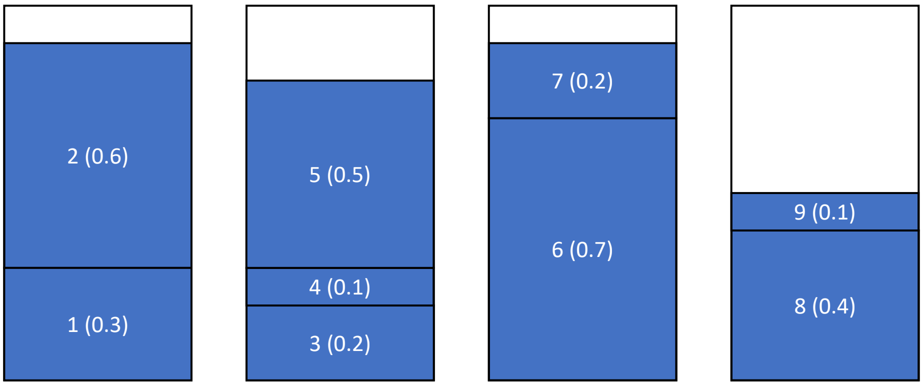


Figure 21: Worst Fit algorithm visualized. Item labels: packing order (item volume)

3.3.7 Harmonic-k Fit

Another algorithm widely employed is the Harmonic-k Fit algorithm, which involves partitioning the capacity-interval (0, 1] into k harmonic subintervals. These subintervals are established by dividing the interval into segments, where the first subinterval ranges from 0 to $\frac{1}{k}$, the second subinterval ranges from $\frac{1}{k}$ to $\frac{1}{k-1}$, and so forth. This yields the set of intervals $I^k = \{(0, \frac{1}{k}], (\frac{1}{k}, \frac{1}{k-1}], ..., (\frac{1}{2}, 1]\}$ for a chosen value of k. For instance, if $k = 4$, the intervals would be defined as $I^4 = \{(0, \frac{1}{4}], (\frac{1}{4}, \frac{1}{3}], (\frac{1}{3}, \frac{1}{2}], (\frac{1}{2}, 1]\}$.

According to Rajagopal (2016), items are classified and assigned to these subintervals based on their sizes. An item i is designated as an "I^m item" if its size falls within the interval $I^m = (\frac{1}{m+1}, \frac{1}{m}]$, where $m > 1$. Conversely, if the item size lies within the interval $I^k = (0, \frac{1}{k}]$, it is referred to as an "I^k item." Initially, k bins are opened to correspond to the k subintervals. When an item i belongs to the subinterval I^m, it is packed into the corresponding bin. Rajagopal (2016) further elucidates: "If that bin is filled and has not enough space to pack item I^m, then it is closed and a new bin is opened for that sub-interval."

An inherent limitation of this algorithm, as pointed out by Rajagopal (2016), is observed when packing items with sizes exceeding $\frac{1}{2}$. In such cases, each item necessitates its own dedicated bin, resulting in a significant waste of available space within each individual bin. This drawback undermines the overall efficiency of the algorithm. Figure 22 provides an explanatory example utilizing the aforementioned intervals, wherein the intervals are denoted as $Small = (0, \frac{1}{4}], Medium = (\frac{1}{4}, \frac{1}{3}], Large = (\frac{1}{3}, \frac{1}{2}], Huge = (\frac{1}{2}, 1]$. Moreover, the arrival sequence of items, characterized by their respective sizes, is again as follows: 0.3, 0.6, 0.2, 0.1, 0.5, 0.7, 0.2, 0.4 and 0.1.

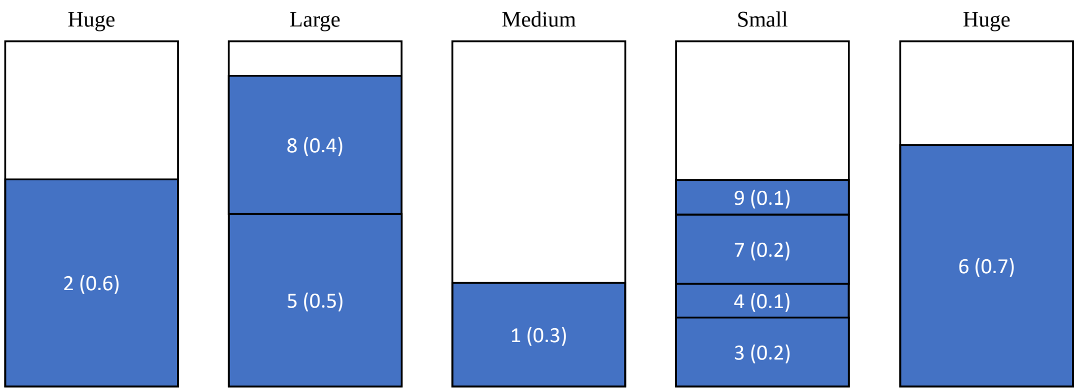


Figure 22: Harmonic-4 Fit algorithm visualized. Item labels: packing order (item volume)

3.4 Conclusion

Based on the insights from the literature, we assert that the problem at hand exhibits more resemblance with the bin packing problem than it does with the knapsack problem. This is mainly due to the characteristic of the bin packing problem, which entails the allocation of all items into bins, as opposed to the knapsack problem, wherein the decision revolves around whether or not to incorporate an item into the storage location. Consequently, we continue with the bin packing problem as the foundational model.

Regarding the heuristics, although the harmonic-k fit algorithm is well-known, its idea will not work for our solution approach. This arises from the fact that we are dealing with fixed item volumes per size group, which are not evenly spread over the bin capacities and thus a lot of items from different size groups will fall in the same interval, while other intervals are not used at all.

For clarification, we present a scenario employing the following volumes in litres for size groups 3XS to L: {0.35, 1.13, 1.17, 2.89, 7.60, and 25.15}. In this example, our focus centres on the shelf with the highest bin capacity of 151 litres, shelf 0. Taking into account a 35% fill rate, the available storage capacity for bins at this particular shelf level is 52.9 litres.

When setting the value of parameter k at four, the resulting intervals for this specific shelf tier would be $\{(0, 13.21], (13.2, 17.6], (17.6, 26.4], (26.4, 52.9]\}$. Upon examining the item volumes, we observe that items belonging to size groups 3XS, XXS, XS, S, and M are all encompassed within the first interval, while only size group L falls within an alternate interval, the third one. Moreover, the second and last intervals remain unused in this configuration.

Only when employing seven intervals or more, does size group M find itself in a separate interval, but at the cost of leaving four intervals unused. In order to also allocate size group S to a separate interval, we have to establish k at a value of 19, meaning that 15 intervals remain unused. These situations, wherein a substantial proportion of items are clustered within the same interval while other intervals are not used, take away the fundamental idea of the harmonic-k fit algorithm. Consequently, we have determined to exclude this heuristic from our solution approach.

The remaining heuristics are possible to use in our environment as they can all be used in their intended way. So to conclude, next to the random assignment of items to bins, we choose to evaluate the following heuristics: *First Fit, Next Fit, Next-k Fit, Best Fit* and *Worst Fit*. These heuristics will be evaluated both with and without the put-away logic of the operators in the warehouse.

4 Model for Company X

In Chapter 4, we explicate the components needed to construct a model specifically tailored for Company X. To establish a comprehensive model, it is essential to possess a clear understanding of its elements like the inputs, objective, and constraints. Section 4.1 describes all the input parameters needed in the model. Section 4.2 describes the objective of our model. Next to that, Section 4.3 shows us the constraints that need to be included in the model. Section 4.4 elaborates on the assumptions incorporated in the model. Lastly, Section 4.5 describes how the computer model works.

4.1 Inputs

We start by explaining the inputs incorporated in the formulation of the new model. In total, there are eight inputs that play an important role in our model.

4.1.1 Size Mix

As explicated in the preceding sections, the *size mix* is a significant input parameter. With this input, we ensure that the warehouse configuration within our model aligns with the anticipated size mix specified by Capacity Steering. By incorporating the size mix as an input, we establish a crucial linkage between the model's representation of the warehouse and the targeted size group distribution.

4.1.2 Item Volumes

Volumes corresponding to each size group play a role in the evaluation of the assignment of items to bins, allowing for the determination of whether an item belonging to a particular size group can feasibly fit within a designated bin. By considering the volumes of items and the capacities of bins, the model can effectively analyze the allocation of products.

4.1.3 Number of Bins and their Capacities

Moreover, we include information regarding the number of bins located on each shelf, as well as their capacities. By incorporating these parameters into our model, we build the physical infrastructure of the warehouse. This information is needed to accurately assess the feasibility of item allocation, as it directly impacts the potential utilization and capacity constraints of the available storage space.

4.1.4 Current State of the Warehouse

Furthermore, in order to enhance the precision of the model, we incorporate the current state of the warehouse. By considering the allocation of items to bins as currently observed in the warehouse, the model reflects the current state of the warehouse. This practice ensures that the model uses the existing items in the warehouse as the initial starting point, thereby enhancing the model's precision.

4.1.5 Shelving Size Mix

The inclusion of the shelving size mix serves as another input within the model. This input plays a role in our logical approach by incorporating the existing assignment of size groups to shelf levels as it is in the warehouse. By considering the distribution of a size group across the different shelves, the model determines the probability that a specific size group is on a specific shelf level. Using this information, the model chooses a specific shelf level with a certain probability. This way, we get more realistic and accurate simulations of storage allocation strategies as we include the logic of the put-away operators.

4.1.6 Fill Rate and Occupancy Rate

Finally, we should underscore the significance of two input parameters: the *fill rate* and the *occupancy rate*. The fill rate plays a crucial role in the determination of the compatibility between an item and a designated storage compartment. To make use of the fill rate, we compute the product of the storage bin's total volume and the fill rate, resulting in the maximum capacity available for a given bin. In circumstances where the combined volume of the items presently occupying the bin, together with the volume of the item selected for addition, exceeds this computed maximum capacity, the addition of the new item to the bin becomes infeasible.

Regarding the occupancy rate, this input parameter determines how many bins can be used at most for the allocation of items. In the event that a shelf level accommodates a total of 100,000 bins and we utilize an occupancy rate of 85%, it becomes evident that the feasible storage allocation is restricted to 85,000 bins.

4.2 Objective

Drawing upon the information found in the preceding sections, we can establish the objective of the model, which can be formulated as maximising the number of items successfully accommodated within the specified set of bins while concurrently ensuring adherence to all the incorporated constraints. However, it should be mentioned that the primary objective of the optimization process does not solely focus on maximizing the count of stored items, as more items being stored does not inherently guarantee the selection of the best method. Instead, the ideal method should aim to approximate the true *stockmax* of the warehouse, as reflected in historical data. By considering historical data, the goal is to identify a packing configuration that aligns closely with the observed stockmaxes over time.

Regarding the objective, we introduce the binary variable x_{iskb}, which equals 1 if item i ($i = 1, ..., N_s$) from size group s ($s = 1, ..., 6$, where 1 = 3XS, 2 = XXS etc.) is assigned to shelf level k ($k = 0, ...5$) and bin b ($b = 1, ..., B_k$). The maximization of the sum of x_{iskb} serves as the central goal in our mathematical model. By quantifying the total number of items assigned to the bins, we establish a quantitative measure that we can use to examine the performance of the model. In a mathematical manner, the objective of our model looks as follows:

$$\max \sum_{b=1}^{B_k} \sum_{k=0}^{5} \sum_{s=1}^{6} \sum_{i=1}^{N_s} x_{iskb} \tag{36}$$

4.3 Constraints

The incorporation of constraints is an integral aspect of our mathematical modelling framework. By defining and imposing constraints, we ensure the adherence of the model to specific conditions, limitations, and requirements that are present in the problem domain of Company X. These constraints play a crucial role in shaping the behaviour and outcomes of the model, aligning them with the desired objectives and capturing the challenges of the real-world context.

4.3.1 Fill Rate

Our first constraint ensures items cannot be added to a bin that has reached its maximum *fill rate*. For this constraint, we introduce the parameter v_s, denoting the volume of products belonging to size group s. Furthermore, we incorporate the parameter V_k, representing the capacity of a bin located on shelf level k as well as FR, denoting the desired fill rate. Also, by imposing this constraint, we enforce the prohibition of placing items with volumes larger than the capacity of a bin. This constraint ensures that only items compatible with the bin's capacity are considered.

$$\sum_{s=1}^{6}\sum_{i=1}^{N_s}(v_s \cdot x_{iskb}) \leq V_k \cdot FR \qquad b = 1, ..., B_k,\ k = 0, ..., 5 \tag{37}$$

$$x_{iskb} \in \{0, 1\} \qquad i = 1, ..., N_s,\ s = 1, ..., 6,\ k = 0, ..., 5,\ b = 1, ..., B_k \tag{38}$$

4.3.2 Occupancy Rate

Next to the fill rate, we also have to stay in line with the desired *occupancy rate*. For this, we introduce the parameter OR, which will be used to indicate the desired occupancy rate. This constraint is used for every shelf level individually to ensure that we do not accept that we can place items in all bins on the lowest shelf, and only exclude the bins on the top shelf. By including this constraint for every shelf level, we exclude the last bins on each shelf individually for the assignment of items to bins. This way, we prevent having an average occupancy rate below our threshold while all large storage locations on the lowest shelf are filled, and thus having an OR higher than our threshold on one specific shelf, which is not desirable. For this constraint, we add the binary variable y_{bk}, which is 1 if a bin b on shelf level k is filled with at least one item and 0 otherwise. It is good to note that M is a large number, which should be larger than the number of items that are in the specific bin.

$$y_{kb} \cdot M \geq \sum_{s=1}^{6}\sum_{1=1}^{N_s} x_{iskb} \qquad k = 0, ..., 5, b = 1, ..., B_k \tag{39}$$

$$\sum_{b=1}^{B_k} y_{kb} \leq B_k \cdot OR \qquad k = 0, ..., 5 \tag{40}$$

$$y_{kb} \in \{0, 1\} \qquad k = 0, ..., 5,\ b = 1, ..., B_k \tag{41}$$

4.3.3 Single Bin Allocation

To enforce the allocation of items to a single bin, we impose the following constraint. For each item in a size group, we compute the sum of its decision variable across all storage locations. This sum must be less than or equal to one. Since the decision variable is a binary variable, an item can only be assigned to one storage location, or to none, meaning that the value of the summation should either be 0 or 1.

$$\sum_{k=0}^{5} \sum_{b=1}^{B_k} x_{iskb} \leq 1 \qquad i = 1, ..., N_s, \;\; s = 1, ..., 6 \tag{42}$$

4.3.4 Size Mix

Furthermore, the incorporation of the *size mix* constraint is essential. This constraint ensures that the proportions of each size group considered in our calculation align with the expected shares determined by Capacity Steering. To enforce this constraint, we introduce the parameter FSM_s, denoting the Forecasted Size Mix of size group s. In the constraint, we count the number of items of a specific size group and divide it over the total number of items that are added to the warehouse to calculate the share of this specific size group. This fraction should be the same as the forecasted size mix as determined by Capacity Steering. Thus, it is important to note that equality is required. This requirement is formally expressed through the following constraint.

$$\frac{\sum_{b=1}^{B_k} \sum_{k=0}^{5} \sum_{i=1}^{N_s} x_{iskb}}{\sum_{b=1}^{B_k} \sum_{k=0}^{5} \sum_{s=1}^{6} \sum_{i=1}^{N_s} x_{iskb}} = FSM_s \qquad s = 1, ..., 6 \tag{43}$$

4.3.5 Shelving Size Mix

In the case of the logical assignment of items to shelves, the incorporation of put-away logic becomes necessary. This constraint involves the existing assignment of size groups to shelves. To represent this logic, we introduce the parameter $ASSM_{sk}$ (Actual Shelving Size Mix), which signifies the current distribution of a size group over the shelf levels. In this constraint, we count the number of items of a specific size group on a specific shelf level, and take its fraction compared to the total number of items of that size group over all shelves. Making sure this fraction equals $ASSM_{sk}$ ensures that our model incorporates the put-away logic implemented by warehouse operators. This constraint can be expressed mathematically as follows.

$$\frac{\sum_{b=1}^{B_k} \sum_{i=1}^{N_s} x_{iskb}}{\sum_{b=1}^{B_k} \sum_{k=0}^{5} \sum_{i=1}^{N_s} x_{iskb}} = ASSM_{sk} \qquad s = 1, ..., 6, k = 0, ..., 5 \tag{44}$$

4.4 Assumptions

In addition to the constraints elucidated in Section 4.3, we incorporate a set of assumptions that will facilitate an optimal problem-solving approach. These assumptions serve as foundational principles upon which our problem-solving methodology relies. Thus, in this section, we explicate the assumptions necessary for our approach.

4.4.1 Size Groups

As outlined in Section 1.1, the warehouse handles items classified within size groups ranging from 3XS to L. However, our analysis, as presented in Section 1.4, revealed that certain products exist beyond the specified size groups or lack any size group assignments altogether. For the purpose of our model, we limit our consideration solely to size groups spanning from 3XS to L, thereby excluding those products falling outside this range.

4.4.2 Item Shapes

An additional assumption to be incorporated into the model relates to the insignificance of item shapes. It is assumed that the fitting of items into storage locations is solely determined by the summation of item volumes, whereby the combined volume of items must not exceed the capacity of the respective storage location multiplied by the *fill rate*. This assumption is grounded in the company's utilization of a fill rate not exceeding 35%, indicating that sufficient space remains available within storage locations to accommodate multiple irregularly shaped items. Furthermore, the inclusion of this assumption is necessitated by the absence of specific information regarding the precise items to be delivered to the warehouse. As a result, there is also no knowledge available about the item shapes, meaning that even if we want to, we cannot include item shapes in the model.

4.4.3 Distribution Boxes

Another assumption we incorporate is the uniform capacity of distribution boxes at shelf level three. Although the external boxes share identical dimensions, the individual storage locations within them may vary in volume due to the presence of either two or three columns. Additionally, some boxes feature four rows while others have five rows. Consequently, despite the uniformity of external box dimensions, the size of individual storage locations within a distribution box may differ. Also, the available data employed by the company does not account for distinctions among the various-sized storage locations within distribution boxes.

4.4.4 Same Shelf Storage Volumes

We posit an additional assumption regarding the equivalence of storage volumes for locations on the same shelf. While some storage locations within the warehouse may deviate from the norm in terms of volume, either larger or smaller, these instances represent exceptions. Given the immense scale of the warehouse, encompassing approximately 1.5 million storage locations, it becomes unfeasible to ascertain the precise volume of each individual storage location. Consequently, the assumption is made that the storage volumes across locations on the same shelf are uniformly consistent, with the aforementioned exceptions accounting for a minor proportion of the total storage space.

4.5 Computer Model

We developed a computer model to test the heuristics from Section 3.3 to calculate the stockmax. Section 4.5.1 delineates the procedural instructions for users of the Stockmax Model pertaining to the input screen, while Section 4.5.2 provides an explanation of the model's operational mechanics.

4.5.1 Input Screen of the Computer Model

To enhance the accessibility of the Stockmax Model for the employees of Company X, the user interface has been deliberately designed to maintain a straightforward entry screen, as depicted in Figure 23.

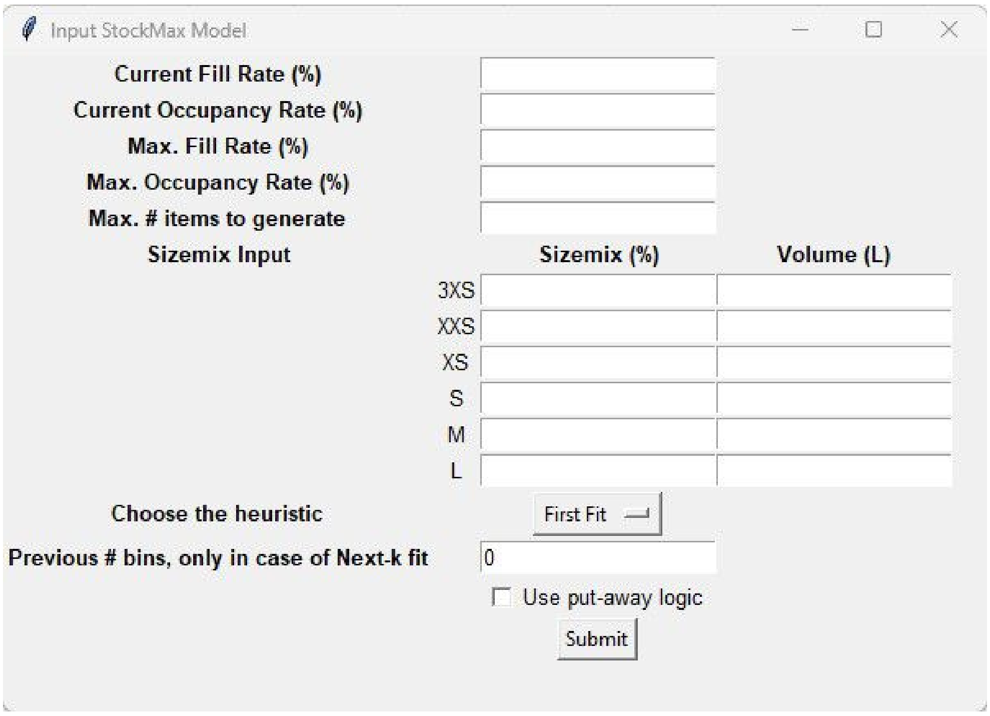


Figure 23: Entry Window of the Stockmax Model

Current Situation

The initial step in employing the Stockmax Model involves downloading a data table from a designated online dashboard, which will be automatically processed by the computer model. This data table furnishes crucial information regarding the quantities of items belonging to specific size groups, situated on designated shelf levels within the warehouse at the present time. Subsequently, the user needs to fill in the current fill rate and occupancy rate values. Leveraging the given fill rate and occupancy rate data, the model undertakes the process of assigning these items to bins on the specified shelf, thereby simulating the current state of the warehouse.

Maximum Fill Rate, Occupancy Rate and Item Generation

Upon inputting the current fill rate and occupancy rate, the following step entails determining the maximum fill rate and occupancy rate. Currently, these values stand at 35% and 85%, respectively. However, the warehouse operations team retains the flexibility to modify these parameters within the entry window, should the need arise.

Additionally, another factor in this entry process is specifying the maximum number of items to be generated. This number essentially corresponds to the length of the item list generated, representing the maximum number of items to be allocated to the bins. Thus, it is imperative for this specified value to exceed the difference between the current number of items present in the warehouse and the anticipated stockmax value. Although the stockmax is the desired outcome of our calculation and is not initially known, we should set a value substantially higher to ensure that we significantly surpass the number of items we will allocate to a bin.

Size Mix and Heuristic Choice

Moreover, we include the size mix, as ascertained by Capacity Steering. The user of the Stockmax Model simply fills in the shares of each size group along with the corresponding volumes of the distinct size groups. These volumes can easily be retrieved by an online dashboard in the company's software. Subsequently, the selection of the preferred heuristic is needed. Should the *Next-k Fit* algorithm be the chosen heuristic, the value of parameter k is to be specified.

Put-away Logic

Furthermore, an additional option is given to employ the put-away logic from the operators in the warehouse. The implementation of the put-away logic involves utilizing the probabilities associated with a specific size group being situated on a specific shelf level. This enables the model to make informed decisions regarding the appropriate shelf level for an item's placement.

Let us clarify this shelf determination process with an example. Let us assume the selected item for inclusion is a 3XS item. Table 8 shows us the number of items in size group 3XS on each shelf level. Based on this distribution, we construct an interval for each shelf level. After these intervals have been constructed, we generate a random number between 0 and 1. Let us assume this random number has a value of 0.56. This would mean that the random number falls in the interval of shelf level three, and thus, the focus is directed solely on this particular shelf level for selecting a bin.

Table 8: Example of shelf level interval determination for size group 3XS

Shelf Level	#Items	Share	Interval
0	5,000	1%	[0; 0,01]
1	10,000	2%	(0,01; 0,03]
2	100,000	20%	(0,03; 0,23]
3	325,000	65%	(0,23; 0,88]
4	50,000	10%	(0,88; 0,98]
5	10,000	2%	(0,98; 1]
Total	**500,000**	**100%**	**[0; 1]**

4.5.2 Pseudocode of the Computer Model

The following part explains the mechanics of the Stockmax Model through the utilization of pseudocode. This pseudocode serves as a tool for comprehending the fundamental functions of the model. The pseudocode, shown in Algorithm 1, is delineated into three separate segments: the first encompasses the retrieval of inputs from the input screen, the second encompasses the implementation of the current situation, and the third entails the steps required to execute the heuristic algorithm.

It is important to note that in the heuristic segment of the algorithm, for every heuristic, an assessment of available bin locations on a given shelf is conducted to ensure adherence to the prescribed maximum occupancy rate. In a scenario featuring a shelf comprising a total of 100,000 bins and a designated maximum occupancy rate of 85%, the heuristic would limit item allocation to the first 85,000 bins, while the remaining 15,000 bins would be excluded from consideration for this purpose.

Furthermore, we should emphasize that every heuristic also incorporates an evaluation of the maximum fill rate for each bin. This entails a check to determine whether the to-be-added item can be accommodated within a given bin. This determination is made by summing the volumes of the items already situated within the bin and subsequently adding the volume of the item under consideration. In the event that this cumulative volume surpasses the maximum allowable volume for the bin, computed as the bin's total volume multiplied by the maximum fill rate, the item is deemed incompatible with the bin in question. Consequently, an alternative bin must be sought for its allocation.

Should the situation arise wherein no bin can be identified to accommodate the item, the heuristic is terminated, signifying that we reached our Stockmax. Moreover, we keep track of the quantity of items successfully assigned to a bin through the heuristic. This quantity is added to the total number of items previously added to the warehouse when we tried to mimic its current situation. The summation of items already within the warehouse and the items added by the heuristic completes the calculation of the Stockmax value.

4.5.3 Termination Threshold Values

Preliminary tests of the random heuristic yielded unfavourable outcomes, characterized by the inclusion of a mere five to ten additional items on top of the items currently in the warehouse. In response, a decision was made to acknowledge a threshold of up to 10,000 items that are unfeasible for incorporation into the storage bins. This implies that the heuristic is not terminated until we have reached 10,000 items that we cannot store in a storage location. With this concession, we include 40,000 to 45,000 more items to be added to the warehouse. In view of the fact that a subset of 10,000 items, predominantly encompassing objects of medium and large sizes, is concurrently bypassed, the heuristic in question fails to align with the intended heuristic objective.

For all other heuristics, we set the threshold value to 1, which implies that the heuristic is terminated if there is 1 item that does not fit in a storage location.

Algorithm 1 Pseudocode of the Stockmax Model

Inputs

```
retrieve current FR from Input Screen (IS)
retrieve current OR from IS
retrieve max FR from IS
retrieve max OR from IS
retrieve #items to generate from IS
retrieve size mix data and item volumes from IS
retrieve which heuristic is selected from IS
if heuristic = Next-k Fit then
    retrieve the value of k from IS
end if
check if put-away logic is selected in IS
```

Current Situation

```
load in data file about current situation in warehouse
add the items currently in the warehouse to bins on the corresponding shelf
store the #items already in the warehouse
```

Heuristics

```
generate the list of items, adhering to the size mix
for i = 1 to #items on list do
    select item i
    if heuristic = Random then
        filter which shelves the item could fit on
        select random shelf level from filtered list
        select random bin number
    else if put-away logic is not selected then
        if heuristic = First Fit then
            start looping from the first shelf
            start looping from the first bin
            select the first bin that the item fits
        else if heuristic = Next Fit then
            start looping from the first shelf
            start looping from the last used bin on that shelf
            select the first bin that the item fits
        else if heuristic = Next-k Fit then
            start looping from the first shelf
            start looping from the last used bin minus k on that shelf
            select the first bin that the item fits
        else if heuristic = Best Fit then
            loop over all shelves
            loop over all bins
            select the bin with the least free space in which the item still fits
        else if heuristic = Worst Fit then
            loop over all shelves
            loop over all bins
            select the bin with the most free space
        end if
```

Algorithm 1 Pseudocode of the Stockmax Model (continued)

```
    else if put-away logic is selected then
        based on current item-to-shelf assignment, make an interval for each shelf
        generate a random number between 0 and 1
        check in which interval the random number falls, select that shelf
        if heuristic = First Fit then
            start looping from the first bin on the selected shelf
            select the first bin that the item fits
        else if heuristic = Next Fit then
            start looping from the last used bin on the selected shelf
            select the first bin that the item fits
        else if heuristic = Next-k Fit then
            start looping from the last used bin minus k on the selected shelf
            select the first bin that the item fits
        else if heuristic = Best Fit then
            loop over all bins on the selected shelf
            select the bin with the least free space in which the item still fits
        else if heuristic = Worst Fit then
            loop over all bins on the selected shelf
            select the bin with the most free space
        end if
    end if
    if item fits in the selected bin then
        add the item to the selected bin
        update the number of items that are added by the heuristic
    else
        stop the for loop
    end if
end for
```

Output

```
Stockmax = #items already in warehouse + #items added by heuristic
```

5 Solution Choice

In Section 5.1, we elucidate the criteria that underpin our selection of a suitable solution. In Section 5.2, we clarify the inputs that will be used for the experiments. Section 5.3 reveals the outputs of the experiments. After the outputs of the model have been collected, we determine the best solution in Section 5.4. Once the solution has been chosen, we evaluate it with the current method in Section 5.5.

5.1 Solution Criteria

To determine the best solution, it is imperative to evaluate the extent to which our model-computed *stockmax* aligns with the actual stockmax. To determine the actual stockmax, we identify instances where the company transitions into or out of an orange scenario, as depicted in Figure 10. Nine distinct points in time are observed where such transitions occur, and the corresponding stock levels at these points are considered indicative of the actual stockmax. The stock levels at these specific instances, reflecting the moments when the company enters or exits an orange scenario, are presented in Table 9.

Table 9: Actual stock level when entering/exiting orange scenario, considered indicative of the actual stockmax value

Date	Stock Level
28/01/2021	3,064,220
16/08/2021	2,964,663
15/10/2021	3,145,350
05/12/2021	2,929,828
27/12/2021	2,842,127
26/01/2022	2,932,930
13/06/2022	2,339,683
17/09/2022	2,274,383
24/12/2022	2,084,491

To assess the quality of a solution, a hard constraint is imposed, stipulating that the computation of the stockmax must be accomplished within a maximum timeframe of one hour. Consequently, should any of the algorithms exceed this threshold, they will be automatically disregarded as a viable solution.

Furthermore, we perform ten iterations of the stockmax calculation for each heuristic, for each date. The reason for running ten iterations is that the generated list of items is in random order, and thus the calculated stockmax can be different if we run the heuristic multiple times. Although no large deviations are expected between the outcomes of the stockmax value for a single date, it is still interesting to see the performance over multiple runs.

Subsequently, the average value derived from these ten runs is employed as the calculated stockmax. Of course, the heuristic yielding a calculated stockmax that exhibits a closer proximity to the actual stockmax is deemed better than a heuristic which shows greater disparity with the actual stockmax. In order to assess the disparities between the computed stockmax and the actual stockmax values, we employ the calculation of the mean deviation. However, it is important to acknowledge that the utilization of the mean deviation carries a limitation. In a scenario where, for instance, the computed stockmax exceeds the actual stockmax by 500,000 items on one occasion and falls short of it by the same quantity on another, the resulting mean deviation will yield a value of zero. Nevertheless, the utilization of the mean deviation affords us the opportunity to check whether a heuristic consistently exhibits an overestimation or an underestimation of the stockmax.

To gain deeper insights into the disparities between the computed stockmax and the actual stockmax, we also incorporate the utilization of the absolute deviation. In the 500,000-item deviation scenario, the absolute mean deviation would be established at 500,000 units, whereas the mean deviation was zero units. The absolute deviation offers better insights into the divergence from the actual stockmax value, albeit it does not provide insights regarding the overestimation or underestimation.

In the scenario where multiple viable solutions are present, our focus extends to the evaluation of the standard deviation in the computed values of the 10 iterations for each selected date. Of course, there exists a favour in calculations that are characterized by their consistency in their outcomes. Consequently, we prefer a heuristic with a low standard deviation over a heuristic with a high standard deviation.

5.2 Experimental Settings

Additionally, in our analysis, it is essential to use the inputs that were employed during the dates identified in Table 9. In the following section, we explicate the different inputs that will be utilized for the experiments. We also give an overview of the values of these inputs.

5.2.1 Overview of Experimental Inputs

Table 10 enumerates the inputs for the evaluation process. It is worth noting that the size mix from Capacity Steering is not available for all the dates stated in Table 9. This is the case for the first transition into an orange scenario, on the 28$^{\text{th}}$ of January 2021. As there is no information about the anticipated size mix available, this hinders the testing of our model's stockmax estimation for this specific date, so this date has been excluded. For the remaining dates, we do have data regarding the size mix. This data was collected by the Capacity Steering team and subsequently distributed to the Warehouse Operations team for integrating the data into this research.

The residual data inputs were acquired through a dedicated dashboard integrated into the organization's software. After selecting the correct date in the dashboard, we can retrieve two critical metrics, the *fill rate* and *occupancy rate*. Additionally, another dashboard enables the retrieval of average volumes associated with each size group for the specified date instances.

In addition to the tabulated data, we fill in the maximum fill rate and maximum occupancy rate as well. Currently, these values have been set at 35% and 85%, respectively.

For the Next-k Fit, we have to choose a specific value for k. Instead of choosing one value, we selected a range of values, encompassing the set $\{1, 3, 5, 10\}$. Initial tests indicated great potential for the designated values of k.

5.2.2 Current Warehouse Situation

In addition to the inputs expressed in Table 10, another requirement is the inclusion of the present state of the warehouse. Access to the dataset is readily facilitated through the company's software. This dataset contains precise information about the quantity of items within a defined size group that are located on a specific shelf. Employing this dataset, the model will assign the quantity of items corresponding to the size group onto the specified shelf. For this procedure, the model selects a random bin on the shelf to assign the item to.

5.2.3 Put-Away Logic

With the exception of the random heuristic, a supplementary logical variant is incorporated in the tests. This variant encompasses the procedural aspects of the allocation of items to shelf locations by operators. Facilitated by a dataset, the likelihood of encountering a specific size category upon a designated shelf level can be easily determined. How this logic is incorporated in the model is explained in Section 4.5.1.

Table 10: Input parameters needed for testing

Date	Occupancy Rate	Fill Rate		3XS	XXS	XS	S	M	L
16/08/2021	84.96%	26.00%	Size mix	15%	5%	21%	25%	25%	9%
			Volume (L)	0.34	1.15	1.1	2.62	7.16	24.75
15/10/2021	85.19%	26.75%	Size mix	15%	5%	21%	25%	25%	9%
			Volume (L)	0.33	1.09	1.11	2.79	7.1	24.33
05/12/2021	84.60%	26.09%	Size mix	13%	5%	21%	30%	24%	7%
			Volume (L)	0.31	1.1	1.07	2.79	7.23	24.41
27/12/2021	84.21%	25.83%	Size mix	13%	5%	21%	30%	24%	7%
			Volume (L)	0.31	1.11	1.07	2.73	7.25	25.22
26/01/2022	86.23%	28.10%	Size mix	14%	5%	21%	25%	26%	10%
			Volume (L)	0.33	1.1	1.06	2.6	7.34	25.06
13/06/2022	84.68%	24.95%	Size mix	12%	5%	20%	23%	27%	13%
			Volume (L)	0.34	1.14	1.21	2.84	7.39	25.22
17/09/2022	85.48%	28.02%	Size mix	11%	6%	19%	22%	28%	14%
			Volume (L)	0.38	1.16	1.21	2.84	7.39	25.58
24/12/2022	84.90%	29.48%	Size mix	16%	7%	19%	23%	23%	12%
			Volume (L)	0.37	1.1	1.21	3.07	8.18	25.45

5.3 Results

In the following section, we elucidate the experimental outputs that were obtained by using the inputs as described in Section 5.2. Using the model outputs shown in this section, we choose the best solution in Section 5.4.

5.3.1 Computational Time

Upon executing the heuristics, it became evident that the computational effort involved in computing the stockmax utilizing the best fit and worst fit heuristics is extraordinary. The computational time for a single iteration of these heuristics approximated six hours, prompting our decision to omit these heuristics from our tests. Consequently, it can be concluded that the best fit and worst fit heuristics will not be selected as viable solutions.

For the remaining heuristics, a series of ten iterative runs is executed for each heuristic, across every distinct date, utilizing the corresponding input configurations delineated in Section 5.2. One criterion stipulated by the organization necessitated that the model is capable of calculating the stockmax value within a time frame of less than sixty minutes. Table 11 exhibits the computational times, which delineates the mean duration for the execution of a single iteration. It is undeniable that all considered heuristics clearly adhere to the one-hour threshold. Ergo, there exists no imperative to exclude any other heuristic from our deliberations pertaining to solution determination.

Table 11: Average computational time per iteration in seconds (and minutes)

Heuristic	*Withouth Logic*	*With Logic*
First Fit	55	1148 (19.1 min.)
Next Fit	56	925 (15.4 min.)
Next-1 Fit	53	1172 (19.5 min.)
Next-3 Fit	54	1098 (18.3 min.)
Next-5 Fit	55	1057 (17.6 min.)
Next-10 Fit	67	1085 (18.1 min.)
Random	46	-

5.3.2 Average Difference

Appendices A and B present the stockmaxes calculated by the heuristics, computed as the mean over a span of the ten iterations for each specific date. Within Table 12, a comprehensive overview is provided of the disparities between the calculated stockmaxes and the actual stockmaxes (as determined in Table 9), averaged across the eight examined dates.

Directing our attention towards the heuristics that do not incorporate the operational put-away logic, we see that the Next Fit heuristic gives noteworthy outcomes, as evidenced by an average disparity of only -2,773 items in relation to the actual stockmax. This discrepancy implies that, on average, the heuristic approximates the stockmax at a value 2,773 items lower than the actual stockmax.

Upon examining the outcomes of the heuristics that include the put-away logic, we make an interesting observation. Namely, these heuristics manifest stockmax values that are lower by approximately 100,000 items when compared to their counterparts without the put-away logic. Given the decrement of the stockmax by a magnitude of 100,000 items, we see that the outcomes are much more aligned with the actual stockmax. This observation underscores the performance of the inclusion of the put-away logic. From the heuristics that incorporate the put-away logic, we see that all of them show relatively great results, except for the Next Fit heuristic.

Table 12: Average differences with actual stockmax over the 8 dates

Heuristic	*Without Logic*	*With Logic*
First Fit	121,334	16,243
Next Fit	-2,773	-97,868
Next-1 Fit	69,128	-32,692
Next-3 Fit	103,057	4,570
Next-5 Fit	110,797	11,769
Next-10 Fit	115,439	13,672
Random	-520,956	-

Upon checking the average differences, a preliminary deduction can be drawn that highlights one particular heuristic attaining great results in the absence of put-away logic, namely the Next Fit heuristic. Meanwhile, five distinct heuristics exhibit noteworthy performance upon their integration of put-away logic. These are: First Fit, Next-1 Fit, Next-3 Fit, Next-5 Fit and Next-10 Fit. The remaining heuristics are not deemed as possible solutions as their deviation from the actual stockmax is considered too high in relation to the previously mentioned heuristics.

5.3.3 Average Absolute Difference

Given the existence of multiple plausible solutions, we extend the analysis to encompass the average absolute differences. This serves to mitigate the potential misrepresentation that may arise from average deviations. For instance, a heuristic may yield a stockmax computation exceeding the actual stockmax by 500,000 items on a given date, yet, conversely, render a calculation that falls short by an equivalent magnitude on another date. When averaged, this would yield a deviation of zero, implying great precision, despite the substantial variability. In contrast, another heuristic might yield a surplus of 20,000 items on one date and 10,000 items on another, thus having an average deviation of 15,000 items. While this might seem less favourable in comparison to the former scenario, it in fact signifies a more desirable outcome. By employing the concept of absolute differences, the aforementioned scenarios would yield an average difference of 500,000 items for the former and 15,000 items for the latter. This procedure permits a more extended evaluation, wherein the second scenario emerges as the more optimal choice. This approach is displayed in Table 13, depicting the computed average absolute differences for the selected well-performing heuristics.

Table 13: Average absolute differences with actual stockmax over the 8 dates of well-performing heuristics

Heuristic	Average Absolute Difference
Next Fit without logic	172,134
First Fit with logic	162,686
Next-1 Fit with logic	144,517
Next-3 Fit with logic	154,797
Next-5 Fit with logic	159,260
Next-10 Fit with logic	161,199

Upon analyzing the findings presented in Table 13, a notable observation emerges. Specifically, the heuristic that previously occupied the apex position in Table 12, denoting superior performance, has descended to showing the worst performance in Table 13. This underscores the proposition that a straightforward arithmetic mean as a performance measure can give an illusion of great performance.

Furthermore, it is evident that the Next-1 Fit heuristic exhibits the least absolute deviation in relation to the actual stockmax values. However, upon examination of Table 12, it becomes apparent that this very heuristic had previously assumed the least favourable position among the set of chosen heuristics considered to be well-performing.

Upon an examination of the heuristic that ranks second in Table 13, it becomes evident that the Next-3 Fit heuristic exhibits a greater absolute deviation when contrasted with the Next-1 Fit heuristic. Nonetheless, it remains superior to the remainder of the heuristics featured within the compilation. Furthermore, upon examination of this heuristic's performance in Table 12, it emerges that the Next-3 Fit heuristic again secures the second most favourable ranking among the entire set of tested heuristics. Given these insights, the Next-3 Fit algorithm seems like a strong contender for being chosen as an adequate solution.

5.3.4 Standard Deviation

As delineated within Section 5.1, in the presence of multiple plausible solutions, we direct our focus to the investigation of the standard deviation of the outcomes of the iterations. Considering the standard deviations of the six well-performing solutions that were identified based on the data provided in Table 12, a pattern emerges. Specifically, for the heuristic lacking put-away logic, the standard deviation approximates 1,900 items, as evidenced in Table 14. Conversely, the well-performing heuristics that incorporate the put-away logic exhibit standard deviations averaging around 4,000 items. From these findings, one could posit that the utilization of put-away logic doubles the standard deviation for the heuristics, thus not using the put-away logic would seem favourable. To get a better understanding of the severity of the standard deviation, we incorporate the Coefficient of Variation in Section 5.3.5.

Table 14: Standard Deviations (St. Dev.) and Coefficients of Variation (CoV)

Heuristic	St. Dev.	CoV
Next Fit without logic	1,914	0.07%
First Fit with logic	4,160	0.15%
Next-1 Fit with logic	3,660	0.14%
Next-3 Fit with logic	4,162	0.15%
Next-5 Fit with logic	3,885	0.14%
Next-10 Fit with logic	4,053	0.15%

5.3.5 Coefficient of Variation

An added layer of analysis through the application of the coefficient of variation (CoV) serves to illuminate a more nuanced perspective. The CoV, which signifies the standard deviation relative to the mean of the outcomes, reveals a proportionality where, unsurprisingly, the CoV values for the heuristics employing put-away logic are approximately twice those of the Next Fit without the put-away logic. Nonetheless, it is imperative to acknowledge that a CoV value lower than 10% is widely recognized as an indication of excellent performance. Upon inspection of Table 14, we see all recorded coefficients of variation reside significantly below the 10% threshold, substantiating the proposition that the difference in standard deviation observed between the Next Fit heuristic and its counterparts is negligible. Evidently, this convergence of CoV values towards 0 underscores the performance exhibited by the entire ensemble of heuristics.

Due to the exceptional performance across the spectrum of heuristics, the consideration of standard deviation is deemed superfluous. Given the fact that the CoV values are well below the 10% mark, it is warranted to assert that each heuristic demonstrates excellent outcomes with regard to the variability among different iterations' results. In light of this finding, the standard deviation shall be omitted as a criterion for evaluation.

5.4 Solution Choice

To choose one solution, we utilize the well-known Multi-Criteria Decision-Making (MCDM) method on the six algorithms that show great potential to be the chosen solution. As mentioned in Section 5.3.5, the standard deviation is excluded in the decision making as the standard deviation for each heuristic shows excellent results.

5.4.1 Normalizing

First off, we normalize the scores such that all scores have a value between zero and one, where the worst score receives zero points and the best score receives one point. This is done by using Formula 45. It is good to note that for the normalization process of the average difference values, we need to take the absolute value.

$$X_{normalized} = \frac{X_i - X_{max}}{X_{min} - X_{max}} \tag{45}$$

Using Formula 45 together with the data we find in Tables 11, 12 and 13, we find the following normalized values for the six algorithms in Table 15.

Table 15: Normalized Scores. Where C.T. = Computational Time, A.D. = Average Difference. A.A.D. = Average Absolute Difference.

Heuristic	**C.T.**	**A.D.**	**A.A.D.**
Next Fit without logic	1	1	0
First Fit with logic	0.02	0.55	0.34
Next-1 Fit with logic	0	0	1
Next-3 Fit with logic	0.07	0.94	0.63
Next-5 Fit with logic	0.10	0.70	0.47
Next-10 Fit with logic	0.08	0.64	0.40

5.4.2 Criteria Weights

In the process of determining the optimal solution through the Multi-Criteria Decision-Making (MCDM) framework, the determination of criterion weights is a necessary step. Given that the computational times for all considered heuristics fall below the designated one-hour threshold, we have assigned a weight of 1 to this criterion. It is worth noting that, since all solutions fall below this threshold, one could contemplate omitting this criterion from the evaluation. However, we still have a preference for a fast heuristic over its slower counterparts.

Subsequently, we are left with two criteria: the Average Difference and the Average Absolute Difference. In light of the richer insights offered by the Average Absolute Difference concerning the proximity of outcomes to the stockmax values, we have allocated a higher weight to this criterion relative to the Average Difference. Consequently, the Average Difference criterion is given a weight of 2, whereas the Average Absolute Difference is deemed twice as important, and thus is assigned a weight of 4.

5.4.3 Chosen Solution

Finally, we multiply the weights of the criteria with the normalized scores. From this, we find the numbers shown in Table 16.

Table 16: Weighted Scores. Where C.T. = Computational Time, A.D. = Average Difference. A.A.D. = Average Absolute Difference.

Heuristic	**C.T.**	**A.D.**	**A.A.D.**	**Total Score**
Next Fit without logic	1	2	0	3.00
First Fit with logic	0.02	1.10	1.37	2.49
Next-1 Fit with logic	0	0	4	4.00
Next-3 Fit with logic	0.07	1.88	2.51	4.46
Next-5 Fit with logic	0.10	1.40	1.86	3.36
Next-10 Fit with logic	0.08	1.27	1.58	2.93

Through the employment of the Multi-Criteria Decision-Making (MCDM) approach, the identification of the best solution is facilitated by choosing the one with the highest score. In our comparison, the *Next-3 Fit with Logic* algorithm has emerged as the best solution, with a score of 4.46 points. Consequently, our MCDM procedure has determined this algorithm to be the superior choice within our research.

5.5 Comparison With Current Method

Having determined the best heuristic, we undertake an assessment of whether this heuristic outperforms the current method for stockmax determination. It is possible that the heuristic identified as the best-performing option may not necessarily exhibit superior performance in comparison to the current stockmax determination approach. Such a scenario would make the heuristic useless.

5.5.1 Current Method Performance

To facilitate an evaluation of our proposed solution in relation to the current method, it is necessary to first ascertain the accuracy of the current approach. This assessment is shown in Table 17, wherein the discrepancy between the stockmax values determined by the Warehouse Operations (WO) team and the actual stockmax is presented. The actual stockmax values correspond to the stock levels observed at the moment of transitioning into or out of an orange scenario (see Table 9).

Table 17: Warehouse Operations Stockmax and Actual Stockmax

Date	**Stockmax WO**	**Actual Stockmax**	**Difference**
16/08/2021	3,900,000	2,964,663	935,337
15/10/2021	4,300,000	3,145,350	1,154,650
05/12/2021	4,300,000	2,929,828	1,370,172
27/12/2021	4,300,000	2,842,127	1,457,973
26/01/2022	3,500,000	2,932,930	567,070
13/06/2022	3,000,000	2,339,683	660,317
17/09/2022	2,600,000	2,274,383	325,617
24/12/2022	2,600,000	2,084,491	515,509
		Average Difference	**873,318**

From the data in Table 17, it is clear that the Warehouse Operations (WO) team computed a stockmax value that exhibited an average surplus of 873k items relative to the actual stockmax. Additionally, it can be inferred that the average absolute difference aligns with the average difference, given the consistent trend of the WO team consistently overestimating the stockmax in comparison to its actual value.

5.5.2 Comparison

Table 18 provides a comparative analysis of stockmax values as determined by the Warehouse Operations team in contrast to those derived through our computational model. An examination of this table reveals the superior performance of our solution, as evidenced by the noteworthy proximity of our model-derived stockmax values to the actual stockmax values, in comparison to those determined by the Warehouse Operations team.

Table 18: The Actual Stockmax, Stockmax as determined by Warehouse Operations (WO) and the Stockmax as determined by the model, together with the differences with the Actual Stockmax in brackets. A.D. = Average Difference. A.A.D. = Average Absolute Difference.

Date	Actual	WO (difference)	Next-3 Fit (difference)
16/08/2021	2,964,663	3,900,000 (935,337)	3,005,425 (40,762)
15/10/2021	3,145,350	4,300,000 (1,154,650)	3,114,122 (-31,228)
05/12/2021	2,929,828	4,300,000 (1,370,172)	3,225,030 (295,202)
27/12/2021	2,842,127	4,300,000 (1,457,973)	3,132,403 (290,276)
26/01/2022	2,932,930	3,500,000 (567,070)	2,944,158 (11,228)
13/06/2022	2,339,683	3,500,000 (660,317)	2,226,752 (-112,931)
17/09/2022	2,274,383	2,600,000 (325,617)	2,049,243 (-225,140)
24/12/2022	2,084,491	2,600,000 (515,509)	1,852,881 (-231,610)
	A.D.	**873,318**	**4,570**
	A.A.D.	**873,318**	**154,797**

It is worth acknowledging that our model does exhibit discrepancies, some of them in the range of 200-300k items on select dates. However, these disparities are overshadowed by the significantly larger disparities observed in the stockmax estimates generated by the Warehouse Operations team, which can occasionally reach up to a magnitude of over 1 million items more compared to our model's deviations. From the comparison in Table 18, we conclude that our model clearly outperforms the current method used by the Warehouse Operations team.

5.6 Actual Sizemix as Input

An additional insight into the performance of the *Next-3 Fit with Put-Away Logic* would be to examine the heuristic when utilizing a 100% accurate size mix. To operationalize this evaluation, the forecasted size mix from Capacity Steering is excluded, and instead, the actual stock size mix existing on the designated testing dates is employed (see Table 19). We should acknowledge that this experiment is unfeasible within an actual operational setting, as the actual size mix remains unknown prior to its realization.

Table 19: Actual stock size mix for the tested dates

Date	3XS	XXS	XS	S	M	L
16/08/2021	17%	7%	23%	24%	23%	6%
15/10/2021	18%	7%	22%	25%	22%	6%
05/12/2021	18%	7%	23%	23%	22%	7%
27/12/2021	18%	6%	22%	25%	23%	6%
26/01/2022	17%	6%	22%	25%	24%	6%
13/06/2022	14%	6%	19%	26%	27%	8%
17/09/2022	12%	5%	19%	25%	28%	11%
24/12/2022	13%	5%	16%	25%	27%	14%

Upon conducting the evaluation of the heuristic with the actual size mix, we present the findings in Table 20. Analyzing the results of the average deviations and average absolute deviations, we conclude that using a 100% accurate size mix does not yield an enhancement in the accuracy of our model.

In contrast to the utilization of the size mix derived from Capacity Steering that led to an average difference of only 4,570 items, we now observe a substantial increase in this discrepancy, with a value of 163,470 items when using the 100% accurate size mix. Furthermore, the average absolute deviation also experiences a clear increase, transitioning from 154,797 items with the Capacity Steering size mix to 267,333 items with the 100% accurate size mix. This indicates that the usage of an improved size mix does not contribute positively to the enhancement of model accuracy when using the next-3 fit heuristic; in fact, it appears to have a negative effect, resulting in decreased accuracy.

An explanation of this increased difference could be due to the fact that in the actual size mix, we see a larger proportion of smaller items when compared to the Capacity Steering size mix. Since smaller items of course take up less space, we can store more items in total, and thus our stockmax is at a much higher value compared to the Capacity Steering size mix.

Table 20: Stockmax values using 100% accurate size mix, compared with the actual stockmax. A.D. = Average Difference. A.A.D. = Average Absolute Difference.

Date	Actual Stockmax	Model Stockmax	Difference
16/08/2021	2,964,663	3,315,234	350,571
15/10/2021	3,145,350	3,405,892	260,542
05/12/2021	2,929,828	3,228,401	298,573
27/12/2021	2,842,127	3,259,218	417,091
26/01/2022	2,932,930	3,148,924	215,994
13/06/2022	2,339,683	2,520,123	180,440
17/09/2022	2,274,383	2,129,879	-144,504
24/12/2022	2,084,491	1,813,540	-270,951
		A.D.	**163,470**
		A.A.D.	**267,333**

Considering the computational time required by the algorithm, we observe an average duration of 1,290 seconds (21.5 minutes) per iteration. Although this computational time slightly exceeds the time of other heuristics that employed the Capacity Steering size mix (see Table 11), it remains comfortably within the predefined threshold of one hour.

Furthermore, we assess the standard deviation. In this regard, the average standard deviation over the eight tested dates is found to be 3,786, a value consistent with the standard deviations observed for the heuristics employing the put-away logic and utilizing the Capacity Steering size mix (see Table 14). Not surprisingly, the coefficient of variation registers at a low value, namely 0.13%. This coefficient of variation, being notably below the 10% threshold, underscores the excellent performance exhibited by the algorithm in terms of its stability and precision.

Should Capacity Steering substantially enhance the accuracy of the size mix, it would be beneficial to undertake a re-evaluation to determine the heuristic approach that offers optimal performance under such conditions. It is plausible that an alternative heuristic may outperform the *Next-3 Fit Heuristic with Put-Away Logic* when confronted with the utilization of a more accurate size mix.

6 Conclusion and Recommendations

In the concluding chapter, within Section 6.1, we summarize our research findings and derive conclusions. Following this, within Section 6.2, we present a set of recommendations, intended for the enhancement of the model and the optimization of overall warehouse operations. We present the limitations of this research in Section 6.3. At last, Section 6.4 describes possible ideas for further research.

6.1 Conclusion

The primary objective of this research was to come up with a method to enhance the accuracy of the *stockmax* calculation. For clarification, the stockmax is the maximum quantity of items the company seeks to maintain within the warehouse, this value is deliberately set lower than the warehouse's actual capacity, as the company accounts for a *fill rate* of 35% and an *occupancy rate* of 85%.

A more accurate stockmax calculation serves to improve operational efficiency within the warehouse, as there is a better understanding to teams about the stock levels at which critical threshold values for both fill rate and occupancy rate will be attained. When exceeding those threshold values, the Traffic Light Model comes into play, which describes the actions to be taken in order to mitigate the consequences. The Traffic Light Model uses four scenarios (green, orange, red and black) based on the value of the occupancy rate and fill rate, where the green scenario is the desired one. If the stockmax calculation is more accurate, the company will be in a green scenario more often. Based on the findings presented in Chapter 5, it is concluded that the *Next-3 Fit Heuristic with Put-Away Logic* exhibits superior performance for calculating the stockmax when compared to the alternative heuristics, as well as the current method.

Root Cause

Prior to arriving at this conclusion, the first step entailed a thorough investigation into the root cause of the operational challenges encountered by the company. This investigative phase involved the delineation of all issues pertaining to the operational issues faced by the company. Subsequently, it was concluded that the central issue underlying these operational anomalies pertained to the imprecision of the stockmax model. Consequently, the overarching objective of this research was to enhance the accuracy of the stockmax calculation, specifically within the shelving area of one of Company X's warehouses.

Model

Based on our literature review in search of a suitable solution, it has been concluded that the problem under investigation bears a notable resemblance to the well-known Bin Packing Problem. In light of this alignment, we have identified five Bin Packing heuristics that may be employed to address the specific issue encountered by Company X. These heuristics are assessed both with and without consideration of the operators' put-away strategies within the warehouse. In addition to these five heuristics, a random allocation strategy was included, wherein items are assigned to shelves and bins in a random manner.

Results

To determine which heuristic performs best in our situation, a bin packing computational model was developed to meet the needs of Company X. This model facilitates an examination of the various heuristics, thereby enabling a subsequent comparison between the heuristics. The experiments in Chapter 5 revealed that the *Next-3 Fit Heuristic with Put-Away Logic* yielded the most favourable outcomes, based on measures such as the average deviation and average absolute deviation in relation to the actual stockmax values. To determine the deviations from the actual stockmax values, we examine the eight dates at which the company transitioned into or out of an orange scenario as determined by the Traffic Light Model. The stock levels at these points in time are considered indicative of the actual stockmax value.

Furthermore, the assessment of heuristics is extended to their computational time. The company requires the stockmax calculations to be completed within a maximum timeframe of one hour. The *Next-3 Fit Heuristic with Put-Away Logic* notably adhered to this temporal threshold, with a mean computational time of less than 20 minutes per iteration.

Lastly, the standard deviations of the heuristics underwent examination. In this regard, it is notable that the standard deviations exhibited excellent performance for all heuristics, further underpinning its efficacy in addressing the stockmax calculation problem.

Comparison with Current Method

Upon determining the best-performing heuristic, a comparative analysis was undertaken against the current methodology. Notably, the current method within the Warehouse Operations team consistently yielded an overestimation of the stockmax, exhibiting a discrepancy of 873,000 items on average over the eight examined dates. This disparity consequently resulted in inflated fill rates and occupancy rates, while the stock levels remained substantially under the computed stockmax value.

In contrast, an evaluation of the *Next-3 Fit Heuristic with Put-Away Logic* revealed a significantly lower average disparity of 4,570 items when compared to the actual stockmax of the eight dates. Furthermore, considering the average absolute deviation, which in the case of the current method equated to the average deviation, yielded a value of 154,797 for the *Next-3 Fit Heuristic with Put-Away Logic*. These findings collectively underscore the superior performance of our proposed model in contrast to the current method.

Conclusion

By making use of the *Next-3 Fit Heuristic with Put-Away Logic* in our computer model, the Warehouse Operations team is able to get a better estimate of the true stockmax value of the warehouse. Using this more accurate estimation of the stockmax, it is expected that the time in a scenario other than the desired green state is minimized, and thereby, the operations in the warehouse run more smoothly and cost-efficient.

6.2 Recommendations

Next to recommending the use of the computer model, we formulate recommendations for Company X based on the findings from this research. These recommendations encompass not only enhancements specific to the computer model but also broader suggestions that may yield advantages for various warehouse operations within the same facility or potentially extend their applicability to warehouse operations across Company X's other warehouses.

6.2.1 Track Accuracy

To gain more insights into the accuracy of the computer model's estimations, it is advantageous to undertake an analysis of the differences between the computed stockmax and the actual stockmax, with the latter representing the inventory level at the point in time when the warehouse attains its maximum capacity. By documenting these disparities, it becomes possible to assess the model's accuracy within the operational environment. If patterns, such as consistent underestimations, come to the attention of the Warehouse Operations team, proactive measures can then be taken to enhance the model's accuracy.

6.2.2 Other Warehouses of Company X

Given Company X's operation across multiple warehouses, it could be of great potential to extend the computer model's implementation to these other locations. Currently, the Warehouse Operations teams at these various facilities adhere to the same methodology as the one observed within the context of this research. This study has revealed certain inaccuracies associated with this methodology for our specific warehouse. However, we still need to check if the same estimation problems with stock levels also exist in other Company X warehouses.

6.3 Limitations

Certainly, our research is not without limitations. The subsequent section elucidates the limitations within our research and their consequences within the context of our study.

6.3.1 Determination of the Actual Stockmax

The determination of the actual stockmax remains uncertain. In our research, we posit that the actual stockmax corresponds to the stock level at which we enter an orange scenario of the Traffic Light Model. Such a scenario predominantly occurs as a consequence of an increased occupancy rate. Our observations reveal that we consistently operate below the prescribed fill rate threshold. Consequently, an opportunity exists to increase the inventory within already occupied bins. Such an increment would serve to enhance the fill rate without increasing the occupancy rate. Consequently, this would enable the storage of additional items prior to the attainment of the actual stockmax threshold, and thus the actual stockmax is expected to be higher than the stock levels that were employed in this research.

6.3.2 Data Quality

As elucidated in Section 1.4, an issue surfaced, namely the presence of erroneous data pertaining to the items housed within the warehouse. Such discrepancies may arise as items in the warehouse are assigned to size groups that are incompatible with the operational parameters of the warehouse, or instances where items lack a size categorization altogether. From the data, we see that instances are located in storage locations in the warehouse which practically should be infeasible. As an example, we see items of the large size group, which have a volume of around 25 litres, that are located on shelf level three, which has bins with a capacity of at most five litres. This limits our representation of the current warehouse situation in our model, as we cannot assign items of 25 litres to bins with a capacity of five litres, and thus these items were left out in our modelled current situation.

6.4 Further Research

Our research has the potential for extension across diverse areas. The subsequent section enumerates plausible topics for further research. The exploration of these topics holds the promise of yielding an improved model, consequently enhancing the accuracy of the stockmax model.

6.4.1 Relation with Under/Overestimation

On average, our computer model demonstrates a relatively close approximation to the actual stockmax. However, on specific occasions, discrepancies can be substantial, with overestimations of up to 250,000 items against underestimations of an equivalent magnitude. To get a better understanding of these instances of over/underestimation, it is interesting to explore potential correlations. Specifically, an examination of how item volumes correlate with the over/underestimation of the stockmax. Furthermore, the size mix parameter represents another compelling input variable to be checked for correlation. It could prove advantageous to check whether the computer model tends to overestimate or underestimate the stockmax when confronted with scenarios characterized by either a substantial proportion of larger or smaller items.

6.4.2 Evaluate Threshold Values in the Traffic Light Model

As delineated in Section 6.3, it is expected that the true actual stockmax will surpass the value of the actual stockmax that we used in our research, as the fill rate falls short of reaching the prescribed threshold level. This implies that there exists potential to accommodate additional items within storage locations already occupied, thereby enhancing the fill rate without an increase in the occupancy rate. This observation raises questions about the threshold values in the Traffic Light Model. It may be necessary to deliberate on whether adjustments are needed, especially given the fact that an orange scenario is driven primarily by an elevated occupancy rate.

6.4.3 Size Mix Accuracy

The Capacity Steering size mix assumes an important role within our model. However, as researched in Section 2.2, its accuracy in relation to the actual stock size mix exhibits discrepancies. One explanation for this disparity lies in the distinction between the stock size mix and the size mix as provided by Capacity Steering, which pertains to the outbound size mix. Additionally, it is good to acknowledge that the retrieval of the most precise size mix data for all assessed dates encountered certain limitations. Given that the size mix gets updated from time to time, the latest iteration of the size mix is expected to be the most accurate. For the initial periods under examination, access to the most accurate size mix data was unavailable. Enhancement of the size mix accuracy could potentially improve the accuracy of the model as we use more precise input parameters.

However, as delineated in Section 5.6, no enhancement in the accuracy of the model was seen in the case of the *Next-3 Fit Heuristic with Logic*. In the event of a substantial improvement in size mix accuracy, we recommend a re-evaluation of the heuristics, employing the stock size mix observed in historical data, as opposed to conducting assessments using the size mix derived from Capacity Steering for these dates.

A Model Outcomes without Put-Away Logic

Table 21: Average calculated Stockmax without using put-away logic over 10 iterations

Date	Actual Stockmax	First Fit	Next Fit	Next-1 Fit	Next-3 Fit	Next-5 Fit	Next-10 Fit	Random
16/08/2021	2,964,663	3,015,810	2,887,133	2,971,929	3,003,171	3,010,663	3,015,085	2,370,292
	Difference	51,117	-77,530	7,266	38,508	46,000	50,422	-594,371
15/10/2021	3,145,350	3,149,786	3,049,844	3,108,622	3,139,373	3,144,217	3,150,885	2,574,256
	Difference	4,436	-95,506	-36,728	-5,977	-1,133	5,535	-571,094
05/12/2021	2,929,828	3,478,276	3,258,133	3,392,811	3,448,183	3,454,633	3,458,155	2,433,612
	Difference	548,448	328,305	462,983	518,355	524,805	528,327	-496,216
27/12/2021	2,842,127	3,410,282	3,191,268	3,326,315	3,375,107	3,386,663	3,390,135	2,344,778
	Difference	568,155	349,141	484,188	532,980	544,536	548,008	-497,349
26/01/2022	2,932,930	2,956,207	2,860,380	2,908,013	2,938,545	2,949,055	2,954,097	2,417,348
	Difference	23,277	-72,550	-24,917	5,615	16,125	21,167	-515,582
13/06/2022	2,339,683	2,417,404	2,335,789	2,378,272	2,403,762	2,411,325	2,415,469	1,800,631
	Difference	77,721	-3,894	38,589	64,079	71,642	75,786	-539,052
17/09/2022	2,274,383	2,105,448	2,037,385	2,070,325	2,092,974	2,098,842	2,103,442	1,791,322
	Difference	-168,935	-236,998	-204,058	-181,409	-175,541	-170,941	-483,061
24/12/2022	2,084,491	1,950,910	1,871,341	1,910,193	1,936,796	1,944,434	1,949,701	1,513,567
	Difference	-133,581	-213,150	-174,298	-147,695	-140,057	-134,790	-470,924
	Average difference	**121,334**	**-2,773**	**69,128**	**103,057**	**110,797**	**115,439**	**-520,956**

B Model Outcomes using Put-Away Logic

Table 22: Average calculated Stockmax using put-away logic over 10 iterations

Date	Actual Stockmax	First Fit	Next Fit	Next-1 Fit	Next-3 Fit	Next-5 Fit	Next-10 Fit
16/08/2021	2,964,663	3,009,099	2,896,512	2,971,250	3,005,425	3,009,980	3,005,658
	Difference	44,436	-68,151	6,587	40,762	45,317	43,995
15/10/2021	3,145,350	3,118,670	3,009,968	3,083,753	3,114,122	3,117,238	3,117,992
	Difference	-26,680	-135,382	-61,597	-31,228	-28,112	-27,358
05/12/2021	2,929,828	3,253,362	3,044,270	3,151,905	3,225,030	3,242,486	3,251,638
	Difference	323,534	114,442	222,077	295,202	312,658	321,810
27/12/2021	2,842,127	3,161,287	2,953,849	3,060,761	3,132,403	3,144,419	3,148,228
	Difference	319,160	111,722	218,634	290,276	302,292	306,101
26/01/2022	2,932,930	2,961,514	2,828,015	2,897,105	2,944,158	2,956,780	2,960,326
	Difference	28,584	-104,915	-35,825	11,228	23,850	27,396
13/06/2022	2,339,683	2,235,627	2,159,068	2,200,166	2,226,752	2,232,404	2,234,660
	Difference	-104,056	-180,615	-139,517	-112,931	-107,279	-105,023
17/09/2022	2,274,383	2,050,989	2,015,655	2,039,703	2,049,243	2,050,571	2,048,717
	Difference	-223,394	-258,728	-234,680	-225,140	-223,812	-225,666
24/12/2022	2,084,491	1,852,850	1,823,176	1,847,274	1,852,881	1,853,732	1,852,252
	Difference	-231,641	-261,315	-237,217	-231,610	-230,759	-232,239
	Average difference	**16,243**	**-97,868**	**-32,692**	**4570**	**11,769**	**13,627**

References

Abdel-Basset, M., Manogaran, G., Abdel-Fatah, L., & Mirjalili, S. (2018). An improved nature inspired meta-heuristic algorithm for 1-d bin packing problems. *Personal and Ubiquitous Computing*, *22*, 1117–1132.

Ackerman, K. B. (1997). Making warehousing more efficient. In *Practical handbook of warehousing* (pp. 297–306). Springer US.

Ali, I. M., Essam, D., & Kasmarik, K. (2021). Novel binary differential evolution algorithm for knapsack problems. *Information Sciences*, *542*, 177–194.

Assi, M., & Haraty, R. A. (2018). A survey of the knapsack problem. *2018 International Arab Conference on Information Technology (ACIT)*, 1–6.

Baruffaldi, G., Accorsi, R., Manzini, R., & Ferrari, E. (2020). Warehousing process performance improvement: A tailored framework for 3pl. *Business Process Management Journal*, *26*(6), 1619–1641.

Chan, F., & Chan, H. (2011). Improving the productivity of order-picking of a manual-pick and multi-level rack distribution warehouse through the implementation of class-based storage. *Expert Systems with Applications*, *38*(3), 2686–2700.

Cosgrave, J. (1997). Estimating the capacity of warehouses. *Disasters*, *21*(2), 155–165.

Eilon, S., & Christofides, N. (1971). The loading problem. *Management Science*, *17*(5), 259–268.

Evans, J., & Zanakis, S. (1981). Heuristic optimization: Why, when and how to use it. *Interfaces*, *11*(5), 84–91.

Faber, N., De Koster, M., & Smidts, A. (2013). Organizing warehouse management. *International Journal of Operations & Production Management*, *33*(9), 1230–1256.

Frazelle, E. (2001). *World-class warehousing and material handling*. McGraw Hill LLC. https://books.google.nl/books?id=mK%5C_Pf9DkOX0C

Garey, M. R., Graham, R. L., & Ullman, J. D. (1972). Worst-case analysis of memory allocation algorithms. *Proceedings of the fourth annual ACM symposium on Theory of computing*, 143–150.

Gourgand, M., Grangeon, N., & Klement, N. (2014). An analogy between bin packing problem and permutation problem: A new encoding scheme. *Advances in Production Management Systems. Innovative and Knowledge-Based Production Management in a Global-Local World: IFIP WG 5.7 International Conference, APMS 2014, Ajaccio, France, September 20-24, 2014, Proceedings, Part I*, 572–579.

Heerkens, H., & van Winden, A. (2017). *Systematisch managementproblemen oplossen.* Noordhoff Uitgevers.

Hifi, M., Kacem, I., Nègre, S., & Wu, L. (2010). A linear programming approach for the three-dimensional bin-packing problem. *Electronic Notes in Discrete Mathematics*, *36*, 993–1000.

Hopper, E., & Turton, B. C. (2001). An empirical investigation of meta-heuristic and heuristic algorithms for a 2d packing problem. *European Journal of Operational Research*, *128*(1), 34–57.

Kalagnanam, J., Davenport, A., & Lee, H. (2001). Computational aspects of clearing continuous call double auctions with assignment constraints and indivisible demand. *Electronic Commerce Research*, *1*, 221–238.

Kenyon, C. (1996). Best-fit bin-packing with random order. *SODA*, *96*, 359–364.

Lewis, H. R. (1983). Michael r. Πgarey and david s. johnson. computers and intractability. a guide to the theory of np-completeness. wh freeman and company, san francisco1979, x+ 338 pp. *The Journal of Symbolic Logic*, *48*(2), 498–500.

Maarouf, W. F., Barbar, A. M., & Owayjan, M. J. (2008). A new heuristic algorithm for the 3d bin packing problem. *Innovations and Advanced Techniques in Systems, Computing Sciences and Software Engineering*, 342–345.

Mann, Z. A. (2017). The top eight misconceptions about np-hardness. *Computer*, *50*(5), 72–79.

Mao, W. (1993). Tight worst-case performance bounds for next-k-fit bin packing. *SIAM Journal on Computing*, *22*(1), 46–56.

Martello, S., & Toth, P. (1990). *Knapsack problems: Algorithms and computer implementations*. John Wiley & Sons, Inc.

Park, B.-C. (2011). Order picking performance: Strategies, issues, and measures. *Journal of Korean Institute of Industrial Engineers*, *37*(4), 271–278.

Pisinger, D., & Toth, P. (1998). Knapsack problems. *Handbook of Combinatorial Optimization*, 299–428.

Rajagopal, K. (2016). *Simulation of online bin packing in practice* (Doctoral dissertation). University of Nevada, Las Vegas.

Sutterer, L. B. (2010). *Development of a tool for forecasting a warehouse facility footprint and enabling rapid scenario analysis* (Doctoral dissertation). Massachusetts Institute of Technology.

Sweep, S. (2003). Three dimensional bin-packing issues and solutions. *University of Minnesota*.

Xia, B., & Tan, Z. (2010). Tighter bounds of the first fit algorithm for the bin-packing problem. *Discrete Applied Mathematics*, *158*(15), 1668–1675.