U.PORTO
FEP FACULDADE DE ECONOMIA
UNIVERSIDADE DO PORTO

Optimization of the Order-Picking Process in Warehouses

António Ricardo Baptista de Almeida

Dissertation
Master in Data Analytics

Supervised by
Jorge Miguel Silva Valente

2025

Abstract

Order-picking is one of the most labor-intensive and costly operations in warehousing. This dissertation seeks to develop a data-driven approach to improve the efficiency of this operation. Using product characteristics, historical order data, and warehouse operational constraints, a model was developed to determine the storage position for each product that minimizes the overall long-term picking cost. The model is operationalized through two use cases also developed in this project: Use Case 1, which recommends storage positions for incoming products, and Use Case 2, which suggests rearrangements of products already stored into more suitable locations. To assess the impact of the new storage policy on productivity, historical orders were simulated. The simulation results show that the policy suggested by the model reduces the average order picking time by 24.86% compared to the current practice. Furthermore, this methodology bridges the gap between theoretical optimization and practical implementation through actionable tools – the Use Cases – that can be seamlessly integrated into existing warehouse management systems.

Keywords: Warehouse, order-picking optimization, storage allocation

Resumo

O *picking* de encomendas é uma das operações mais trabalhosas e com maiores custos em armazéns. Esta dissertação procura desenvolver uma abordagem baseada em dados para melhorar a eficiência desta operação. Utilizando as características dos produtos, os dados históricos de encomendas e as restrições operacionais do armazém, foi desenvolvido um modelo para determinar a posição de armazenamento de cada produto que minimize o custo global de *picking* a longo prazo. O modelo é operacionalizado através de dois casos de uso também desenvolvidos neste projeto: o Caso de Uso 1, que recomenda posições de armazenamento para os produtos recebidos, e o Caso de Uso 2, que sugere a rearrumação de produtos já armazenados para locais mais adequados. Para avaliar o impacto da nova política de armazenamento na produtividade, foram simuladas encomendas históricas. Os resultados da simulação mostram que a política sugerida pelo modelo reduz em 24,86% o tempo médio de *picking* por encomenda, em comparação com a prática atual. Além disso, esta metodologia faz a ponte entre a otimização teórica e a implementação prática através de ferramentas acionáveis – os Casos de Uso – que podem ser integradas de forma fluida nos sistemas de gestão de armazém existentes.

Palavras-chave: Armazém, otimização de *picking*, alocação de armazenamento

Table of Contents

1 Introduction and Motivation 1
2 Problem Description 3
3 Literature Review 7
3.1 Supply Chain and Logistics 7
3.2 The Warehouse 9
3.3 Design of Storage and Order-Picking Processes 15
3.4 Final Considerations on the Literature Review 23
4 Methodology 24
4.1 Layout Mapping 24
4.2 COI Calculation 27
4.3 Scenario Generation 33
4.4 Simulation 45
4.5 Use Cases 53
5 Results 67
5.1 Simulation Data 67
5.2 Comparative Results Across Scenarios 69
5.3 Impact of the Use Cases 72
6 Discussion 76
6.1 Ideal Storage Plan 76
6.2 Use Cases Discussion 78
7 Limitations and Future Work 81
8 Conclusion 83
References 85

List of Tables

Table 1 - Numeric example of positions needed for product j 30
Table 2 - Numerical example of real protection period for product j 32
Table 3 - Statistical summary of the results of each scenario 71

List of Figures

Figure 1 - Warehouse layout map 4
Figure 2 – Representation of routeing strategies. Reprinted from "An evaluation of order picking routeing policies", by C. G. Petersen, 1997, International Journal of Operations & Production Management, 17(11), pp. 1098–1111 21
Figure 3 - Cumulative distribution of pickings per cumulative percentage of products (ranked by frequency) 35
Figure 4 - Representation of the return routing policy. Reprinted from "An evaluation of order picking routeing policies", by C. G. Petersen, 1997, International Journal of Operations & Production Management, 17(11), pp. 1098–1111 49
Figure 5 - Weekly order volume over the past 12 months 68
Figure 6 - Percentage distribution of the number of distinct products per order 69
Figure 7 - Boxplots of average picking time per order for each scenario (100 repetitions per scenario) 70
Figure 8 - Stacked bar chart with decomposition in replenishment and pure picking time for each scenario 72
Figure 9 - Projected evolution of layout efficiency under exclusive application of UC1 73
Figure 10 - Relationship between cumulative execution time of UC2 suggestions and layout efficiency improvement 74

1 Introduction and Motivation

Logistics and supply chain management are concepts that have been gaining popularity in public debate in recent years. The growing concern with these themes is due to consumers' recognition of the impact that supply chain problems have on their daily lives and the vulnerability of the supply chain has been exposed by events such as the pandemic, wars, and other disruptive events (Lara & Wassick, 2023).

Warehouses are one of the fundamental elements of the supply chain, as they allow decoupling the moment of production from the moment of product consumption. Companies thus place considerable emphasis on improving warehouse operations (Boysen & de Koster, 2025). The picking operation, which retrieves the products necessary to respond to orders, is the most important operation in a warehouse because it is the most labor-intensive and is directly related to the service level provided (Hackman & Bartholdi, 2019). It is therefore the priority operation to optimize when seeking to improve overall efficiency.

Aligned with this reality, this dissertation seeks to develop a data-based approach to optimize the picking process in a logistics company's warehouse. Although the focus is on improving picking, related operations such as storage are also included in the project scope, since there is great interdependence, as the way products are stored will determine future picking costs.

This topic has been the subject of research by various authors over time, as will be evident in the literature review. However, it is a complex problem whose solution depends on many factors such as the characteristics of the products to be stored, the characteristics of the orders, and the operational constraints of the warehouse. Thus, off-the-shelf approaches are not adequate, and it is necessary to develop solutions adapted to the specific context found, which is what will be done in this project.

The remainder of this dissertation is organized as follows. Chapter 2 provides a detailed description of the problem and the specific characteristics of the warehouse under study. Chapter 3 presents the literature review, introducing relevant concepts and discussing state-of-the-art approaches to warehouse storage and picking optimization. Chapter 4 describes the methodology developed, including the rationale behind the proposed model and its implementation. Chapter 5 reports the results of the simulation experiments, and

these results are discussed in Chapter 6. After that, Chapter 7 outlines the limitations of the study and proposes avenues for future research. Finally, Chapter 8 concludes the dissertation by reviewing the overall process and consolidating the main conclusions.

2 Problem Description

Understanding the problem context and the current reality of the warehouse where the project will be implemented is fundamental to ensuring that the developed methodology is appropriate.

This warehouse operates as a third-party logistics (3PL) provider. This means that it does not own any of the products it handles. Instead, it offers logistics services to client companies, which contract the warehouse to receive, store, and ship their products on demand. When a client places an order, the warehouse is responsible for preparing and dispatching the requested products to the delivery addresses specified by the client. This model is commonly used by retail chains or franchising companies that need to distribute goods to multiple points of sale.

The products arrive over time and must be received and stored until they are needed for an order. Traditionally, products are received in larger volumes than those in which they are shipped; for example, a pallet full of a product can be received that will then allow to respond to several orders of smaller volumes.

The warehouse infrastructure is based on a standard racking system commonly used in the logistics industry, where products are stored in metal racks with five vertical levels, allowing goods to be stacked vertically to optimize the use of available space. These racks are arranged into 12 parallel aisles, creating corridors through which warehouse operators can move to store or retrieve products.

Figure 1 displays the macro-layout of the warehouse. It is possible to observe that the reception and expedition areas coincide (EXP), and that the corridors allow for movement in two directions, as they have only one entrance.

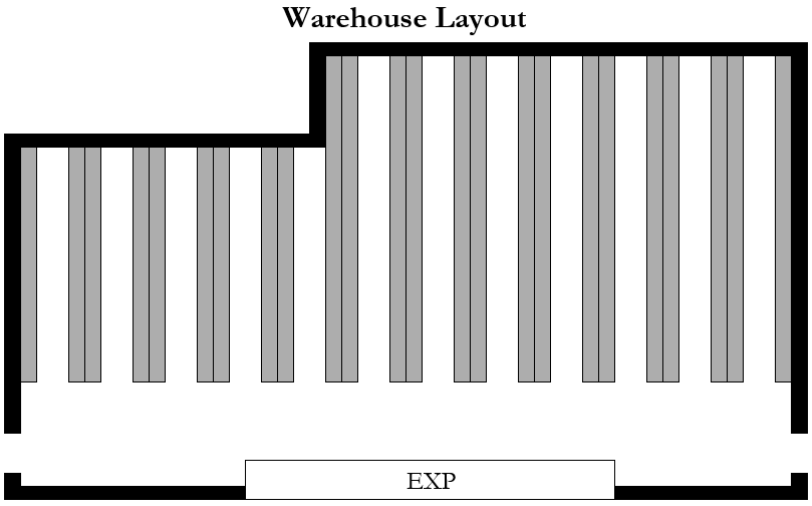


Figure 1 - Warehouse layout map

Another relevant aspect of the warehouse operation concerns the picking process for products with expiration dates, which represent the majority of items handled. In these cases, the First Expire First Out (FEFO) criterion is used, meaning that when an order requests an item, units with the shortest expiration date in the warehouse are picked first. This criterion is managed by the Warehouse Management System (WMS), which during the order picking process guides the operator to each position to retrieve items for the order, showing the position where units with the shortest validity date are located. Due to this policy, there is an operational constraint not to place units of the same item with different expiration dates in the same position, as this coexistence could lead to errors during picking or, at least, would require the picker to spend more time identifying which units should be collected for the order in question.

The mentioned constraint has a significant impact on storage since one way to guarantee it would be to place in one position only the units of a product with the same expiration date. However, the quantity of each combination is often much less than the capacity of a position, and this approach would waste immense storage space in each position. To ensure this constraint while trying to maximize available space usage, the warehouse uses another approach, allowing different products to be placed in each position, since with different products the risk of the picker collecting the wrong units is very low, and better utilizing position capacity. This method gives the storage policy in this warehouse a distinctive characteristic: storage positions are multi-product, potentially having a high number of different SKUs if their physical characteristics and quantities allow it.

Currently, after being received, products are stored by warehouse workers in a position of their choice, with the only defined policy being to place products from the same client in the same aisle, but allocation within that aisle is completely left to the discretion and intuition of the person doing the storage. This unsystematic approach means that the time and distance traveled when these products need to be retrieved are not optimized, hindering long-term operational efficiency.

This optimization consists of strategically storing products based on their characteristics, so that the average time required to prepare orders is minimized. For example, it would be preferable for the most requested products, those that appear in many orders, to be placed closer to the shipping dock, since these positions will be accessed frequently, and shorter travel distances can significantly reduce total picking time. However, this problem is complex because there are multiple criteria beyond product turnover that must be considered, such as the physical characteristics of the products. For instance, placing large-volume items in the most accessible locations may result in these positions being filled quickly with only a few units, leading to a suboptimal use of high-value storage space due to the low number of units stored. In addition, operational and physical warehouse constraints must also be taken into account. These include, for example, the need not to put food products and toxic materials in the same position, or the need to prevent fragile items such as glass bottles, or heavy pallets, from being stored on the upper levels of the racks for safety reasons.

Thus, the objective is to develop a model that weighs the relevant criteria to establish a systematic, data-based product allocation policy to minimize future order preparation time. This model should be able to indicate for each product received the ideal storage position in the warehouse. In practice, workers would only need to use a barcode reading device to scan an incoming product, and the model would return the position where the product should be placed, on that same device.

Additionally, the model should also be capable of generating recommendations for rearranging products that are already stored. These recommendations will be crucial in an initial phase to bring the existing storage closer to optimal storage. In the long term, this functionality allows accommodating turnover variations, for example, in products with highly seasonal demand.

To enable the development of this project, three categories of data were made available. First, historical orders, covering the last two years, which provide information on

the date of preparation and detailed content of each order, including the products and quantities requested. Second, historical stock data, documenting the products and quantities stored in each warehouse position on a daily basis over the same two-year period. Finally, warehouse physical data, including the warehouse layout, the arrangement and dimensions of aisles and storage positions, and relevant physical movement restrictions.

To identify the best storage policy and evaluate its impact, historical orders will be used to simulate the current policy, which serve as the baseline scenario. Subsequently, the same historical orders will be simulated under alternative storage policies, allowing the identification of the most efficient one. This policy will then be used by the model to generate storage recommendations that improve picking efficiency.

By implementing this model and following its recommendations, the warehouse organization will progressively converge toward the optimal arrangement, making it possible to address the previously identified operational inefficiencies.

3 Literature Review

To substantiate the approach followed in this dissertation, it is crucial to understand the concepts, challenges, and existing solutions in the literature on warehouse management and operations optimization.

The following review will present the existing research on relevant topics hierarchically, starting by introducing the importance and evolution of supply chain and logistics; then focusing on warehouses, their purpose and operations; and finally analyzing in detail the available research on factors influencing picking efficiency.

This theoretical base will allow establishing a solid foundation for methodology development, ensuring that best practices and the most recent research advances are followed.

3.1 Supply Chain and Logistics

The supply chain encompasses all activities from production, distribution, and delivery of goods to consumers, thus being the backbone of global commerce. Over the decades, its importance has grown, supported by digital transformation, globalization, and the evolution of consumer expectations (Lara & Wassick, 2023). This section explores the core concepts of supply chain management, the trends that have been emerging, and frames the role of warehouses in this dynamic system.

3.1.1 The Importance of Supply Chain Management

Proper supply chain management (SCM) is essential to ensure an efficient flow of goods, information, and finances throughout the entire network. This chain creates value along the path that connects suppliers, manufacturers, distributors, retailers, and customers. Competent supply chain management enhances business competitiveness, reduces costs, and allows for increased customer satisfaction (Lara & Wassick, 2023).

Recently, the vulnerability of global supply chains was exposed by events such as the COVID-19 pandemic, geopolitical tensions, and some natural disasters; this exposure caused organizations to prioritize adaptability and resilience (Boysen & de Koster, 2025).

Supply chains are also being shaped by the integration of technologies such as advanced analytics, artificial intelligence, or automation, which allow for real-time decision-making and offer predictive insights about the future (Lara & Wassick, 2023).

Modern business management has led to a paradigm shift in supply chains; traditional linear models are giving way to more complex, interconnected, digital, and flexible networks. This modification can be attributed to the growth in global trade complexity, the strong growth of e-commerce, and the development of digital technology (Boysen & de Koster, 2025).

A central point of this paradigm shift is that individual businesses no longer compete as autonomous entities but as members of supply chains. In the current scenario, there is inter-network competition, where it is not brand versus brand or store versus store, but supply chain versus supply chain. A business's success depends on its ability to establish relationships and integrate into a competent network (Lambert & Cooper, 2000).

SCM catches the synergy of intra- and inter-company integration, denoting a new way of managing both businesses and their relationships within the supply chain. Companies increasingly try to leverage the possibilities this integration offers to increase customer satisfaction by offering faster deliveries, more personalized services, and easier returns. This goal has led to the development of advanced fulfillment networks (Boysen & de Koster, 2025).

The accelerated evolution of supply chains and customer expectations has introduced opportunities but also challenges. In terms of opportunities, the strategic partnerships that companies form and the growing volume of data they have access to allow them to share resources and information, creating a win-win ecosystem; furthermore, the integration of AI-powered systems allows for more accurate demand forecasting, route optimization, and inventory management (Lara & Wassick, 2023). On the other hand, challenges arise as the exponential growth of e-commerce implies a greater demand for rapid fulfillment and last-mile delivery solutions (Boysen & de Koster, 2025); the integration across business units and the orchestration of human and machine decision-makers also becomes challenging (Lara & Wassick, 2023).

3.1.2 The Role of Warehouses in Supply Chains

Warehouses are a crucial part of supply chain operations, due to their ability to store, manage, and distribute goods centrally. They have a strategic role in ensuring supply chain efficiency,

as they allow for cost-effective inventory management while meeting customer expectations (Hackman & Bartholdi, 2019).

Inventory management plays a fundamental role in this dynamic, as some supply chain members may bear a disproportionate inventory burden. In order to reduce costs, members of the network often stock unfinished or partially finished products because their maintenance is less expensive than that of finished goods. To improve efficiency and redistribute responsibilities, a comprehensive supply chain network rationalization can have a significant influence on all members (Lambert & Cooper, 2000).

Warehouses have long been valuable assets for businesses by allowing them to accommodate fluctuations in supply and demand, and also by allowing products to be stored closer to consumers, thus enabling delivery time and increasing service level. However, the growing demand for the ability to process and deliver almost instantaneously, strongly promoted by the growth of e-commerce, forced supply chains to respond and improve; in warehouses, this step forward was ensured by technologies such as automated storage and retrieval systems (AS/RS) and autonomous mobile robots (AMRs) capable of improving efficiency, accuracy, and safety (Boysen & de Koster, 2025).

The supply chain paradigm is undergoing profound transformation, promoted by technological progress but also by consumer expectations in an increasingly global market. Warehouses, as an essential and highly strategic component, are forced to keep up with this transformation to achieve operational excellence and thus be able to respond to challenges while having the agility to take advantage of improvement opportunities when they appear. In the following sections, key aspects will be analyzed in detail, as well as the operations that occur in a warehouse.

3.2 The Warehouse

Warehouses play a pivotal role in the supply chain, serving as facilitators of the movement, storage, and management of products. The ability to store inventory and smooth the flow throughout the supply chain makes warehouses efficiency drivers, cost minimizers, and promoters of a greater balance between supply and demand. This section explores these fundamental aspects of warehouses while providing a necessary framework to understand the operations that occur within them.

3.2.1 The Purpose of Inventory and Warehouses

Inventory acts as a buffer that mitigates the uncertainty and variability that the supply chain is subject to. It enables operations to proceed continuously by accommodating fluctuations in demand, supply disruptions, and variations in lead time (Hackman & Bartholdi, 2019). Inventory has several purposes and brings added value at multiple stages of the supply chain, from raw materials to finished goods, including:

1. **Demand fulfillment**: by allowing businesses to store products closer to consumers, it is possible to reduce delivery time and thus increase service level (Gu et al., 2007).
2. **Economies of Scale**: as stock can be stored to be consumed or distributed over time, it is possible to buy in larger quantities than needed in the short term and thus acquire at lower unit prices.
3. **Hedging against uncertainty**: stored inventory protects businesses from market volatility, such as sudden demand spikes or supply-side breakdowns and delays.
4. **Seasonal stocking**: in sectors where demand is highly seasonal, inventory management allows businesses to maintain constant production over time, accumulating reserve stock to be used during peak periods.

Despite all the benefits presented, inventory is also associated with some costs such as storage costs, handling, depreciation, and obsolescence. Therefore, it is important that warehouses are managed strategically to balance the added value with the costs that maintaining inventory entails (Hackman & Bartholdi, 2019).

Although inventory offers the benefits previously discussed, it is important to note that warehouses are not just static storage infrastructures; they can also have a dynamic role in product development and contribute to adding value to it. Some examples of these activities or services are assembly, packaging, labeling, and returns processing (de Koster et al., 2007). Such activities align warehouses with broader organizational goals, such as agility, cost reduction, and customer satisfaction.

It can then be stated that warehouses can also have a prominent role in customization, facilitating last-mile alterations to meet specific consumer or legal requirements (Hackman & Bartholdi, 2019). Furthermore, their role in reverse logistics allows for efficient handling of returned products, reducing waste and adopting a sustainable approach, while providing good customer support by being able to manage returns

competently. The concept of centralizing inventory in a location that facilitates the distribution process becomes relevant again in the reverse process (Rushton et al., 2014).

3.2.2 Types of Warehouses

According to Hackman & Bartholdi (2019), there are various types of warehouses, with the main driver for classification being the customers they serve.

1. **Retail distribution center**: these centers supply retail stores, with regular and scheduled deliveries. Thus, demand is traditionally predictable, deliveries are of large volumes, and planning can be managed in advance.
2. **Service parts distribution center**: this type is challenging since the diversity of stored products tends to be high, ranging from car or airplane parts to medical equipment. The demand for these parts is difficult to predict, requiring high reserve stocks, and the existence of frequent emergency orders entails high operational costs.
3. **Catalog fulfillment or e-commerce distribution center**: in these warehouses, the focus is on individual customers who make small quantity purchases, but with high frequency and need for almost immediate service.
4. **Third-party logistics (3PL) warehouse**: storage operations can also be outsourced to this type of warehouse that typically serves multiple customers and allows for easier achievement of economies of scale. These services are also very useful for dealing with seasonality or accommodating sudden temporary peaks.
5. **Perishables warehouse**: focused on food products or those with strict restrictions, such as vaccines, these warehouses deal with items whose shelf life is very short and must meet storage and handling restrictions such as specific temperatures or prohibition of stacking that could lead to contamination.

Although there are various types of warehouses, there are systematic concepts that are transversal regardless of the industry in which each operates (Hackman & Bartholdi, 2019).

3.2.3 Challenges in Warehouse Management

As market and customer needs evolve, the logistics chain must adapt, and warehouses are also subject to these transformations. Naturally, this process can encounter some challenges, such as space limitations or labor shortage that can limit ambitious expansions; also,

technological integration can raise problems since, despite the significant benefits of advanced technologies, it is not always smooth to integrate them with legacy systems. To address these problems, there is an increasing demand for automation and robotics solutions, as well as the use of big data analytics and machine learning to generate actionable insights that allow predicting and optimizing operations (Gu et al., 2007).

We can then say that warehouses are crucial for supply chain success, and understanding their purpose and capabilities is essential to respond to emerging trends; moreover, keeping up with innovation in warehouse management is decisive for businesses to remain competitive in the future of logistics.

3.2.4 Warehouse Flow and Operations

During their journey to the final consumer, goods make a pause in the warehouse, whose duration can vary considerably, and during their passage through this element of the supply chain, they need to be handled, which consumes time and space. In the following subsections, the main warehouse operations will be described and explored: receiving, put-away, order-picking and shipping (Tompkins et al., 2010).

3.2.4.1 Receiving

Receiving corresponds to the entry of goods into the warehouse. At this moment, received products are unloaded and organized to proceed either to storage or cross-docking. This operation involves various activities, such as the assignment of docks to trucks arriving at the warehouse and the management of workforce between loading and unloading operations. Efficient management that promotes error-free operation is essential to ensure that products are processed within the expected time and directed to the right location.

There are two main ways of operating: on the one hand, there are cross-docking warehouses where received products are immediately transferred to shipping. The aim is to minimize storage time, ideally making it non-existent, with products always in movement. This approach allows for cost reduction and saves the time that storage operations would imply. On the other hand, there are traditional warehouses where goods proceed to storage after being received. In these, it may be necessary to prepare products for storage with tasks such as depalletization, application of barcode labels, or verification of quantities and condition of goods (Rushton et al., 2014).

Typically, receiving is responsible for only 10% of the operational costs of a typical warehouse (Hackman & Bartholdi, 2019).

3.2.4.2 Storage

Storage is an operation with great strategic character that involves a series of decisions that have a major impact on the efficiency of the warehouse as a whole. The activity itself consists of taking the merchandise to the location where it will remain until needed; this location can be a position on the floor, in a rack, in a refrigerated chamber, or in another support that meets the characteristics of the product in question.

The complexity of this process comes from the difficulty in knowing where to place the product since its location will be decisive for the future picking cost (Hackman & Bartholdi, 2019).

The choice of this position depends on various decisions; some warehouses opt to divide the space between picking zone, a compact zone with small quantities of each product where picking tasks can be carried out much more quickly, and reserve zone, where the rest of the stock is stored; however, there is a trade-off since the smaller the picking area, the more often internal product replenishment will be necessary, moving them from the reserve zone to the picking zone. Thus, the decision of the relative size of the two areas becomes an optimization problem where the specific context of each warehouse must be considered (van den Berg et al., 1998).

Furthermore, within each zone, it is necessary to define the storage strategy, that is, which products will be in which positions. This decision depends on the physical characteristics of the products and positions but should also aim to minimize future picking costs; to address this ambition, a series of product sorting criteria were developed (de Koster et al., 2007) which will be analyzed in detail later.

Depending on the type of warehouse and its strategic autonomy, the inventory level to maintain for each product over time can also be a decision variable (Gu et al., 2007)

When product is put away, the position where it was placed should be recorded to later be used in the construction of optimized picking routes. This storage process is typically responsible for 15% of operational expenses (Hackman & Bartholdi, 2019).

3.2.4.3 Picking

Picking corresponds to retrieving articles from their storage locations to fulfill orders that request them. This process requires a series of tasks such as pre-action, searching, travel, and the actual collection of the article. According to Chen et al., 2015, travel is particularly important since it typically accounts for at least 50% of this operation's processing time. The chosen picking policy, for example doing picking for one order at a time or for multiple orders simultaneously, has a major impact on other decisions in warehouse design and operation (Gu et al., 2007); picking policies will be explored later.

Once the products for picking are defined, picking lists are created that pickers must follow; these lists include the type of article and quantity. Furthermore, an effective product location system helps accelerate the task, since without it operators waste time searching for items that are on the list (Tompkins et al., 2010).

This operation is responsible for approximately 55% of a warehouse's operational costs. Being such an expensive and labor-intensive operation, it is a crucial activity that directly impacts customer service level and therefore requires very competent planning and management (Hackman & Bartholdi, 2019).

Chen et al. (2015) analysis attributes even greater weight to the picking operation, suggesting that its cost can represent 60 to 70% of operational costs, making it the number one priority for efficiency improvements.

Furthermore, a factor that cannot be neglected is due dates, since customer orders must be completed within certain deadlines, whose fulfillment is essential as this entire process is articulated with other stakeholders, for example distribution trucks that have scheduled departures (Gademann et al., 2001).

3.2.4.4 Shipping

Shipping corresponds to the process of products leaving the warehouse. At this point, the articles, which have been collected and consolidated in the appropriate support, are loaded onto the means of transport that will allow them to continue their journey to customers.

As mentioned previously, it is fundamental to maintain operations at the planned pace; frequently, shipping does not happen continuously but rather in specific time windows designated for this purpose. Thus, if there are delays and an order misses its scheduled

shipping time, it will have to wait for the next shipping period, which can lead to delivery dates not being met and compromising customer satisfaction (de Koster et al., 2007).

This task accounts for the remaining 20% of operating costs (Hackman & Bartholdi, 2019).

3.3 Design of Storage and Order-Picking Processes

Now that we have a better understanding of the warehouse and its processes, we will focus on the factors that have the greatest impact on picking efficiency. As previously indicated, picking and storage are highly related since the picking cost directly depends on where each product is stored. Therefore, the factors that impact both processes will be analyzed in the following subsections.

3.3.1 Layout Design

The design of a warehouse layout, also known as macro layout, is a crucial decision as it has a direct impact on picking efficiency and overall warehouse productivity. This concept refers to the physical configuration of storage facilities, including the quantity, length, and orientation of aisles, as well as the relative arrangement of different functional areas. An ideal design allows minimizing travel distances, which in turn decreases the time required to complete tasks (Caron et al., 2010).

De Koster et al. (2007) divides the layout into two subproblems. The first, which they call 'facility layout problem', consists of deciding where to place the different departments, such as receiving, storage, picking and shipping. Traditionally, the sequence of activities is considered, and departments are organized so that the flow of goods is as linear as possible. However, this problem is rarely trivial and has therefore been the subject of considerable research; in their paper, Meller & Gau (1996) comprehensively review different paths to solve it, going through variations of the branch-and-bound approach, approaches based on graph theory, and also heuristics. In addition to the location of each area, it is also necessary to define the relative size of each one; in this regard, Heragu et al. (2005) proposed a solution that uses a mathematical model together with a heuristic algorithm to determine the size of each area while simultaneously considering the flow of products within the warehouse.

The second subproblem of de Koster et al. (2007) is the 'internal layout design'; this refers to determining the quantity and dimensions of aisles and blocks within the storage area, where the objective function to be minimized is the travel distance.

Roodbergen (2001) suggested a nonlinear objective function that considered both the average number of picks and the average number of aisles that needed to be visited when preparing an order, to determine the aisle configuration that minimizes the average distance of a picking route.

Also relevant are the conclusions obtained by Caron et al. (2010), who estimate that the picking area layout can impact the expected picking travel distance by more than 60%. It was also concluded that the optimal layout is interdependent with other strategic decisions, notably the picking policy, since this directly influences the average number of picks per tour. A general rule proposed is that a layout with a minimum of aisles is preferable if the average distance between adjacent pick locations (defined as the ratio between the overall length of picking aisles and the number of picks per tour) is short, but for higher average distances, it is worth increasing the number of aisles, with the ideal number depending on the total length of the picking area.

The impact of aisle characteristics was also studied by Petersen (2002), who used simulation to study the effect of aisle length and number of aisles on total travel time. However, beyond the impact of average order size, Petersen demonstrated that a volume-based storage policy (storage policies will be analyzed in detail later) leads to shorter distances than a random policy, regardless of the picking zone configuration, once again underlining the complex interaction between different strategic decisions.

Given the need to consider so many distinct decisions and the absence of a comprehensive methodology that would allow making them in an integrated way, the work of Sprock et al. (2017) emerges, which instead of seeking to optimize individual aspects, seeks to optimize the solution as a whole, through a hierarchical approach, dividing the design into a set of subproblems. Then, an intensive search of the solution space is performed, the best candidates for consideration are identified, and they are presented to a decision maker who can evaluate the trade-off of each option.

The articles presented so far focus essentially on low-level systems, that is, systems where products can comfortably be reached by pickers when they are at ground level. However, three-dimensional warehouses present particularities that distinguish them, and the gaps in the available literature on this topic become evident.

An exception is the work of Park & Webster (1989), which focuses on finding the ideal location for products in rack positions according to their rotation, where in this scenario the aim is to minimize travel time and not necessarily travel distance, since the speed of ascending and descending to rack positions is different from the speed of movement on the ground.

The review of these articles shows that the layout design decision relates to other aspects of warehouse management, and it is of utmost importance to consider all factors, instead of trying to optimize each problem individually, to arrive at a robust solution.

3.3.2 Storage Assignment Policies

After merchandise is received in the warehouse and until it is picked to fulfill customer orders, it needs to be stored in some position. The storage assignment policy is a set of rules that allows assigning storage locations to each product (de Koster et al., 2007)

A crucial decision that must be made initially is whether or not to adopt a forward-reserve system. In this system, the warehouse is divided into 2 areas: in the forward area (sometimes called picking area), reduced quantities of each product are stored and this is where picking will be done; while in the reserve area, the remaining stock of each product is stored, which will serve to replenish the forward area according to needs. The major advantage of this approach is that by using a smaller area, picking becomes more efficient due to the shorter distances that need to be traveled. However, there is a relevant trade-off: the smaller the forward area, the less quantity of each product it can store and thus will require more frequent replenishments. Naturally, this internal replenishment operation also requires labor, and the more intense it is, the more resources it will need – resources that are no longer available for other operations (van den Berg et al., 1998).

Hackman & Bartholdi (2019) mathematically demonstrate that not all products should be placed in the forward area. Even if a product is very popular, what should be considered is the time savings between picking in the reserve or forward area and comparing that time with the cost of doing internal replenishment. A simple illustrative example is a product whose picking in forward saves 2 minutes per pallet compared to reserve, but whose replenishment cost is 3 minutes. In this case, putting the product in forward results in a net loss of 1 minute per pallet.

After the forward-reserve decision, rules must be defined that will allow allocating products to specific locations in the warehouse. In the literature, several options are presented, each with advantages and disadvantages, which will be explored next.

In a random storage policy, any received product is placed randomly in one of the free positions, all having the same probability of receiving the product. Choe & Sharp (1991) concluded that in general, this type of policy allows for greater space utilization but is not efficient in travel distance, compared with other strategies.

The attempt to implement a random policy in practice, can actually lead to another policy, which is the closest open location store. In this case, the product is placed in the first empty position that the picker finds when storing it, so the probability that a position will receive a product is not the same for all positions. This approach leads to locations closest to the entrance being almost always occupied, but as one moves away from the entrance, positions become gradually emptier (de Koster et al., 2007). Although this policy tends to store products in the most prime positions of the warehouse, Hausman et al. (1976) suggest that performance is not better than in random storage if movements are only of complete pallets.

Another suggested policy is dedicated storage. In this strategy, each product is always stored in the same position, which remains reserved for it. This approach allows increasing pickers' familiarity with product locations, which can bring efficiency gains; however, it requires positions to be unavailable to receive other products even when those assigned to them are out of stock. This makes this policy inefficient in terms of space utilization. A factor to consider is that this policy can be used to leverage picking efficiency when product weight is relevant for routes. In cases where the route must start with heavier products, so they end up at the bottom of the pallet, these products' positions can be strategically assigned so they can be visited efficiently at the beginning of the route (de Koster et al., 2007).

A fourth approach that can be used is to allocate based on product characteristics. In this type of policy, different criteria can be used, but the three most frequent are popularity, maximum inventory, and Cube-Per-Order Index (COI) (Gu et al., 2007).

The popularity criterion consists of placing the most requested products in the best positions, that is, positions closest to the shipping area. The rationale is that if the picker has to go to this location many times, we want the trip to be as short as possible (Gu et al., 2007). A limitation of this criterion is that it doesn't consider product size; this can lead to large-volume products being placed in the best positions, quickly occupying them with a small quantity of items (Kallina & Lynn, 1976).

Another criterion is maximum inventory. In this, the space needed in the warehouse to store all inventory of a product is considered, and products are then ordered by placing those

that require less space in the best positions (Gu et al., 2007). In this case, a relevant limitation is the fact that it completely disregards product popularity.

In an attempt to leverage the advantages of the previous two criteria while mitigating their limitations, Heskett (1963) suggests a new criterion, the Cube-Per-Order Index (COI or COP), which is defined as the ratio between the maximum allocated storage space and the number of storage/retrieval operations of a product per unit of time. This criterion allows balancing the trade-off between popularity and volume, favoring products with higher turnover but penalizing them if they are very voluminous and occupy more warehouse positions. In practice, the COI of each product is calculated and then they are assigned to positions in ascending order with products of lower COI getting better positions.

Initially, COI was presented as a heuristic method without claims of optimality; however, over time, some studies proved that under certain assumptions, this criterion effectively guarantees optimality when the objective is to minimize the long-term average order picking cost (Gu et al., 2007). One of the main studies in this matter was done by Kallina & Lynn (1976), who pointed out important assumptions such as compatibility and complementarity, which are, respectively, the possibility of all products being stored next to each other (without contamination problems, for example) and the treatment of products that are frequently sought together as one. In the work of Malmborg & Bhaskaran (1990), COI optimality was evidenced for dual-address models, that is, in each picking cycle, a storage operation and a retrieval operation are paired.

The use of policies based on product characteristics, despite proven theoretical efficiency gains, can be quite difficult to manage in practice. Not only because they require greater effort to administer, as exhaustive information about products is needed, whether demand, size, or both, but also because factors such as demand can have great volatility, which would imply constant rearrangements of the warehouse to keep it according to the adopted criterion (Petersen et al., 2004).

To capitalize on the gains of these policies without making the workload overwhelming, the concept of class-based storage emerged. This method uses one of the measures presented previously to classify products into a class. For example, when turnover approximately follows a Pareto distribution, a division into 3 classes can be used: A, B, and C; where the 15% of products with highest demand generate around 85% of turnover and are classified as A; the next 35% are classified as B, and the remaining 50% will be C. In this strategy, each

class is then assigned to an area of the warehouse, with class A getting the best positions, and the respective products are stored in that area (de Koster et al., 2007).

Petersen et al. (2004) studied the efficiency gap obtained by using class-based storage, and his results indicate that a system with only 2 classes already allows obtaining 80% of the benefits of a popularity-based policy, while requiring much less effort and time to manage. As the number of classes increases, the efficiency gap decreases, and this policy increasingly approaches a pure popularity or COI-based policy.

3.3.3 Routing methods

Travel time is a necessary expense of warehouse operations; however, it is a task that adds no value. For this reason, it is an activity that wastes labor hours and should be optimized to increase the level of customer service (Hackman & Bartholdi, 2019). The objective of routing is to find the shortest path that allows visiting a list of positions – the positions where the items of a given order are located. This problem can be classified as a Steiner Travelling Salesman Problem (STSP), a special case of the traditional Travelling Salesman Problem, since, on one hand, only some warehouse positions must be necessarily visited, and on the other hand, nothing prevents a position/aisle from being visited more than once (de Koster et al., 2007). Since this is an NP-hard problem, generally not solvable in polynomial time, several authors dedicated themselves to finding heuristics that would allow finding acceptable solutions in reasonable time.

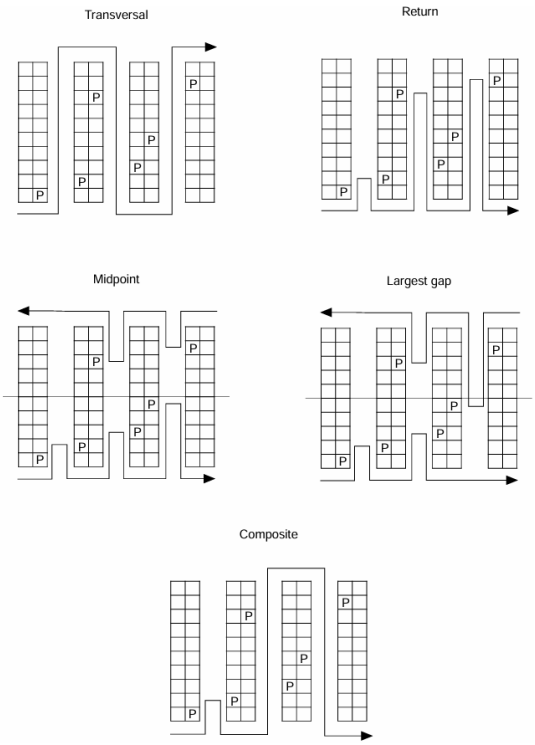


Figure 2 – Representation of routeing strategies. Reprinted from "An evaluation of order picking routeing policies", by C. G. Petersen, 1997, International Journal of Operations & Production Management, 17(11), pp. 1098–1111

In Petersen (1997), several routing policies are presented. These routing policies are illustrated in Figure *2* and will be described next. In the transversal strategy, the picker enters the aisle at one end and exits at the other, repeating this movement in all aisles that need to be visited, and finally heads to the shipping area. In the return strategy, the picker enters the aisles where picks must be made, but after visiting all mandatory positions in that aisle, turns back and exits through the same end where he entered. There is also the midpoint strategy, where each aisle is divided into two parts of equal length and the picker visits only the half on the side of the end where he entered, with the other half potentially being visited later, if necessary, when he is repeating the process for the other aisle ends. There is also the largest gap strategy, which is an extension of the midpoint strategy; in this one, the distance between the start of the aisle and the first pick, the distance between adjacent picks, and the distance between the last pick and the end of the aisle are compared – the picker enters an aisle as far as the largest gap within it, this one not included. For example, if the largest gap is between the beginning of the aisle and the first pick, the picker does not enter the aisle at that end, carrying out the respective picking when he is going through the other ends of the aisles. This approach never performs worse than the midpoint method (Hall, 1993). It is also possible to combine features of the return and transversal strategies to obtain a composite

strategy. This approach allows gaining efficiency by minimizing the distance between the farthest picks in two adjacent aisles, since between consecutive aisles it uses the transversal strategy only if it allows to travel a shorter distance.

Despite the difficulty of this problem, Ratliff & Rosenthal (1983) managed to develop a procedure that guarantees the optimal solution, in warehouses with certain characteristics, quickly and with reasonable computational power. However, even in warehouses that meet these characteristics and where the optimal solution can be obtained quickly, it may be beneficial to use heuristics. This argument is justified by the simplicity of routes generated by heuristics, in contrast to often complex and intricate routes that the optimal solution implies; thus, in practice, optimal routes may seem illogical leading to lower picker adherence; while heuristic routes are easy to understand and end up having more adherence (Gademann & Velde, 2005). In the study conducted by Petersen (1997), the results indicate that the gap between the optimal solution and the best heuristic solutions is only 5%, thus reinforcing the usefulness of using heuristic methods that generate near-optimal routes, are more consistent and easy to understand, and avoid the complexity that optimal solutions often imply.

3.3.4 Order-Picking Policies

The choice of picking policy is another fundamental strategic decision, since it can enhance or limit the efficiency of the picking process. The choice made will have consequences in other aspects, such as routing, and is, itself, affected by decisions like the storage assignment policy, therefore, it is important to consider the interdependence of these problems instead of trying to optimize them individually (van Gils et al., 2018).

The two main approaches presented by de Koster et al. (2007) are pick by article (batch picking) and pick by order. In pick by order, each picking route only collects items for the same order. This method allows the picker to be dedicated to only one order, reducing errors in collecting items for it; however, it leads to inefficient routes in the long run, since it doesn't capture the synergies that could exist if, when the picker goes to an aisle, they immediately recognize the products from that aisle/zone that will be needed in subsequent orders. This picking policy is more suitable when orders are very large, making it worthwhile to have one or more pickers exclusively dedicated to it. On the other hand, pick by article, more appropriate for smaller orders, explores the advantage of pickers dedicating themselves to more than one order in the same route and thus reducing total travel time. In this strategy, several orders are prepared simultaneously by the operator, for example, if there are several

orders that require the same product, the operator can fetch the entire quantity needed at once and distribute it among the orders more efficiently than if he did it one by one. The grouping of orders can be done through criteria such as the proximity of required product positions or the time window in which orders were received or need to be shipped.

The paper from Petersen & Aase (2004) evaluates the impact of picking, routing, and storage policies on the efficiency of the picking process. The results indicate that using batch picking as a picking policy is the factor that, by itself, has the greatest impact on operational efficiency, being particularly evident when order size is small.

3.4 Final Considerations on the Literature Review

The literature review proves that warehouse operations optimization is a widely studied topic, thus demonstrating the practical importance of this problem. Moreover, it is possible to understand that it is a complex issue in which several interdependent factors need to be optimized holistically to obtain robust solutions.

Various proposed approaches were presented in different strategic vertices, such as layout, storage policy, picking policy, or routing methods. Based on these contributions, a model adapted to the specific warehouse context will be developed, grounded in best practices and principles to ensure that the obtained solution is appropriate.

4 Methodology

The methodology adopted is based on two initial pillars, which together support what will be referred to as the macro plan – essentially, the assignment of products to storage positions that minimizes the long-term average picking time. We will begin with the positions pillar, where we will detail how the positions were mapped and the criterion used to identify the best ones – with best understood as those that are fastest to access. We will then move to the products pillar, in which the step-by-step process for calculating the COI of each product will be explained. Finally, these two pillars will underpin the generation of the storage plans for different scenarios, which will be evaluated through simulation in order to identify the one that minimizes picking cost, and which will be adopted as the macro plan. The macro plan will then be the guide for the two use cases we will also explore.

4.1 Layout Mapping

In order to have a faithful representation of the warehouse that could be used not only for simulations but also for the generation of the macro plan, the approach used was to map the warehouse layout in an Excel file. Each Excel cell was defined to correspond to a 1m × 1m square in reality. This scale proved useful because it approximates the dimensions of a storage position, which corresponds to the size of a European pallet (120 cm × 80 cm), and also allows mapping the aisles and remaining operational area with adequate precision for the warehouse dimensions. In the Excel file, all cells corresponding to impassable locations, such as walls, pillars, the racks themselves, or other obstacles were filled with -1, and passable cells with "WASD". This acronym was used because these keys are traditionally used on keyboards to make forward, left, backward, and right movements, respectively. The intent was for this method to handle scenarios where some zones/aisles were unidirectional – in those cases, cells would be filled only with letters representing the directions in which movement could exist. However, in the warehouse in question, this restriction was not observed, and all passable cells could be traversed in any direction. Additionally, all ground-level storage positions were mapped; in this warehouse, storage is done in 5-tier racks, so above each ground position, there are four more positions in height. Each position is identified in the WMS by a code, and these codes were used to identify the ground positions

in the corresponding Excel cells. The shipping area was also identified, placing the code "EXP" in the central cell of this area.

The layout file, originally stored in Excel, was imported and mapped onto a grid structure using Python, enabling efficient handling. The objective of this Python script was to obtain the traveling distance between all pairs of ground positions in the warehouse, taking into account layout constraints and the physical obstacles mentioned before, identified with -1. To obtain these distances, the A* algorithm was used, a pathfinding method that finds the shortest path between a starting point and an arrival point (Foead et al., 2021). To use this algorithm, it was first necessary to create a function that identifies for each grid cell the neighbors that can be visited, i.e., the passable cells, allowing both orthogonal and diagonal movements to approximate the obtained routes as closely as possible to the actual possible routes in the warehouse. After defining which movements could be made from each cell, considering the warehouse's physical restrictions, the A* algorithm was iteratively applied to all pairs of positions.

In this specific case, we were only interested in obtaining the distances and not the path itself, so a modification was implemented for the algorithm to return only the distance and not the route. Recalling that each cell represents 1 m² and that orthogonal and diagonal movements are allowed, the distance required by each of these movements is 1 m and $\sqrt{2}$ m, respectively. Furthermore, knowing that in this warehouse there are no unidirectional movement restrictions, it was possible to increase the algorithm's efficiency by calculating only the distance between each pair of positions in one direction, since the shortest path in the opposite direction would be the same and consequently the distance as well. After running the algorithm, we obtained a table with columns for initial position, final position, and distance, filled with all combinations of ground positions and their respective distance in meters, to which we also added the distance between positions in the reverse direction. This table was saved in the database.

As mentioned earlier, the warehouse has positions at five height levels, so using distance to compare positions and determine which are most advantageous for picking is somewhat limited. On one hand, if we consider only the distance traveled on the ground, we are neglecting all the effort to go up and down to pick from higher positions; on the other hand, it would be too crude an approximation to consider a position's height in meters and add it to the ground distance, since the speed of movement on the ground is different from the speed in height. For these reasons, it was decided to use the time between positions

instead of distance. This decision implies that we should calculate the time needed to go from one position to another, knowing the height of both positions and the ground distance needed to travel between them.

To address this challenge and accurately model the time required to move between positions, empirical time measurements were carried out during regular warehouse operations. While pickers worked normally, the time they took to move between pairs of positions was measured, as well as the time they needed to retrieve a pallet from a given level and put it back before being ready to proceed to the next position.

The times were then used to adjust linear regressions and obtain coefficients that allow calculating the time between positions. At this point, the focus is on calculating the time between the shipping area and each storage position, since this allows us to order all positions in increasing time to expedition and thus identify the most valuable positions, those that are quickest to access, and the least valuable positions in the warehouse, those that take longer to reach.

It is important to note that these time functions will be reused at a later stage, when simulating actual picking routes. In that context, the objective will be to calculate the time between consecutive positions in a route, based on the products requested in each order and the exact location of the units to be picked, which depends on expiration date and stock distribution at that moment. However, for now, the analysis remains agnostic to those route-specific details and focuses solely on time-to-expedition as a proxy for position value.

In the case of the ground movement component, the measurements confirmed that travel time is linearly dependent on the distance to be covered. On average, operators take 1.565 seconds to travel each meter, therefore, time can be obtained as: $travel\ time = 1.565 \times distance$. Note that we used the measured times and the distance between the respective positions that had already been calculated through the A* algorithm applied to the Excel mapping.

In the case of the height movement component, 2 subcomponents were considered: retrieval time and storage time. The reason for this is that these 2 operations have some operational differences: while at the moment of retrieving the pallet to pick a product, the pallet is removed from the rack and placed on the ground, a spacious area where the pallet can be placed without requiring precise alignment; at the moment of storage, the pallet must be placed back in its rack position. In this process, precision is much more important to ensure that the pallet is not only fitted into its designated space but also placed in safe

conditions and remains stable. Thus, the times required to retrieve the pallet from a certain level and the time required to put it back were measured separately. Using the measured times, the level associated with each measurement and knowing that each level has a height of 2 meters, it was possible to fit two linear regressions and obtain functions that indicate the time required for each operation based on the height in meters of a position. For retrieval time, the function obtained was $retrieval\ time = 6.8 + 7.75 \times position\ height$, while for storage time, it was $storage\ time = 15.8025 + 5.5864 \times position\ height$.

After obtaining these functions, it was possible to calculate the time from the shipment area to each position. For this, a table was first created with columns for initial position and final position. In the initial position, all rows were filled with "EXP" and in the final position, they were filled with all distinct values of the position codes on the ground and in height. Next, a column was added with the ground position of the final position, i.e., in the case of positions that are already on the ground, this position is itself; in the case of positions in height, this corresponds to the code of the position that is in the same column but at ground level. Identifying this position is important because it is the one that will be used to evaluate the distance and consequently the time to travel on the ground from the shipping area to any position. In addition, a column with the height of the final position was added; there was no need to add the height of the initial position since shipping is always at ground level. At this point, the time between shipping and all positions in the warehouse was then calculated, using the time functions presented previously to calculate: the time needed to travel on the ground from shipping to the site of each position, to which was added the time needed to retrieve and store a pallet at the height of that position. The final result was a table with columns for initial position "EXP", final position, and time from expedition, which was saved in the database.

4.2 COI Calculation

As presented in the literature review, COI is a product allocation criterion in warehouses that seeks to minimize the labor costs required for picking orders. It attempts to balance the trade-off between the space needed for storage and the popularity of each product.

$$COI_j = \frac{picking\ volume_j}{popularity_j} \quad (1)$$

As shown in equation 1, the COI of a product j can be calculated by dividing the volume of that product to be stored by its popularity, measured as the expected number of orders during a defined time period.

At this stage, it is important to recall the concept of dividing the warehouse into picking and reserve. In this system, the warehouse is divided into 2 zones: in the picking zone, reduced quantities of each product are stored and this is where picking will be done; while in the reserve zone, the remaining stock of each product is stored, which will serve to replenish the picking zone according to needs. Typically, the picking zone gets the best positions, that is, those closest to the shipping dock, and the reserve zone is placed in less accessible areas of the warehouse. The major advantage of this approach is that by using a smaller zone, picking becomes more efficient due to the shorter distances that need to be traveled. However, there is a relevant trade-off: the smaller the picking zone, the less quantity of each product it can store and thus more frequent replenishments will be required. Naturally, this internal replenishment operation also requires labor, and the more intense it is, the more resources it will need (van den Berg et al., 1998).

This distinction is important because, in this dissertation, COI will be leveraged in different scenario configurations, and while some will include a division of picking and reserve zones, others will not have this division. In the scenarios without such a division, the entire stock volume of each product will be considered. In contrast, in the scenarios with a division between picking and reserve zones, the volume used in the COI calculation refers only to the quantity allocated to the picking zone.

Having presented the COI formula and explained the differences to be considered in systems with and without reserve zones, we now turn to the explanation of how each of the required components was obtained, namely, the numerator and the denominator of the COI.

Beginning with the numerator – the volume – a major challenge immediately arose in the warehouse under study: there was no information available about the volume of the products to be stored, and measuring the volume of each individual product was not a feasible option due to the large number of distinct items handled by the warehouse. However, historical daily stock data was available, which made it possible to determine, for each day, which products and in what quantities were present in each storage position. Using this data, it was possible to infer the quantity of each product that can be placed in a single storage position. The available historical data, covering approximately two years, was

analyzed to identify the maximum quantity of each product that had ever been present in a single position, and this value was used as the maximum quantity of that product that can be stored in a single location. It should be noted that all warehouse positions have identical dimensions.

In a warehouse with multi-product positions, where several items can coexist in the same storage location, the number of potential combinations of products and quantities is extremely large. To overcome this challenge, the presented single-product logic was crucial: historical stock data was used to estimate the maximum quantity of each product that can be stored in a single location, assuming it was stored alone. This estimation makes it possible to then calculate the number of positions required for storing any set of products and quantities by summing up the required positions for each product. In practice, storage often follows a multi-product logic, but the total number of positions needed remains consistent across both perspectives: single-product (theoretical calculation) and multi-product (operational reality).

This methodological solution – the development and application of the single-product logic to a multi-product warehouse – represents one of the key contributions of this project. It provided a robust and practical way to address the complexity of multi-product storage in the absence of volume data, enabling the design of realistic and feasible storage plans.

Given this information, the effective volume in the original COI calculation was replaced by the number of positions required to store the relevant quantity of each product. With the new COI calculation being as shown in equation 2:

$$COI_j = \frac{picking\ positions_j}{popularity_j} \quad (2)$$

This approximation presents two limitations. On the one hand, for some products, on the day and in the position where the maximum was recorded, other products were also present in the same position. In these cases, if those other products had not been there, it might have been possible to store an even greater quantity of the product being analyzed, depending also on the packaging format and stacking arrangement. On the other hand, even in cases where the maximum was achieved with the product stored alone, there is no guarantee that the full storage capacity of the position was actually utilized, since the true maximum for that specific product might never have been reached. Despite these two limitations suggesting that the maximum quantity per product per position might be

underestimated, the availability of two years of stock history in a high-rotation warehouse provided a sufficiently robust volume of data points to consider the approximation reliable.

Moreover, using the number of storage positions instead of the product volume is even more appropriate for the specific context of this warehouse. It should be noted that the importance of volume in traditional COI calculations lies in the fact that placing smaller volumes closer to the shipping area reduces travel distance. However, since storage is constrained to rack positions with fixed dimensions, if the products within a position do not occupy the entire available space, the picker still has to traverse the full length or height of that rack position to reach positions after it. Thus, the number of positions a product occupies has a greater impact on the distance traveled by pickers than its volume.

After calculating the quantity of each product that could be stored in a position, the next step was to calculate the average stock on hand (SOH), meaning the average quantity of units of each product stored in the warehouse, again using the historical stock data.

Subsequently, the average SOH for each product was divided by the maximum quantity that can be stored in a single position. The result was rounded up, since, under the single-product macro plan logic, a partially filled position is still dedicated entirely to that product. This number corresponds to the average number of positions needed to store all the stock of each product.

To illustrate this calculation, consider an example involving a product j, for which the maximum quantity ever stored in a single position was 50 units. This observation proves that at least 50 units of this product can be stored in one position. Given that the average SOH for this product is 130 units, the number of positions required to store this average quantity can be estimated by dividing 130 by 50, resulting in 2,6 positions. However, as previously explained, this result must be rounded up, therefore, product j is considered to require 3 storage positions. This example is summarized in *Table 1*.

Table 1 - Numeric example of positions needed for product j

Product	**Maximum Quantity**	**Average SOH**	**Positions Needed**
j	50	130	3

In the case of scenarios without a reserve zone, the number of positions required for each product to be used in the COI calculation has been found, since in these scenarios the entire stock of each product is considered.

On the other hand, in scenarios that include a separation between picking and reserve zones, it is still necessary to determine, among the total number of required positions for each product, how many should be assigned to the picking zone – picking positions – and how many to the reserve zone – reserve positions – given that only the picking positions are considered in the COI calculation. This division depends on the picking protection period, defined as the time interval of orders that the stock held in the picking zone should be able to fulfill.

To define the protection period, several factors were considered, including the operational cycle, the type of warehouse, and its position within the supply chain. The analysis of reception and shipment patterns revealed the existence of a weekly logistical cycle: the containers in which products arrive at the warehouse are transported by two ships per week (although the exact arrival days may vary) and are then transported by truck from the port to the warehouse; the shipment of orders, while subject to some volatility, tend to follow relatively stable weekly patterns, with regular orders from the main clients. This weekly cycle affects not only inbound and outbound flows but also the planning and allocation of resources by warehouse management. A protection period that does not align with this cycle could lead to decisions that are out of sync with operational reality, compromising planning efficiency.

Additionally, another relevant factor is the warehouse's position within the supply chain. As a facility located in the final stages of the supply chain – delivering directly to end destinations such as retail stores, supermarkets, and restaurants – its role is not to accumulate large volumes of inventory for extended periods, but rather to ensure fast and frequent deliveries of high-rotation products. As a result, stock coverage is relatively low and inventory turnover is high. In this context, adopting a protection period longer than one week would result in a significant portion of the inventory being stored in the picking zone, undermining the strategic purpose of maintaining a functional separation between picking and reserve areas. Therefore, a one-week protection period proves to be the most appropriate, as it reflects both the natural operational cycle and the specific logistical role performed by this warehouse.

Having defined the protection period, the historical order data was used to calculate the average quantity of each product needed to fulfill orders over one week. This is the quantity that must be available in the picking area. Thus, the number of picking positions

required to accommodate this quantity was calculated by dividing the product picking quantity by the maximum quantity that can be stored in one position.

It was also necessary to determine how many reserve positions should be assigned to each product. Since the total number of positions required and the number of picking positions had already been calculated, the number of reserve positions could be easily obtained by subtracting the picking positions from the total number.

The number of picking positions is what will be used for the COI calculation in scenarios with reserve zone. However, before detailing this calculation, it is important to highlight an adjustment made to ensure that valuable picking positions are used as efficiently as possible. To do this, the "real picking quantity" was calculated. The idea behind this adjustment is to make full use of picking positions, even if the originally required quantity would not fill them completely. This concept can be illustrated with the numerical example used before: imagine a product for which a position can store 50 units, but the average weekly demand is only 40 units. Initially, only the 40 units would need to be stored in the picking zone, but it would not make sense to allocate a valuable position in the warehouse with only 80% of its capacity filled. Therefore, the remaining capacity is used to store more units of the same product. Thus, the "real picking quantity" is obtained by multiplying the number of picking positions by the maximum quantity that can be stored in one position. This way, the protection period for the product is effectively increased, as the picking quantity now covers more than one week of orders. The "real protection period", in weeks, can be calculated by dividing the real picking quantity by the weekly demand for that product. *Table 2* illustrates the numerical example discussed above.

Table 2 - Numerical example of real protection period for product j

Product	**Maximum Quantity**	**Average SOH**	**Positions Needed**	**Picking Quantity**	**Picking Positions**	**Real Picking Quantity**	**Real Protection Period**
j	50	130	3	40	1	50	1,25

The concept of "real protection period" is important because it impacts the frequency of replenishments needed between the reserve and picking areas: the smaller the picking quantity, the more frequent the replenishments; conversely, increasing the picking quantity, as done here, reduces the need for replenishments over the same period. The

methodology used to estimate replenishment effort across different scenarios will be discussed in more detail in the following sections, when the simulation model is presented.

The explained calculations of total positions and picking positions allow to obtain the COI numerator of each product in the different scenarios. The denominator is the product's popularity, which was calculated as the number of weekly orders for each product. Since the same time period is applied uniformly across all products, its specific duration becomes irrelevant: increasing or decreasing the time window would proportionally affect all products and therefore would not impact their relative COI values. It is important to note that the number of weekly orders is different from the weekly quantity ordered: when analyzing the number of positions needed, the interest was in the total quantity, but for the denominator of COI, what matters is how many orders request the product each week, since this determines how many times the picker will need to go to that position. It was assumed that when a product is ordered, the picker can retrieve the total requested quantity at once – an assumption that is appropriate for the type of operation and orders handled by the warehouse.

Finally, the COI for each product was calculated by dividing the corresponding number of positions by the number of weekly orders for that product. Returning to the example of product j, and using an illustrative value of 5 orders per week, the COI can be calculated as follows. In scenarios without reserve, where all stock is considered, the COI is obtained by dividing the total number of positions by the product's popularity: $COI = 3 \div 5 = 0{,}6$. In scenarios with a division between picking and reserve zones, only the number of picking positions is considered in the numerator. In this case, product j requires only 1 picking position, resulting in: $COI = 1 \div 5 = 0{,}2$.

4.3 Scenario Generation

After classifying storage positions based on their accessibility and products according to their COI, we are now prepared to generate storage plans. The idea is to create plans that reflect different scenarios and later evaluate each of them by simulating historical orders, with the goal of identifying the configuration that minimizes the average order picking time. In this sense, the storage plans will serve as inputs for the simulation.

During the COI calculation, we had already determined the number of positions required for each product based on its average stock on hand (SOH). Additionally, we

identified how many of those positions should be allocated to picking and how many to reserve.

However, an adjustment was necessary to account for products that are only stocked on a seasonal or temporary basis. If the allocation was based solely on the average SOH of these products, they would be considered as requiring positions throughout the year, even though in practice they are only present for a few weeks, being replaced by others over time. This would artificially inflate the number of positions required. To mitigate this issue, we used the average number of positions actually occupied by each client during the last month as the reference for determining the required positions. This ensures that the dimensioning is grounded in historical reality. In each scenario, the number of positions by class and function (picking or reserve) was then adjusted proportionally, while ensuring that the overall number of positions per client remained consistent with historical values. It is important that the reference period is not too long, as a larger window would be less sensitive to recent changes in client storage needs, for instance, an increase in their stock levels. For this reason, a one-month period was adopted, ensuring both accuracy and responsiveness to recent trends.

As mentioned earlier, one distinctive feature of this warehouse is its multi-product storage policy, where multiple products can be stored within the same position. The rationale behind this policy is to promote more efficient use of space. However, despite this flexibility, there are operational constraints that must be considered – most notably, the prohibition against placing units of the same product with different expiration dates in the same storage position. This restriction exists to prevent picking errors. Recall that the WMS guides operators to the positions where products should be picked, specifically directing them to the locations that contain units with the earliest expiration date for each item. If other expiration batches of the same product were stored in those positions, there would be a risk of picking the wrong units or, at the very least, the picker would spend more time selecting the correct ones.

To accommodate this restriction while still capturing the benefits of a multi-product policy, a class-based storage strategy was adopted. In this approach, instead of assigning fixed positions to individual products, each product is assigned to a class, and each class is in turn assigned to a group of positions in the warehouse.

The classes were defined based on COI values using an ABC classification scheme. Class A includes the 20% of products with the lowest COI; these are considered the most

strategic from a storage efficiency perspective and are therefore assigned to the most accessible positions in the warehouse. The next 30% of products were allocated to Class B and the remaining 50% to Class C.

This classification was supported by an analysis of historical picking data, as shown in *Figure 3*. The graph displays the cumulative distribution of pickings over the past year, plotted against the cumulative percentage of products, sorted in descending order of picking frequency.

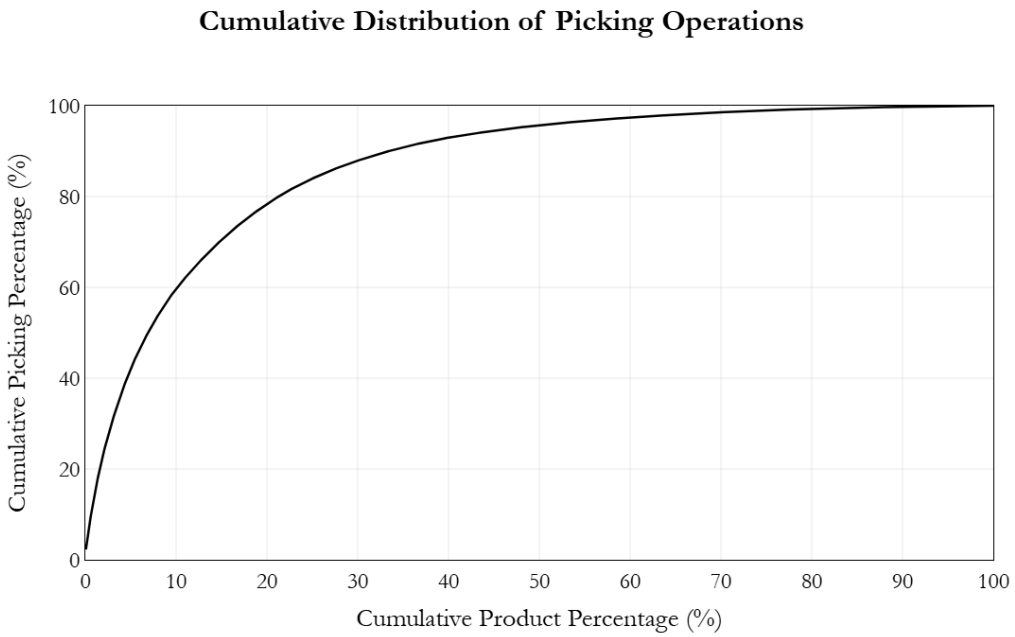


Figure 3 - Cumulative distribution of pickings per cumulative percentage of products (ranked by frequency)

This distribution analysis reveals a highly concentrated pattern of operational effort. Specifically, the top 20% of products, those with the highest picking frequency, are responsible for almost 80% of all pickings recorded during the past year. In other words, four-fifths of all picking operations are concentrated in just one-fifth of the product catalog. The following 30% of products account for an additional 15% of the total pickings, while the remaining 50% of products, representing half of the assortment, contribute to only 5% of the picking activity.

These proportions provide empirical evidence of a highly skewed distribution of operational workload, reinforcing the logic of the ABC categorization. The 20-30-50 breakdown aligns naturally with this observed pattern, allowing the storage system to

prioritize high-frequency items and optimize accessibility for the products that most impact daily picking effort.

It is important to clarify why the thresholds for the ABC classification were determined based solely on product popularity, rather than on the COI values themselves. Although COI is the main criterion used to drive product placement decisions in the proposed storage plans, it is a composite indicator, a ratio that reflects the trade-off between space requirements and popularity. As such, the COI does not possess a cumulative physical meaning. In other words, while one can mathematically compute the cumulative COI for a set of products, this cumulative value lacks direct operational interpretation. By contrast, the number of pickings is a concrete, observable measure of workload in the warehouse. Stating that the top 20% of products account for 75% of all pickings is a meaningful insight that reflects real operational effort; saying that 20% of products account for a certain percentage of cumulative COI would not convey any actionable or interpretable message. Therefore, for the specific purpose of defining ABC classification thresholds, popularity (as measured by historical pickings) provides a more intuitive and robust basis for decision-making.

To illustrate the practical advantages of this flexible, class-based, and multi-product strategy, consider the example of two products for which the desired picking quantity exactly matches the capacity of one storage position each. However, to reach that quantity, units from two different expiration batches are required for each product. According to the warehouse policy, units with different expiration dates for the same product cannot be stored together. If a fixed-position policy were used, this would require four positions (two per product – one for each expiration date).

However, with a class-based, multi-product strategy, one expiration group of the first product can be placed in one position and one group of the second product in another. The remaining space in each of those positions can then be used to store the missing expiration group of the other product.

This approach respects the operational constraint while using only two positions to store the required quantities, maximizing space efficiency and minimizing the potential for error. This example highlights how a multi-product storage strategy is particularly well-suited to the constraints of this warehouse.

The class-based, multi-product logic was consistently applied both in the generation of the storage plans and in the simulation model. However, in the simulation, products were

assigned to positions in the required quantities, but it was not considered whether those quantities included units with different expiration dates, and consequent incompatibilities.

This simplification was necessary because the combinations of expiration dates and quantities vary significantly over time and depend on the actual stock available at each moment. Incorporating this level of granularity would make it difficult to obtain a stable and repeatable solution.

Despite this simplification, it is important to emphasize that it is not expected to affect the accuracy of the simulation results. The simulation focuses on the picking routes, which depend only on which positions must be visited, not on the full content of each position. In other words, it is irrelevant whether a visited position contains one or multiple expirations batches of the product, what matters is that it contains the product that must be picked for the order in question.

After clarifying the key concepts previously discussed, we now present the four developed scenarios: *Baseline*, *Owner w/ Reserve*, *Owner w/o Reserve*, and *No Owner*. These scenarios represent varying degrees of disruption compared to the warehouse's current setup, and the objective is to determine which scenario leads to the most efficient performance. It is important to point out that, in this context, the terms owner and client are used interchangeably. Since the warehouse operates as a third-party logistics provider (3PL), it does not hold ownership of the products it stores, the actual owners of the inventory are the clients who contract the warehouse's services.

Starting with the *Baseline* scenario, it is designed to replicate the warehouse's current operating logic. As previously explained, the only existing storage rule is that each client's products must be stored in the aisle or aisles assigned to that client. Depending on the size of the client – that is, the number of positions needed to accommodate their stock – the assigned area may vary: some aisles are shared among multiple clients, others are exclusively assigned to a single client, and certain clients require more than one aisle. Within these designated areas, the exact storage position of each product is left to the discretion of the warehouse operator.

In the second scenario, *Owner w/ Reserve*, the client-to-position assignments remain unchanged. However, a layer of optimization is introduced within each area by organizing products according to their COI. Specifically, within each client's zone, products are sorted by COI so that the most critical items (i.e., those with the lowest COI) are placed in the most accessible positions. This approach allows for more deliberate and structured storage

decisions, reducing reliance on operator intuition, while still maintaining the organizational structure familiar to the warehouse staff.

In this scenario, each client area is divided into a picking zone and a reserve zone, with the picking zone occupying the most accessible positions and the reserve zone located farther away. As previously explained, the idea is that by picking from a smaller, more concentrated area, travel distances are reduced and picking becomes more efficient. However, it is important to ensure that the time saved through more efficient picking is not offset by the additional time required for replenishments, which consist of moving stock from the reserve zone to the picking zone as needed.

To assess this trade-off, we introduce a third scenario: *Owner w/o Reserve*. This scenario retains almost all features of the previous one: client-to-position assignments and COI-based organization, but removes the division between picking and reserve. In other words, picking is performed across the entire client area. This leads to a larger picking zone, which may increase the average time per pick due to longer travel distances but eliminates the replenishment effort. It is possible that the net time savings remain positive, which justifies the need to simulate this scenario as well.

In the fourth scenario, *No Owner*, the constraint of assigning clients to specific aisles is removed entirely. Products are now placed throughout the warehouse based solely on their COI, regardless of which client they belong to. This allows for full flexibility in product placement and ensures that the most critical products across all clients are stored in the most accessible locations.

The rationale behind these three alternative scenarios lies in the hypothesis that leveraging COI for product placement can lead to more efficient storage configurations and reduced picking times. In the *Owner* scenarios, the optimization is limited by the client-based zoning, which may restrict its effectiveness. For example, one aisle may contain several high-priority products competing for limited prime locations, while another aisle assigned to a different client might contain mostly low-priority items, resulting in valuable positions being underutilized. This imbalance suggests that removing client zoning, as in the *No Owner* scenario, may lead to a globally more efficient allocation.

However, one important operational factor must also be considered: product complementarity. In this warehouse, each order includes only products from a single client – items from different clients are never ordered together. This means there is a high degree of intra-client complementarity (products from the same client tend to be picked together),

but no inter-client complementarity (products from different clients are never picked together). Therefore, it may be beneficial to store a client's products near each other to reduce the travel time between items during order preparation, even if that increases the average distance between the client zone and the shipping dock.

Later, through simulation, we will evaluate which of these strategic factors has the greater impact on reducing picking time: the proximity between complementary products from the same client, or the optimal placement of high-priority products through a globally optimized, COI-based allocation.

Now that the scenarios to be explored have been introduced, we will examine in detail the methodology used to construct each one.

4.3.1 Baseline Scenario

We begin with the *Baseline* scenario, which was designed to replicate as accurately as possible the warehouse's current storage and picking process. This scenario is essential, as it serves as a reference point for assessing the efficiency gains of the alternative configurations. In doing so, it allows for a clear quantification of improvements relative to the warehouse's current operations.

In this scenario, there is no distinction between picking and reserve positions, as this concept does not exist in the present system. Accordingly, the total number of positions required for each client was used directly.

As previously explained, in the current operation, clients are assigned to specific aisles. However, this assignment is not one-to-one: some aisles are shared by multiple clients, and some clients require more than one aisle to store their stock. Therefore, the first step was to obtain the mapping of clients to their respective aisles. Then, based on these relationships, we defined operational zones, understood as sets of positions that can serve the same client(s). For example, a client occupying two aisles would be associated with a single zone comprising those two aisles. Conversely, if three clients share the same aisle, that aisle becomes a single zone to which all three clients are assigned. This approach ensures that each client and each aisle is uniquely linked to one zone.

This system was crucial to ensuring a balanced and realistic allocation, avoiding two risks: assigning all prime locations in a shared aisle to a single client (neglecting others), or inefficiently filling one aisle completely before using the next one, even when both are part of the same client's storage space.

To generate the storage plan, we began by getting the number of positions required by each client, based on the historical average observed in the last month. Then, for each zone, we aggregated the position needs of all the clients within that zone. This gave us the total number of positions that should be used within each zone in this scenario. The next step was to identify which specific positions to use, based on their location codes.

Initially, the idea was to randomly assign the required number of positions within each zone. This approach reflects the current operational logic of the warehouse, where operators are free to store products in any available position within the designated aisle, resulting in a non-systematic and approximately random product distribution.

However, field observations and analysis of historical stock data indicated that this randomness is not entirely uniform. In practice, operators tend to favor lower rack levels, which are easier to reach and require less physical effort during daily tasks. This hypothesis was confirmed through a review of storage patterns, which showed that occupancy rates decrease with increasing rack height.

To incorporate this behavioral tendency and further improve the realism of the scenario, we introduced probabilistic modeling: rather than assigning all positions with equal probability, positions were still selected randomly but weighted according to the historical occupancy distribution by level.

This adjustment ensures that the *Baseline* scenario aligns not only with the warehouse's formal policies but also with implicit behaviors of its operators, resulting in a more faithful representation of the real-world situation.

In addition to the preference for lower racks, it is reasonable to hypothesize that operators may also rely on their experience to keep frequently picked products closer to hand. While this behavior could introduce an additional source of non-randomness in product placement, the effect of such informal adjustments would be hard to model and is unlikely to be systematic, as they are made by different operators and rely on intuitive, momentary judgments of product popularity rather than on accurate or consistent data.

The final output of this process is a set of positions for each zone in which the products are assumed to be stored. Picking is permitted from any of these positions, since there is no separation between picking and reserve areas in this scenario.

This plan will be used during the simulation phase to establish the benchmark against which all alternative strategies will be evaluated.

4.3.2 Owner w/ Reserve

The *Owner w/ Reserve* scenario maintains the current assignment of clients to specific warehouse zones but introduces an additional layer of optimization by organizing products within each zone based on their COI. Moreover, it incorporates the division between picking and reserve zones. This scenario allows for a more structured and systematic organization of the warehouse while preserving the familiar operational layout for warehouse staff.

In this scenario, COI classifications were applied within each client. Specifically, for every client, the 20% of products with the lowest COI values were classified as Class A, the next 30% as Class B, and the remaining 50% as Class C. Using this classification, along with the previously calculated number of picking and reserve positions required for each product, it was possible to determine the total number of Class A, B, and C picking and reserve positions needed by each client.

Note that the final numbers were adjusted so that the total number of positions allocated to each client matches the historical average observed in the last month. This is important to account for distortions introduced by seasonal products, as previously explained. This adjustment was made proportionally, meaning that the relative distribution of positions across classes (A, B, and C, for both picking and reserve) was preserved, while the overall total was scaled to align with historical values. This ensures that the storage plan reflects recent operational reality while keeping the class proportions derived from the COI-based calculations.

Based on the established mapping between clients and zones (which mirrors the same client-area allocation used in the *Baseline* scenario), the total number of required positions for each class was then aggregated per zone.

Next, the storage positions in each zone were assigned to classes using a constructive algorithm based on the travel time from each position to the shipping dock ("EXP"). The assignment process within each zone followed a sequential, time-based logic.

All positions in the aisles associated with a given zone were first identified by their position codes. The previously calculated travel time from each of these positions to the dock area was then used to guide the assignment process.

The procedure began by allocating the most accessible positions – those with the shortest travel times – to fulfill the Class A picking requirements. Once all Class A picking

positions had been assigned, the next most accessible positions were used to meet Class B picking, followed by Class C picking.

After all picking requirements had been fulfilled, the same logic was applied to assign reserve positions: the best remaining positions were assigned to Class A reserve, followed by Class B reserve, and then Class C reserve.

Any remaining unassigned positions – typically those furthest from the dock and least accessible – were labeled as Class D reserve by convention. These positions are not expected to be used under normal occupancy conditions but are included in the storage plan to ensure operational flexibility, especially during peak periods when more reserve space than anticipated may be needed. These positions thus serve as a safety buffer.

This sequential allocation process was repeated for each zone, resulting in a complete storage plan for this scenario. The plan was saved in a structured table containing the following columns: position code, class (A, B, C, or D), and position type (picking or reserve).

This scenario assumes that all picking activity is performed exclusively from the designated picking positions, with internal replenishment carried out from the reserve to the picking area to ensure that it remains adequately stocked with the planned quantities. The replenishment effort required over time is accounted for in the simulation, with the respective section detailing the calculation method. Incorporating these assumptions into the simulation model is essential to accurately capture the operational implications of this strategy.

4.3.3 Owner w/o Reserve

The next scenario, *Owner w/o Reserve*, is similar to the previous one in that it retains the current allocation of clients to predefined warehouse areas and uses COI-based classification to organize products within those areas.

The key difference, however, is that no distinction is made between picking and reserve zones. In this configuration, picking can be performed from any position, and no internal replenishment operations are required.

For this scenario, the COI calculations were based on the assumption that no protection period exists. Consequently, each product is assigned a number of picking positions equal to its total required positions, and the number of reserve positions is zero. As in previous scenarios, the number of positions in each class was adjusted so that the total number of positions allocated to each client remains consistent with historical values, while

still preserving the proportions of positions across classes. Each product is also assigned a COI class as previously defined (A, B, or C).

From this point onward, the logic used to construct the storage plan mirrors that of the previous scenario. First, the number of positions required in each class (A, B, and C) is calculated per client. Next, based on the client-to-zone allocation (identical to that used in the *Baseline* and previous scenarios), the total number of positions required per class is aggregated for each client zone. Finally, using the previously calculated travel times to the shipping dock, positions in each area are allocated constructively: the most accessible positions in each area are assigned to Class A, the next most accessible to Class B, and finally the required positions to Class C are assigned.

Since this scenario does not include a reserve zone, any positions that remain unassigned after allocating the required Class A to C positions are automatically classified as Class D picking positions (instead of Class D reserve, as in the previous scenario). The practical implication of this classification is that, in the event of high warehouse occupancy requiring use of Class D positions, picking will be performed from these positions as well, given that no internal replenishment processes are in place to relocate stock closer to the dock before picking.

After processing all areas using this logic, the final storage plan for this scenario was generated and stored in a table following the same structure used in previous plans: position code, COI class (A, B, C, or D), and position type (in this case, all entries are labeled as picking, but the column is retained for consistency across scenario plans).

4.3.4 No Owner Scenario

The *No Owner* scenario introduces the highest degree of flexibility among all proposed configurations. In this approach, the constraint of assigning clients to specific aisles is completely removed, allowing all products to be stored in any aisle regardless of their ownership. This unrestricted setup enables a globally optimized allocation, where the most strategically important products, based on their overall COI, can be placed in the most accessible positions across the entire warehouse.

In this project, the *No Owner* scenario was only tested without a reserve area. The alternative configuration of dividing picking and reserve was not considered, following a deliberate decision aligned with the warehouse management's perspective. The reason for this exclusion is the high likelihood of congestion in the picking zone if such a policy were

applied. Under this configuration, all class A products across the warehouse would compete for a small area of limited positions near the shipping dock. This would inevitably concentrate the picking activity of the most critical items in a restricted area, generating excessive traffic and operational bottlenecks. Moreover, replenishment tasks would also need to be performed in the same area, further intensifying congestion.

Although congestion effects are a relevant operational dimension, they could not be incorporated into the simulation model due to the lack of detailed data on picker movements and simultaneous activity. Consequently, rather than testing a scenario that was judged unrealistic and operationally unfeasible, we followed the operational sense of warehouse management and excluded *No Owner w/ Reserve* from the set of tested configurations. This ensures that the analysis remains focused on scenarios that are both technically sound and practically viable.

The storage plan for this scenario was based on the COI calculations previously performed. It's worth noting that, while the numeric COI value of each product remains the same whether calculated within a client or across all products, the ABC classification differs when performed globally. In this case, the top 20% of all products with the lowest COI values, regardless of ownership, are classified as Class A. Similarly, the next 30% and the remaining 50% are assigned to Classes B and C, respectively. In theory, all top-ranked products could even belong to a single client.

The assignment of storage positions followed the same constructive, time-based logic used in earlier scenarios but now applied to the entire warehouse without any client-specific constraints. First, the best positions of the warehouse, those closest to the shipping dock, were allocated to Class A. Once Class A needs were satisfied, the next best available positions were assigned to Class B, followed by Class C. Any positions left unassigned after this process were labeled as Class D.

Unlike previous scenarios, this process was performed only once, since there is no subdivision of the warehouse into client-assigned areas. The resulting plan covers the entire warehouse and was stored in the standard format used for the other scenarios: position code, COI class (A to D), and position type (in this case, and for the reasons previously described, only picking).

This is clearly the most disruptive scenario relative to current warehouse operations and layout, as it departs entirely from the client-based logic that operators are accustomed to. It is therefore essential to carefully evaluate the simulation results of this scenario to

analyze the trade-off between globally optimized product placement and the benefits of keeping complementary products closer together.

Should the simulation indicate that this scenario leads to the lowest average picking time – and is thus selected as the macro plan – change management becomes a critical success factor. It will be necessary not only to clearly communicate the long-term benefits to warehouse staff, namely, reduced picking time, but also to plan and manage the initial transition, which may require substantial reorganization efforts to be carried out alongside regular warehouse operations.

4.4 Simulation

4.4.1 Objective and General Approach

After generating the different scenarios (hereinafter also referred to as storage plans), a simulation was built with the objective of evaluating the performance of each one by comparing the average time required to pick an order under each plan. This comparison will allow us to identify the most efficient strategy and, consequently, select the corresponding storage plan as the macro plan, that is, the plan to be adopted.

The approach used consisted in simulating the routes that operators would have taken to prepare the orders received over the past year, assuming that the products were stored according to each of the proposed storage plans. Travel and picking time functions were then applied to calculate how long each route would take, thereby determining which plan would result in faster routes on average. The use of real historical order data ensures that the results obtained are relevant and adapted to the reality of this warehouse, and using the same set of orders across scenarios guarantees the comparability of results.

The following subsections will detail the complete simulation construction process, including how product positions were defined, the assumptions considered during route generation, the estimation of route duration, and the calculation of each scenario's performance.

4.4.2 Product-to-Position Assignment per Scenario

To generate the picking routes for each order, it is necessary for every product to be assigned to one or more specific positions in the warehouse, in accordance with the logic and assumptions that characterize each of the scenarios. Only then is it possible to identify the

locations to be visited in each route. Therefore, product-to-position assignments were simulated for each scenario. It should be noted that this assignment will only be used for the simulation itself, since, as explained earlier, the actual storage plans follow a class-based policy rather than a fixed-position policy, where each product would be permanently assigned to specific positions.

The way in which the selected storage plan will be implemented in practice will be presented in detail when the use cases are described, as they will support that implementation. However, in order to clarify some of the assumptions used in this product-to-position assignment process, it is relevant to anticipate a few explanations.

When the use cases are in place, upon receiving a product in the warehouse, operators will scan its barcode using their Personal Digital Assistant (PDA), and based on the product's information, namely, its owner and COI class, the eligible positions for storage will be identified. For example, in the *Owner w/o Reserve* scenario, suppose a product j belonging to $Owner_j$ and classified as COI class A arrives. The eligible positions for this product will be those in the area assigned to that owner that were marked as class A in the storage plan. Once the eligible positions are identified, the algorithm will select one that is currently either empty or, not being empty, still has space to accommodate product j, and this position will be shown on the operator's PDA so that the product can be stored there.

Therefore, the distribution of incoming products across eligible positions will depend on the stock status at the time of storage, a factor that is highly volatile due to the constant inflow of new products and the outflow of products being picked for orders throughout the day. Consequently, the distribution of products across the eligible positions will tend to follow a random pattern over time. This behavior was taken into account during the position assignment process, which must be simulated in order to generate the picking routes.

Now, explaining how the product-to-position assignment was performed in each scenario, according to its specific assumptions: in the *Baseline* scenario, which aims to replicate the current reality of the warehouse, products were randomly assigned to the positions selected in the storage plan. The number of positions assigned to each product corresponds to the number previously calculated during the COI computation. As a reminder, in this scenario there is no distinction between picking and reserve zones, nor any ABC classification of products or positions. The random assignment reflects the current storage logic, where operators are free to choose where to place products within the area associated with each client.

In the *Owner w/ Reserve* scenario, products are stored in the areas assigned to their respective clients, but here it is important to account for both the picking/reserve distinction and the COI class. Previously, products were classified into a COI class based on intra-client analysis. In addition, the number of required picking and reserve positions had already been calculated for each product. Using this information, and the corresponding storage plan, eligible positions were identified for each product, that is, positions located in the client's area, of the same class, and with the same function (picking or reserve). Among these eligible positions, the required quantity was randomly assigned to each product. This random allocation within the eligible positions reflects the logic of storage that will happen in the operation: products will be stored in positions of their class based on each position's capacity at the time of storage.

In the *Owner w/o Reserve* scenario, the client-to-area structure and the ABC classification are maintained, but there is no picking/reserve division. Thus, the identification of eligible positions does not need to account for that distinction. The number of positions to assign is equal to the total number of positions required per product, and, again, the allocation within eligible positions was done randomly.

Finally, in the *No Owner* scenario, the logic of assigning clients to specific warehouse areas is abandoned. Products can be stored anywhere in the warehouse, regardless of the client to which they belong. In this case, the previously defined global ABC classification was used, along with the previously computed number of positions per product. Products were then randomly assigned to positions matching their class throughout the entire warehouse, without client-related restrictions.

Once again, it is important to emphasize that this random assignment within each class is intentional and reflects the expected behavior once the new storage plan is implemented. In real operations, products will be stored in positions belonging to their class, based on real-time availability. Therefore, the assignment model adopted here aims to realistically and consistently represent these dynamic operational decisions.

4.4.3 Route Generation

After assigning specific storage positions to each product for every scenario, we move to the next step: defining the picking route, which requires determining the exact position from which each product will be picked. Since a product may be stored in multiple positions, it

was assumed that the picking location is chosen randomly from among the available picking positions for that product.

This approach is aligned with the level of granularity used in the simulation and provides a neutral representation of what would happen in the actual operation: the WMS will indicate to operators the specific positions to visit based on the expiration dates of the units (following the FEFO policy). However, since this simulation is based on an average representation of stock over time and not on real daily snapshots of inventory (which would contain detailed expiration data), it is not possible to determine exactly which positions would be selected by the WMS at any given moment.

The use of randomness, restricted to picking positions only, reflects this uncertainty in a neutral way. Over time, picking will occur across the various positions containing the product, depending on which units have the earliest expiration date at each moment. Moreover, this ensures neutrality in scenario comparison: by applying the same random selection logic across all scenarios, we guarantee that any differences in simulation results are due solely to differences in the storage plans themselves and not to variations in the picking model.

Once the picking position for each product has been identified, the next step is to define the sequence in which positions are visited within each order. This sequencing decision is known as the routing policy, and it is considered out of scope for this project.

Therefore, the policy used in the simulation is the one currently applied in the warehouse: the return routing policy. In this policy, the picker moves through the aisles sequentially, entering only those where picking is required. After visiting all the necessary positions within an aisle, the picker exits through the same end of the aisle they entered from, proceeding to the next aisle that contains items to pick. Within each aisle, the picker first visits all the required positions on one side, up to the furthest one, and on the way back, performs the picking on the opposite side. This routing policy is illustrated in *Figure 4*.

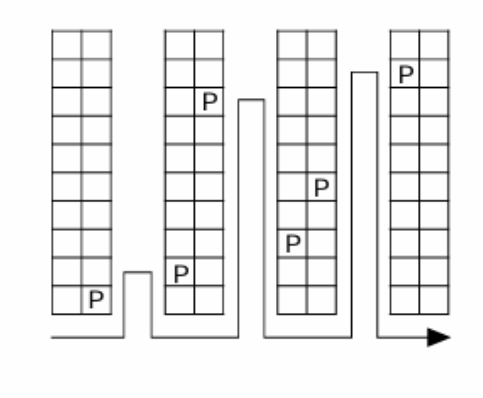


Figure 4 - Representation of the return routing policy. Reprinted from "An evaluation of order picking routeing policies", by C. G. Petersen, 1997, International Journal of Operations & Production Management, 17(11), pp. 1098–1111

Using the same routing policy as the one in place during real operations is essential to ensure that the simulation faithfully replicates actual operator movement patterns, a critical aspect, since the sequence of picking has a direct impact on the total travel time.

In addition to defining which picking positions to visit, it was also established that each route starts and ends at the shipping area. This assumption is consistent with the actual operation: when operators begin picking, they first collect their picking equipment or forklifts, which are parked in the shipping zone. After retrieving the equipment, they perform the picking route and return with the picked items to the shipping area, where the order will be staged for shipment, before starting the next picking route from that same zone. As such, all routes, including the first and subsequent ones, begin and end at the shipping area.

This process of generating routes for historical orders was repeated across all scenarios, resulting in a set of routes that would have been performed if each plan had been implemented. These routes will then be used in the next step to estimate the time required to prepare each order under the respective storage scenarios.

4.4.4 Estimating the Duration of Picking Routes

To calculate the time required to execute each picking route, the time measurement data previously collected during warehouse operations, already used to estimate the time between "EXP" and each storage position, was leveraged once again. However, when applying these functions to full picking routes, it is important to include additional components that reflect the effort involved in specific tasks performed during the product retrieval process.

Recalling the time functions already used to estimate the time between each storage position and the shipping area, they provided: the travel time on the warehouse floor between positions, the retrieval time to remove a pallet from its rack level, and the storage time to return the pallet to its original location. Now, two new components are introduced.

The first new component is the product picking time, which accounts for the task of actually picking each product. That is, after retrieving the pallet containing the item needed, the operator must transfer the required units onto the pallet he is using to carry the products during the route. This time naturally depends on the quantity of the product being picked in that order. During on-site measurements, both the duration of this picking action and the number of units picked at each position were recorded. From these observations, it was possible to calculate the average picking time per unit, leading to the following estimation function: $product\ picking\ time\ = 11.9333 \times number\ of\ units\ to\ pick$.

The time required to handle products can vary, as it is also influenced by product characteristics such as weight and volume. However, this level of detail was not included in the time model, since product weight and volume data is not available in the warehouse management system. Nonetheless, care was taken during the warehouse time measurements to include a wide range of products with diverse typologies, sizes, and weights, ensuring that the coefficient obtained would reflect the variability found in practice.

The second additional component included in the model is the marshalling time, which refers to the final task performed at the end of each picking route. Once the operator completes a route and returns to the shipping area with a pallet containing the picked items, they must stabilize the load. This involves placing the pallet on the ground and securing it with plastic film to ensure the stack is safe and stable, preventing items from falling. Only after completing this task can the operator begin a new route. To account for this effort, marshalling time was measured and incorporated into the simulation. The average time recorded for this step was 37.25 seconds, which is added to the total duration of each picking route.

Thus, to estimate the total time for a picking route, the following components are summed: travel time between consecutive positions on the ground; retrieval and storage time for each pallet, considering the height of the corresponding position; product picking time, based on the number of units to be picked at each stop; marshalling time, added at the end of the route.

This procedure was repeated for all historical orders, across all scenarios. In the end, by summing the times of all routes in each scenario, we obtain the total predicted time required to prepare the orders from the past year. This will allow a fair comparison of the relative performance of the different storage plans based on a common KPI: the average order picking time.

4.4.5 Replenishment Effort in Scenario with Reserve

A key factor that must be considered to ensure fair comparability between scenarios is the internal replenishment effort. As previously described, one of the scenarios implements a division between picking and reserve zones. In this scenario (*Owner w/ Reserve*), only a subset of each product's inventory is stored in the picking zone while the remaining stock is stored in less accessible reserve positions. This strategy is intended to shorten the distance traveled during picking, making routes faster.

However, to maintain this operational model, it is necessary to periodically replenish the picking zone with units from the reserve zone. This process is known as internal replenishment, and it introduces an additional operational cost. To fairly assess and compare the total picking effort across scenarios, it is essential to include this replenishment effort in the simulation. Without it, scenarios with a picking zone might appear more efficient simply because the additional time required to keep that zone stocked is not accounted for. In reality, the time saved by picking in a smaller zone might be outweighed by the additional time required for replenishing that zone.

To determine the total internal replenishment effort during the simulation period, two components had to be calculated: the number of replenishment operations and the time required per replenishment. Starting with the number of replenishments, it is important to recall how the picking zone was sized: it was designed to hold enough stock of each product to cover expected demand for one week, the defined picking protection period. After this sizing, a refinement was applied to calculate the real protection period, assuming that partially filled picking positions were topped up with additional units of the same product. The rationale is that picking positions are highly valuable, and their storage capacity should be fully utilized.

This adjustment impacts the expected number of replenishments. By increasing the quantity stored in the picking zone, the stock held there is expected to last for more than one week. Without this adjustment, the replenishment rate would simply be one per position per

week. With it, the expected number of replenishments depends on the product's real protection period.

Thus, to calculate the expected number of replenishments per week for each product we used the inverse of the real protection period because it represents how frequently a picking position will need replenishment. Multiplying this rate by the number of picking positions assigned to the product gives the total expected number of replenishments per week for that product across the warehouse. This logic ensures that the replenishment effort is scaled both by how long the stock lasts and by how many positions are being maintained for that product. This calculation is modeled on equation 3.

$$number\ of\ replenishments\ per\ week = \frac{1}{real\ protection\ period} \times number\ of\ picking\ positions \quad (3)$$

For example, if a product has a weekly demand of 40 units, but 50 units are stored in its picking zone (fully utilizing one position), the real protection period becomes 1.25 weeks. In this case, the expected replenishment rate is $1 \div 1.25 = 0.8$ replenishments per week, assuming one picking position.

This calculation was repeated for all products to estimate their individual replenishment needs. Since the size and layout of each client's area can vary (and influence replenishment times), the results were then aggregated by client area and COI class. Aggregating at this level allows for more accurate estimations, as it accounts for the specific characteristics of each client and area rather than simply summing replenishments across the warehouse. Finally, to extrapolate this to the full simulation period, the weekly figures were multiplied by 52, giving the expected number of replenishment movements per year for each client area and class.

The second component required was the time needed for each replenishment operation. To estimate this, the following operational logic was assumed: units are picked from a reserve position and transported to a picking position of the same COI class (A, B, or C) and within the same client area.

Based on the storage plan of each scenario, a matrix of average travel times was generated between reserve and picking positions of the same class within each client area. For each class and area, the average replenishment time was calculated using the previously defined time functions. Thus, the total replenishment effort for each pair of client area and

class was calculated as: $total\ replenishment\ time = number\ of\ replenishments \times average\ time\ per\ replenishment$.

This calculation was applied to all combinations. Summing the resulting values gives the total internal replenishment effort for the scenario with reserve, over the one-year period considered in the simulation.

4.4.6 Scenario comparison

Finally, to compare the performance of the different scenarios, the key performance indicator (KPI) used to identify the most efficient scenario was calculated: the average picking time per order.

To obtain a simple average, one could divide the total picking time over the simulation period (as previously calculated) by the number of orders in that same period. However, as previously explained, it would be unfair to make this comparison while ignoring the replenishment effort required in the scenario with a reserve zone.

To ensure the comparison holistically reflects all operational efforts, the total replenishment time for each scenario was added to its corresponding picking time. Only then was the total time divided by the number of orders. The resulting value, in addition to reflecting the time spent on the actual picking routes, also incorporates the replenishment time required to implement a storage strategy with a split between picking and reserve zones, thus guaranteeing a fair and balanced comparison across scenarios.

4.5 Use Cases

In this section, the two Use Cases (UC) will be presented. They are responsible for translating into practice the storage plan identified by the simulation as the most efficient. Although both UCs share the ultimate goal of increasing picking efficiency, they address different stages of the product lifecycle within the warehouse and therefore complement each other.

UC1 assists operators in storing incoming products by suggesting the most suitable position, taking into account both the type of product and the current state of the warehouse. UC2 focuses on products that are already stored. By suggesting the relocation of products to other positions, this UC increases the alignment between the actual storage arrangement and the macro plan. In scenarios with a separation between reserve and picking zones, this UC

also suggests internal replenishment movements, allowing units to be transferred to the picking area in order to maintain adequate stock levels.

To measure the alignment between the actual storage arrangement and the macro plan, the layout efficiency metric was established. In the warehouse context, a support is the basic storage unit used in the WMS to track stock. Each support corresponds to a batch of a single product, identified by a unique alphanumeric barcode. This barcode uniquely links the support to one product code and one expiration date, meaning that a support can never contain multiple products or different expiration dates. In practice, each pallet can contain multiple supports, i.e., multiple products, and throughout the warehouse there can be several supports of the same product. The defining characteristic, however, is the one-to-one relationship between the barcode and a homogeneous batch of the product.

Layout efficiency corresponds to the ratio between supports stored in positions that match their product's COI class and the total number of supports; it can be interpreted as the percentage of supports correctly stored.

This will be a key metric to assess macro plan alignment over time and will allow the impact of the use cases to be evaluated, both through the proactive suggestions of UC1 and the corrective suggestions of UC2.

4.5.1 Use Case 1

4.5.1.1 Context and Objective

Use Case 1 is triggered whenever new products arrive at the warehouse and need to be stored. Its goal is to ensure that storage is carried out in line with the macro plan, enabling a systematic and efficient allocation of products that reduces future picking costs. Through this use case, when operators are about to store a product, a specific location is recommended to them to ensure full alignment with the storage strategy previously identified as the most efficient.

The recommendation provided by the algorithm depends on the storage positions currently available and takes into account the product's COI class, the owner area it belongs to (in scenarios where such divisions exist), and the type of position required (picking or reserve). It also incorporates the operational constraint of not placing units of the same product with different expiration dates in the same position. Additionally, safety constraints are considered to ensure feasibility, such as weight limits for upper-level positions and the

separation of food and toxic products. The recommendation is communicated to the operator in real time via the PDA, ensuring that using this use case does not disrupt the operational flow.

4.5.1.2 Position Recommendation Algorithm

When a product arrives at the warehouse, it is unloaded from the truck and immediately assigned an internal barcode. This barcode allows the WMS to access a range of product-related information, such as product code, owner, and expiry date. When the operators responsible for storing the product scan its barcode using a PDA, all the product information is sent to the UC1 algorithm.

Let's now walk through what happens in this use case from the moment the product information is received to the moment a storage location is recommended. First, using the product code, the algorithm queries the database to retrieve the COI classification assigned to the product, as defined in the current storage scenario. By combining this classification with the active macro plan, it becomes possible to identify eligible locations for storing the product, that is, locations that belong to the correct owner area, match the product's COI class, and have the appropriate function type (picking or reserve, if applicable).

Identifying locations that match the customer and COI class is straightforward, as it simply involves filtering for locations that meet both criteria. However, deciding whether to store the product in a picking or reserve location requires a more nuanced assessment based on the current state of the warehouse. In scenarios without picking-reserve separation, all locations are considered picking, so this decision doesn't happen. In scenarios with reserve zones, the algorithm must assess whether the existing quantity of that product in the picking area already covers the defined protection period. If it does, the new units are directed to reserve. If not, it may make sense to store the new units in picking but only if there are no units in reserve with an earlier expiry date.

This is because, when a picking route is later generated, the WMS selects product locations based on units with earliest expiry dates. If units with earlier dates are stored in reserve, then picking will occur there. To prevent this inefficiency, the algorithm will send the new units to reserve if they have later expiry dates than the ones already in reserve. Those older units can then be moved to the picking area during the next internal replenishment cycle. However, if there are no units in reserve or if they have a later date, then the new units of that product can be sent to picking positions, until they fulfill the protection period stock.

Once eligible locations have been identified, the algorithm filters them based on their current availability. This availability depends not only on how full the locations are but also on the operational constraints in place. There are four types of constraints to consider: weight limits, incompatibility restrictions, height limitations and expiration date conflicts.

Weight limits apply because upper-level slots should not exceed a defined maximum load for safety reasons. To assess whether a location can accept the new product, the algorithm calculates the total weight already stored there, based on the product weights and quantities, and checks whether adding the incoming units would surpass the limit. If so, that location is deemed unavailable.

Incompatibility restrictions exist because food and toxic products must not be stored together. To handle this, when evaluating locations for a product classified as toxic – this classification is available in the database – any location already containing food products is excluded. Conversely, when storing food products, locations that contain toxic materials are also excluded.

Height limitations apply to fragile products, such as, those in glass packaging, which can only be stored up to a certain height level for safety reasons. Information on which products are considered fragile is also available in the database, allowing the algorithm to automatically exclude any locations above the permitted level for those products.

Then, the algorithm also considers the constraint related to expiration dates: units of the same product with different expiration dates cannot be placed in the same position. Therefore, any location already containing units of the same product with a different expiration date is automatically excluded from the eligible options.

Once all constraints are taken into account, the algorithm must assess whether the remaining candidate locations have enough free space to store the product. While empty locations can accommodate any product, it is preferable to consolidate products in locations that already contain compatible items but still have available capacity. This approach is in line with the multi-product policy used in the warehouse. It avoids unnecessary dispersion of stock throughout the warehouse and promotes more efficient space usage.

This is one of the major challenges of this use case: as mentioned earlier, there is no volume data available for products. Therefore, a strategy had to be developed to estimate whether a location containing specific products and quantities – information that can be retrieved from the real-time WMS stock – still has enough space for more units.

The solution was to leverage the extensive historical stock data available. In addition to current stock levels, the system holds two years of historical stock data. Based on this historical data, it was possible to estimate how many more units of a product could be stored in a partially filled location. For each product, all historical combinations in which it was stored were analyzed, and the maximum quantity stored was recorded as a function of the number of other products in that location. This analysis enabled the algorithm to estimate how many additional units of a product could likely fit in a location with a specific existing stock configuration. Using these estimates, the algorithm excludes all locations that don't have enough capacity for the incoming product.

If partially filled locations with available capacity exist, they are prioritized to optimize space usage. If none are available, the empty locations are considered. Finally, from the remaining candidate locations, the one with the shortest travel time from the shipping dock is selected as the recommended position.

While this approach allows the system to estimate location capacity using a data-driven method, it does have limitations. Although there may be many data points for each product, they often involve different product combinations, and naturally, their volumes vary. The specific mix of products currently stored in a location might not allow for additional units, and not just because of volume. It could be due to shape incompatibility or how the items fit together.

To address this issue, operators are allowed to reject a suggested location and request an alternative. In that case, the next best candidate is returned. This feature adds flexibility for dealing with unexpected situations such as incorrect product registrations (wrong location or missing product data) or inaccurate product classifications, for example, a toxic product that isn't flagged as such in the system.

Another limitation involves new products, for which there is no historical storage data. In these cases, the system still allows consolidation, acknowledging that some recommendations may be rejected at first. Operators can reject unfeasible suggestions and, if needed, manually assign a location, as long as they register it in the system. Over time, as more data points are collected, the system's consolidation estimates for these products become increasingly accurate, and the algorithm's ability to make suitable recommendations improves accordingly.

Once the final storage location to be suggested is identified, it must be communicated to the operator through the WMS. This requires a highly responsive integration between the algorithm and the warehouse systems, which is detailed in the next subsection.

4.5.1.3 Algorithm Integration

The selected storage location returned by the algorithm is sent to the operator's PDA so the product can be stored accordingly. A critical factor for the successful adoption of this use case is that the storage recommendation is delivered almost instantly after scanning the product's barcode. Otherwise, operators would be forced to wait while the algorithm runs, which would interrupt the workflow and reduce operational efficiency during the storage process.

This requirement means that the algorithm must be highly optimized to ensure fast computational performance, and that the connection between the WMS and the algorithm must have low latency.

To meet these performance demands, the algorithm was developed in Python due to its balance of computational efficiency and integration flexibility, including:

1. **Real-time communication with the WMS** via APIs, enabling seamless integration.
2. **Efficient connection to the database**, where key information such as product and location COI classifications are stored.
3. **Support for parallel and concurrent execution**, which significantly improves processing speed.
4. **Ease of maintenance**, thanks to Python's widespread adoption in the industry.
5. **Scalability**, as the algorithm can be deployed in other warehouses within the company, supported by Python's modular architecture.

This optimized integration not only ensures that recommendations are delivered in real time, but ensures a robust implementation geared towards future scaling of the algorithm's use.

4.5.1.4 Expected Impact

The implementation of this use case is expected to generate significant operational benefits by promoting more efficient storage practices. By systematizing the storage policy and reducing reliance on operator intuition, it becomes possible to lower future picking time.

Additionally, prioritizing the consolidation of products into partially filled locations helps prevent unnecessary stock dispersion across the warehouse and maximizes the use of each storage position. This contributes to a more efficient utilization of the available storage capacity.

The integration of operational safety constraints into the algorithm ensures that recommendations comply with safety requirements and minimizes the risk of errors that could occur due to a lack of product knowledge on the part of the operators.

Thanks to the algorithm's integration into the WMS and its optimized performance, which enables near-instantaneous suggestions on the operator's PDA, this use case fits seamlessly into the operational workflow without disrupting the pace of work. At the same time, it retains flexibility to handle new products or unexpected situations, such as incorrectly registered incompatibilities, ensuring that the system remains robust and adaptable to real-world conditions.

By aligning storage with the strategic plan designed to optimize picking, this use case directly contributes to reducing the average order preparation time, the key KPI targeted for improvement. As such, it has a direct impact on overall warehouse efficiency, generating tangible and long-lasting benefits.

Moreover, the approach used is generalizable, allowing the algorithm to be easily adapted and scaled to other warehouses within the company, even if they have different layouts or product assortments. This scalability broadens the potential positive impact of the solution across multiple logistics units.

4.5.2 Use Case 2

4.5.2.1 Context and Objectives

Use Case 2 aims to enhance the operational efficiency of the picking process through the strategic rearrangement of products already stored in the warehouse. This use case has two main objectives: (1) to ensure that, over time, the picking area maintains the necessary stock levels to fulfill orders for the protection period, without requiring inefficient pickings from the reserve area; and (2) to reduce misalignments between the actual product arrangement in the warehouse and the optimal layout defined by the macro plan. Naturally, in scenarios without a reserve area, this use case focuses exclusively on the second objective.

To achieve these goals, the use case proposes two types of movements. Internal replenishment movements, in which units are transferred from the reserve area to the picking area, prioritizing products with the greatest shortages in picking; and permutation movements, in which the contents of two positions are swapped by directly exchanging the pallets in each position, allowing products to be stored in locations that are better aligned with their COI class and thus improving compliance with the macro plan.

Use Case 2 is triggered on demand by the warehouse's operational management and can be accessed through a dedicated page in the desktop version of the WMS, designed for managerial use. On this page, the user can generate rearrangement suggestions and, based on the results, which include both movement types, review key metrics for each suggestion, such as the estimated execution time and the expected impact on layout efficiency. Filters can also be applied to the generated suggestions, such as filtering by movement type, owner, or aisle, enabling the selection of movements that can be executed in harmony with the daily operational flow of the warehouse. This ensures that Use Case 2 can be carried out without disrupting ongoing activities, such as picking or storing incoming products.

After reviewing and selecting the suggestions to be executed, the operations manager can confirm the operation within the UC2 page. Once confirmed, the corresponding tasks are automatically sent to the PDAs of the operators responsible for physically moving the pallets. This workflow ensures that strategic decisions are made centrally while execution remains fast, controlled, and fully integrated into the existing WMS.

4.5.2.2 Rearrangement Suggestion Algorithm

The algorithm for Use Case 2 was developed to generate suggestions for the two types of movements described earlier. While the underlying logic for generating recommendations for each type of movement differs, there is a sequential structure that integrates the generation of both, as the required inputs are similar.

The first step is to load all necessary data: the warehouse's current stock, including the position codes, the product codes stored in each position, their quantities, and expiration dates. This real-time stock information can be extracted from the WMS. In addition, the algorithm requires the tables previously generated during this project that contain the active macro plan, in which each position is associated with a COI class, a function type, and, if applicable, an owner; as well as the table containing the COI classification of each product (including the recommended quantity of that product that should be stored in picking

positions – information used to compute the COI numerical value). These tables can be accessed from the database. Based on this data, it will be possible to compare the actual warehouse arrangement with the optimal configuration defined in the macro plan.

Starting with internal replenishment movements, the goal is to ensure that the picking area contains enough stock of each product to fulfill orders for the protection period. In these movements, units are transferred from the reserve area to the picking area. A key operational requirement set by warehouse management is that only full-pallet movements should be made. In other words, products in each position are stored on a pallet, and when positions are exchanged, operators will move the entire pallet, transferring all products it contains. This requirement was established because moving individual products within a pallet would be highly inefficient, resulting in granular movements whose operational cost would outweigh their benefits. Based on this, internal replenishment movements are designed to strategically suggest transferring pallets from reserve positions to empty picking positions, prioritizing products with shortages in the picking area, i.e., products whose current picking quantity is below the required level for the protection period.

To identify and rank the suggested transfers, the algorithm first calculates, for all products stored in the warehouse, the total quantity currently in picking positions. It then checks the quantity that should be in picking (previously computed as the amount needed to cover orders during the real protection period). By comparing these two values, products with shortages in picking are identified and flagged for internal replenishment. However, as there may not be enough available picking positions to replenish all shortages, the algorithm prioritizes products using the ratio between actual picking quantity and the ideal picking quantity. This ratio can be interpreted as the percentage of the protection period that the current picking stock of a product is expected to cover.

The algorithm then enters an iterative cycle: it starts with the product with the lowest ratio and checks whether there are units available in reserve. If no units are available (e.g., due to stockout or discontinuation), replenishment is not possible, and the algorithm moves to the next product with the lowest ratio. If units are available, it identifies the position holding the batch with the earliest expiration date, flags that position for replenishment, and adds it to an ordered list. Next, it recalculates the ratios for all products, assuming that this pallet will be transferred to picking – note that additional products, possibly also in shortage, may be moved along with the originally selected product because they were in the same pallet. The updated ratios may change the priority order, and the cycle repeats, continuously adding

positions to the ordered list. Positions added to the list are no longer available for selection in subsequent cycles. This process continues until all shortages are addressed or there are no more eligible positions to move. The result is a prioritized list of reserve positions whose contents should be transferred to picking positions, ranked by urgency based on the lowest coverage.

Next, the algorithm identifies the available picking positions in each customer area (if such division applies in the current scenario) and within each COI class. It then begins assigning destination picking positions to the reserve positions from the ordered list, starting with the top priority. For each reserve position, it searches for available picking positions within the same customer area and COI class that also meet operational constraints**,** specifically weight and height restrictions. The pallet's stored weight must not exceed the capacity of the target position's level, and if height restrictions apply to certain products in the pallet, only levels meeting those restrictions are considered. Note that, since replenishment movements transfer pallets from reserve to empty picking positions, expiration date constraints are not relevant in this context, as the combination of products doesn't change.

If at least one suitable position is found, the closest one to the shipping dock is selected as the destination. If no suitable position exists, replenishment for that reserve position is deemed infeasible for that execution of the use case. This process repeats for all flagged reserve positions. At this stage, positions without an assigned picking destination are excluded, resulting in a list of reserve and picking position pairs for internal replenishment movements, including only those that are feasible at that moment.

The algorithm then proceeds to generate permutation suggestions, which involve swapping the contents of two occupied positions to improve layout efficiency, specifically, to increase the number of supports stored in positions matching their COI class. Like internal replenishments, these swaps are executed with full pallets. One operational limitation of this decision is that a pallet containing products from multiple COIs can never be perfectly aligned with any single COI position. However, over time, as UC1 is applied to incoming products and older products are shipped out, positions are expected to naturally converge toward containing only one COI class, making this issue negligible.

The first step in the permutation stage is to exclude positions already involved in internal replenishment movements, avoiding conflicts between the two movement types. The algorithm then analyzes the current stock and identifies positions where all products are

correctly aligned with the position's COI class. These positions are already optimal and are excluded from permutations. The remaining occupied positions form the candidate pool.

All possible pairs of candidate positions are then generated and operational constraints are applied to exclude swaps that disregard them:

1. **Weight constraint:** The pallet weight in each position must be within the allowable limit of the level in the other position in the pair. If at least one move in the pair exceeds weight limits, the pair is discarded.
2. **Height constraint:** Positions containing products with height restrictions are excluded from swaps that would place them in levels exceeding those limits.

In permutation movements, expiration date constraints do not need to be considered, as the contents of the positions are exchanged in full, preserving the existing product combinations.

Next, the impact on efficiency is calculated for each remaining pair. To do that, the algorithm simulates swapping their contents and measures the change in layout efficiency, assuming no other stock movements occur. Pairs are sorted in descending order of impact, and any with negative impact (i.e., efficiency would decrease) are discarded.

At this point, multiple pairs may involve the same position: remember that the initial list of pairs was obtained by combining each candidate position with all the others.

However, in the final set of suggestions, each pallet should be moved at most once. This decision resulted both from technical limitations of the WMS and from operational considerations discussed with warehouse management. First, the WMS only supports direct transfers between two positions. Therefore, if a pallet were included in multiple swaps within the same UC2 execution, operators would have to perform consecutive manual movements to achieve the intended placement. Breaking down triangulated swaps into a sequence of direct transfers would significantly increase operational complexity and the risk of execution errors, such as incorrect placement or inconsistencies in stock traceability.

Second, it is important to remember that UC2 must coexist with regular warehouse operations, and allowing multiple movements per pallet in a single execution would create longer interruptions and greater overlap with ongoing activities. By restricting each pallet to a single movement, suggestions remain simple, easier to plan, and executable within the limited time windows available, without disrupting the broader workflow. This approach ensures a balanced compromise between improving layout efficiency and maintaining operational feasibility.

To enforce this decision, the algorithm starts by keeping the highest-impact swap first and removes all other swaps involving either position, it then does the same for the next best swap remaining and repeats the process until no positions are duplicated. The remaining swaps form the final permutation suggestions.

4.5.2.3 Algorithm Integration

After executing the algorithm described, two lists of movements are obtained: internal replenishment movements and permutation movements. These lists are merged into a single table, which is presented to the user responsible for deciding which movements to execute.

In the resulting table, internal replenishment movements are always shown before permutation movements. They are prioritized because failing to perform them could lead to executing pickings from reserve positions, which is worse than picking from an incorrect COI class within the picking area. Remember that all picking positions in a given area are better than the best reserve position in that same area, in terms of travel time from the shipping dock.

Within the replenishment list, movements are displayed in descending order of product shortage, which corresponds directly to the order in which these suggestions were generated. Within the permutation list, swaps are ordered in descending order of their impact on layout efficiency, calculated as described in the previous subsection.

For each movement, the user is provided with detailed information to support execution decisions: movement type, positions involved, estimated execution time, impact on efficiency (only for permutation movements), aisle and owner associated with each position involved.

The estimated execution time is calculated using the time functions presented earlier. For internal replenishment movements, the estimate considers the time needed for the operator to travel to the initial reserve position, retrieve the pallet, travel to the destination picking position, and store the pallet. For permutation movements, the calculation includes traveling to the first position, retrieving the pallet, traveling to the second position, retrieving the existing pallet, placing the pallet from the first position into the second, returning to the first position, and placing the second pallet in its new location. These time estimates are crucial for operational managers to assess how many and which operations can be executed within the available time without disrupting other daily warehouse activities.

Interaction with Use Case 2 is carried out through a dedicated page in the desktop version of the WMS, designed for operational management. After generating the suggestions, the user can apply filters to narrow the list by movement type, customer, or aisle. These filters allow the execution of the use case to be aligned with ongoing warehouse operations, minimizing conflicts with other activities such as storage or active picking tasks.

Once filters are applied and suggestions are reviewed, the operational manager selects the movements to execute. Upon confirmation in the WMS, the corresponding tasks are automatically sent to the PDAs of the operators responsible for the physical pallet movements, ensuring coordinated, fast, and fully integrated execution within the existing system.

4.5.2.4 Expected Impact

The impact of Use Case 2 can vary over time, depending on the level of alignment between the actual warehouse arrangement and the macro plan – this can be measured through the layout efficiency. In the initial stage, when the algorithms are first introduced and the current arrangement shows significant misalignment with the macro plan, permutation movements tend to have a strong effect on improving efficiency. The strategic swapping of pallets between positions allows the actual warehouse layout to be brought closer to the ideal configuration relatively quickly, generating performance gains in the picking process.

Over time, however, as Use Case 1 is continuously applied to the reception of new products, and as outbound shipments gradually reduce the presence of products stored before the algorithms' implementation, the warehouse arrangement naturally evolves toward configurations that are better aligned with the macro plan. Under these conditions, both the frequency and the relevance of permutation movements tend to decrease, as the layout will already be largely optimized.

Despite this trend, certain events can reintroduce the need for impactful permutations, such as:

1. **Updates to the macro plan**, either during periodic revisions or due to the onboarding of new owners that require its recalculation.
2. **Changes in the COI class assigned to products**, for example, as a result of demand fluctuations, requiring the relocation of specific items to positions better suited to their new class.

Regardless of the expected decline in permutations over time, internal replenishment movements remain critical, in scenarios that include reserve zones. Their regular execution ensures that the picking area consistently maintains sufficient stock to cover the protection period, avoiding inefficient picking operations in the reserve area. As such, UC2 remains a relevant and valuable operational support tool, even when overall alignment with the macro plan is high.

5 Results

In this section, the results obtained from the application of the proposed methodology are presented, enabling the evaluation of the efficiency of the different storage policies tested, as well as the impact of the developed Use Cases.

The first subsection explores the historical order data that serves as input for the simulation. Next, the simulation results for each storage policy are compared using the previously presented KPI: the average picking time per order, adjusted to include the internal replenishment activities inherent to policies with a reserve area. This key indicator makes it possible to identify the policy that minimizes the operational effort required to prepare orders, in terms of time.

Based on the selected macro plan, the section also examines the expected impact of the Use Cases, quantifying their contribution to layout efficiency over time and demonstrating how they enable the practical implementation of an arrangement converging toward the macro plan.

Thus, the results presented in this section are essential for identifying the most suitable policy for this warehouse and, together with the Use Case impact analyses, provide a valuable contribution to the discussion that will be presented in the following section.

5.1 Simulation Data

First, it is important to understand the data that will be used in the simulation, namely the orders received by this warehouse over the past 12 months. The exploratory analysis of these data shows that a total of 6,682 orders from 9 different owners were included. The distribution of weekly orders over time is presented in *Figure 5*.

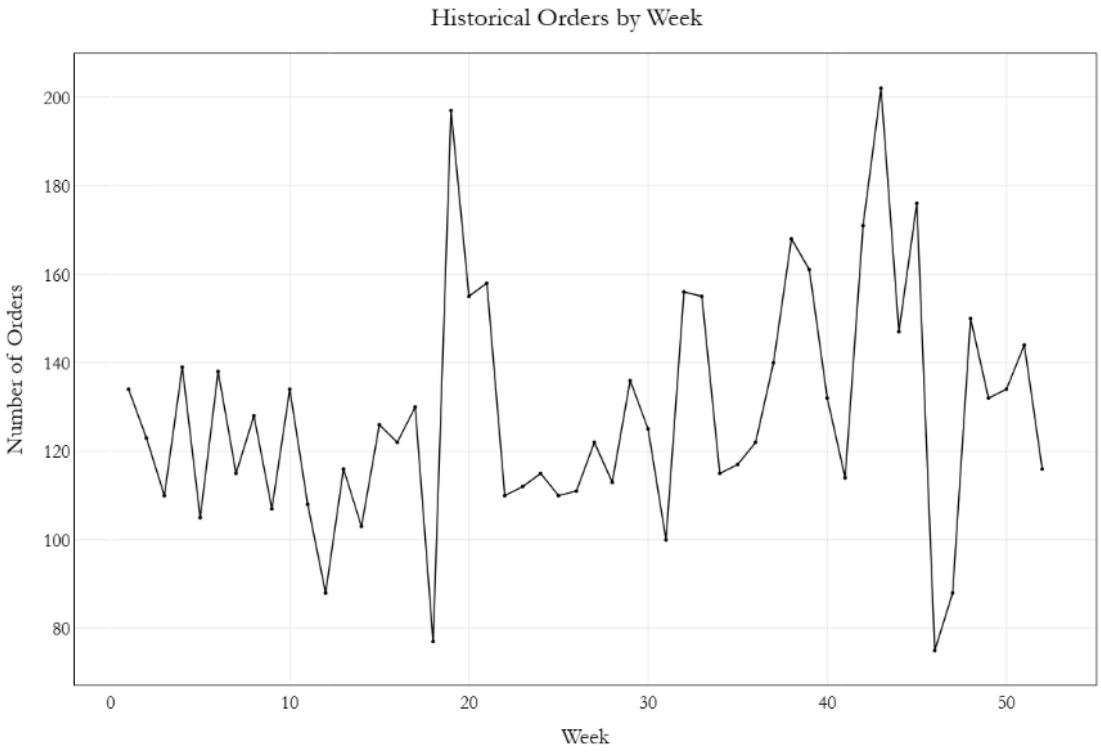


Figure 5 - Weekly order volume over the past 12 months

From the analysis of the figure, it can be observed that in most weeks the number of orders ranges between 100 and 140. However, there are some abrupt fluctuations, including atypical weeks of lower demand, with around 80 orders, and peak weeks with nearly 200 orders. From a statistical perspective, the average number of weekly orders was 128.5, with a standard deviation of 26.67 and a coefficient of variation of 20.75. This indicates a relatively high level of variability, meaning that weekly order volumes tend to fluctuate considerably around the mean.

Another relevant factor concerns the size of the orders, measured by the number of distinct products requested. The bar chart in *Figure 6* shows the percentage distribution of orders according to their size.

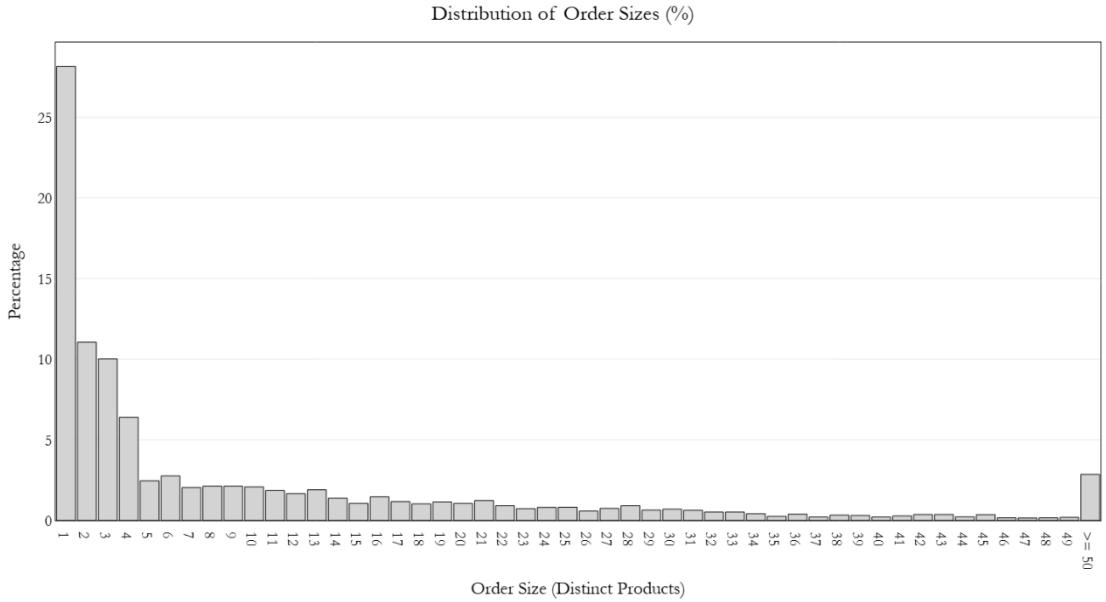


Figure 6 - Percentage distribution of the number of distinct products per order

As can be seen, the most frequent order size is 1, with 28.15% of all orders consisting of a single product. There is also a considerable percentage of orders with 2, 3, and 4 products, corresponding to 11.05%, 10.01%, and 6.4% of orders, respectively. After these sizes, the frequency decreases rapidly; however, the distribution exhibits a long tail, with some cases of orders containing several dozen products, including extreme cases of 50 or more products, which represent only 2.86% of the total.

Overall, 55.61% of orders contain fewer than 5 products, indicating that the warehouse predominantly handles small orders, but occasionally receives large orders as well.

5.2 Comparative Results Across Scenarios

This subsection presents and analyzes the results of the simulation. These results are crucial as they reveal which scenario is the most efficient, that is, minimizes the picking times.

It should be recalled that the storage plan generated for each scenario only includes the assignment of a class and a function to each location (in addition to the assignment to an owner in the relevant scenarios). To execute the simulation, it is necessary to assign specific positions to products according to their characteristics and the plan being simulated. As explained in the methodology, this assignment was performed randomly within each class to reflect what would happen in practice. Given the effect of randomness on the assignments, the simulation was repeated 100 times for each scenario, with a new random assignment

generated in each run. This approach allows for the evaluation of both the average performance and the variability of results for each storage policy under allocation uncertainty.

The average picking time per order obtained in the simulations of each scenario is shown in *Figure 7*. Each boxplot represents the values obtained across the 100 runs for that scenario.

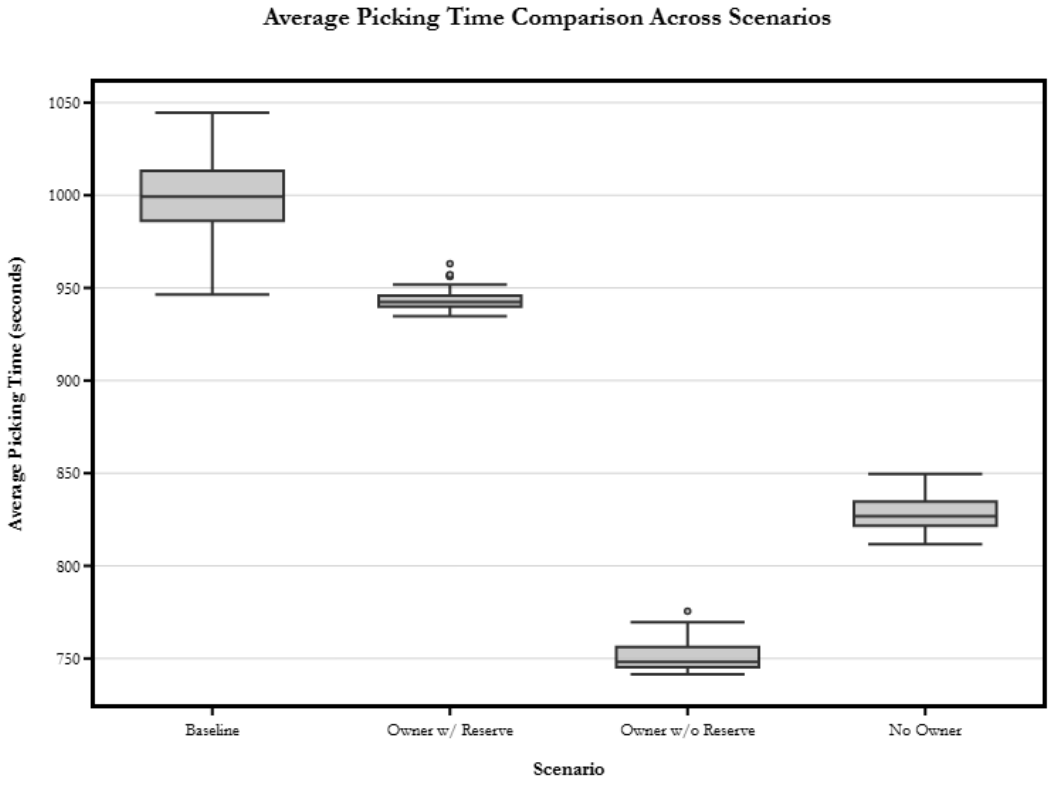


Figure 7 - Boxplots of average picking time per order for each scenario (100 repetitions per scenario)

The mean, standard deviation, and coefficient of variation obtained for each scenario are also presented in *Table 3*, along with the percentage variation of the mean picking time relative to the *Baseline* scenario.

Table 3 - Statistical summary of the results of each scenario

Scenario	Mean (s)	SD (s)	CV (%)	Relative Improvement (%)
Baseline	998.94	19.69	1.97%	–
Owner w/ Reserve	943.37	5.66	0.60%	-5.56%
Owner w/o Reserve	750.63	6.81	0.91%	-24.86%
No Owner	828.36	9.34	1.13%	-17.08%

The combined analysis of *Figure 7* and *Table 3* shows that the *Baseline* scenario has both the highest average picking time and the greatest dispersion, indicating that performance varies considerably between runs. This suggests that the policy is highly sensitive to how products are arranged and offers little operational predictability.

The *Owner w/ Reserve* scenario displayed very low variability, showing that it is the least sensitive to random product placement within classes. However, its average performance is only slightly better than the baseline, reducing the average picking time per order by just 5.56%. The gains provided by this policy are clearly smaller compared with the other policies tested.

The *Owner w/o Reserve* scenario achieved the lowest average and also exhibited low variability. This means that it not only proved to be the most efficient in terms of average picking time but also delivered consistent results across runs, being less sensitive to product placement randomness than both the *Baseline* and *No Owner* scenarios. This policy reduced the average picking time by 24.86% relative to the baseline.

Finally, the *No Owner* scenario produced gains of 17.08% compared to the baseline but lagged behind the previous scenario. Moreover, its variability was higher than in scenarios with owner-based zoning, showing that removing this restriction reduces the predictability of operational efficiency.

From the perspective of results analysis, it is important to detail, in scenarios with reserve zones, the impact of picking from a smaller area and to quantify the additional effort introduced by internal replenishment movements. To support this analysis, the stacked bar chart in *Figure 8* is presented. This chart shows the average picking time obtained for each scenario. In scenarios without reserve zones, this corresponds only to the pure picking time, while in scenarios with reserve zones, the bars are divided into two components: pure picking time and the time of internal replenishments, which was distributed across picking tasks, as explained in the methodology.

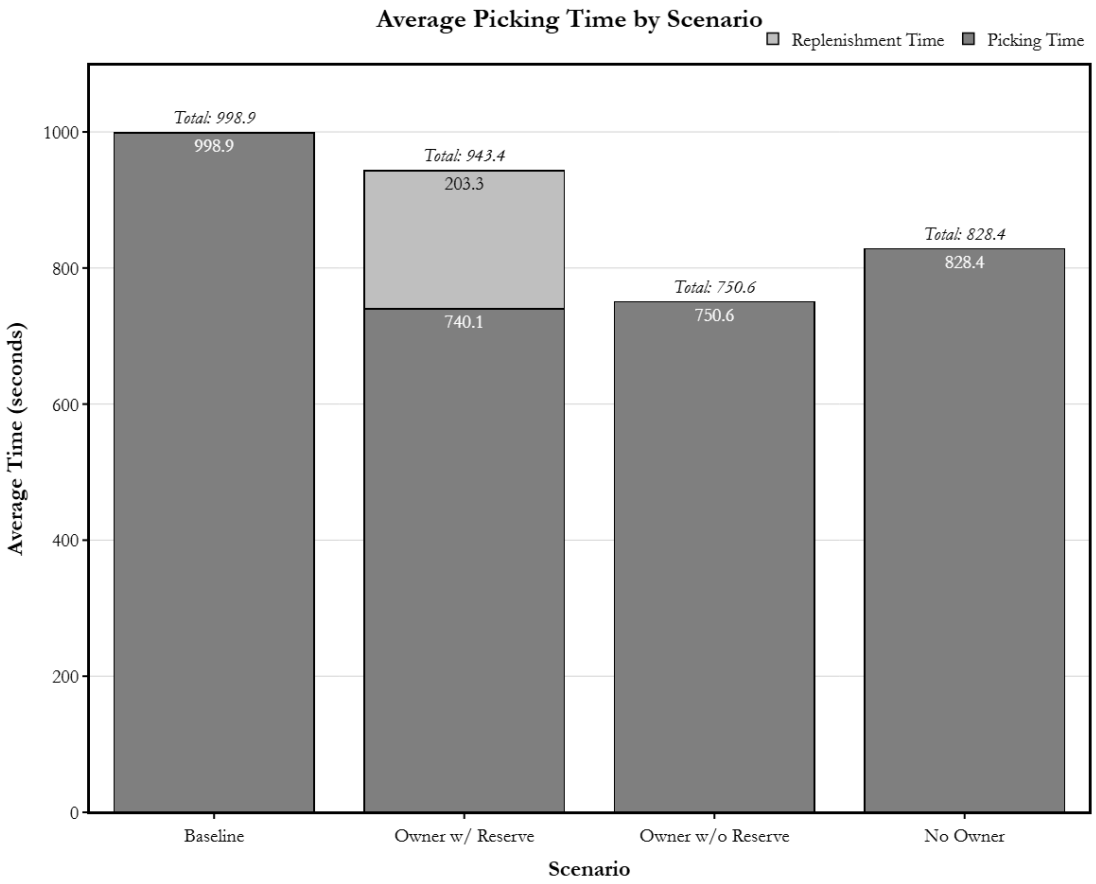


Figure 8 - Stacked bar chart with decomposition in replenishment and pure picking time for each scenario

The chart shows that, although the picking/reserve separation reduces pure picking time – as expected, since picking is done in a smaller area – the additional effort of replenishment more than cancels out this advantage. In other words, the reduction of the picking area does not compensate for the overhead introduced by internal replenishments when assessing the overall operational effort.

Based on the results obtained, the *Owner w/o Reserve* scenario is the one that minimizes the average picking time while also presenting low variability across runs. Therefore, its storage plan will be adopted as the macro plan and used by the Use Cases to guide the generation of recommendations.

5.3 Impact of the Use Cases

Using the selected macro plan and the current state of the warehouse, it is possible to estimate the initial impact of the Use Cases on layout efficiency.

In the case of UC1, we can estimate its impact over time in a situation where it is the only use case being executed. In other words, given the current state of the warehouse, we can project how efficiency would evolve if UC1 were continuously applied. For this analysis, two factors must be considered: the reception of new products and the shipping of products that were already stored in the warehouse before the implementation of UC1.

First, the current layout efficiency was measured. Then, the estimated shipping date of each support was calculated based on the demand for that product and the supports currently in stock, always considering a FEFO (First Expired, First Out) shipping policy. With the forecasted shipping date of each support, it becomes possible to predict when each misplaced support will be shipped and will no longer negatively affect efficiency. On the other hand, for newly received products, it was assumed that the storage recommendations would be fully followed and that all products would be stored in positions belonging to their respective class. It was also assumed that warehouse stock remains constant, with every shipped support being replaced by a new one. Based on these assumptions, the graph in *Figure 9* was created, showing the projected evolution of efficiency over the weeks through the exclusive application of UC1.

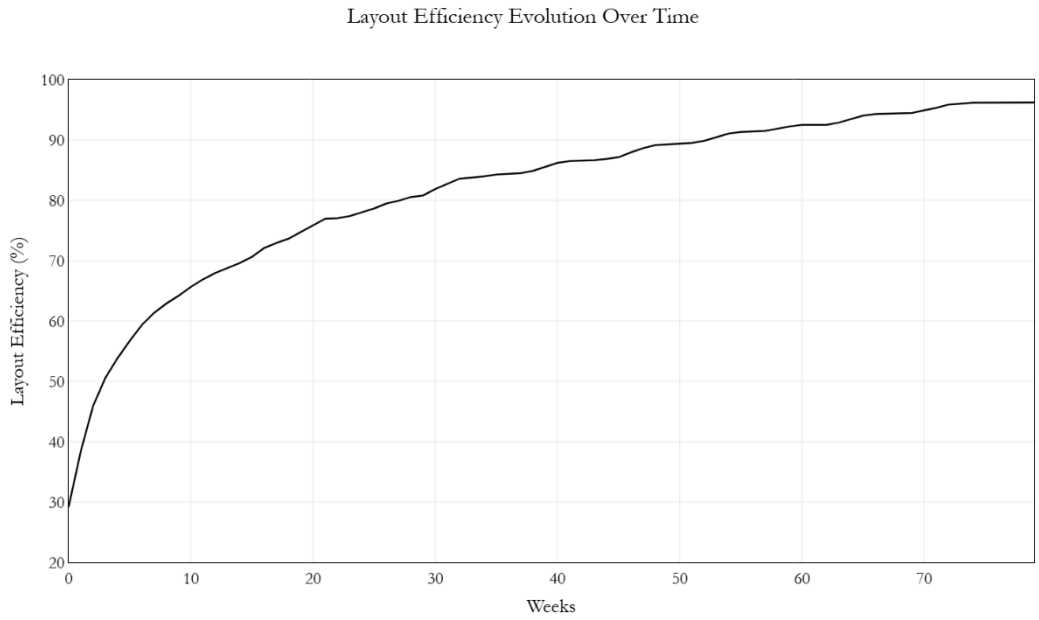


Figure 9 - Projected evolution of layout efficiency under exclusive application of UC1

The results show that the initial layout efficiency is 29.2%. Moreover, the graph reveals that if this use case were implemented on its own, its impact would gradually decrease over time, with more significant improvements in the first weeks, allowing efficiency to reach approximately 65% within 10 weeks. However, in the following weeks, the impact diminishes progressively, resulting in only minor weekly improvements.

It should also be noted that efficiency never reaches 100%, instead plateauing at 96.2% by week 74, due to products whose forecast indicates they will never be shipped. This occurs in the case of new products in the warehouse, i.e., products that have never been shipped before and therefore have a demand forecast of zero based on historical data. The fact that demand for these items cannot be forecasted is a limitation, but the percentage they represent shows that their weight in stock is residual.

For UC2, we executed a run of the algorithm and analyzed the suggestions generated based on the current state of the warehouse. All the suggestions generated were permutation movements, since the macro plan does not include a reserve zone, making internal replenishments unnecessary. Furthermore, we analyzed the relationship between cumulative execution time and cumulative efficiency improvement resulting from the hypothetical execution of the suggested swaps. This relationship is represented in the graph in *Figure 10*.

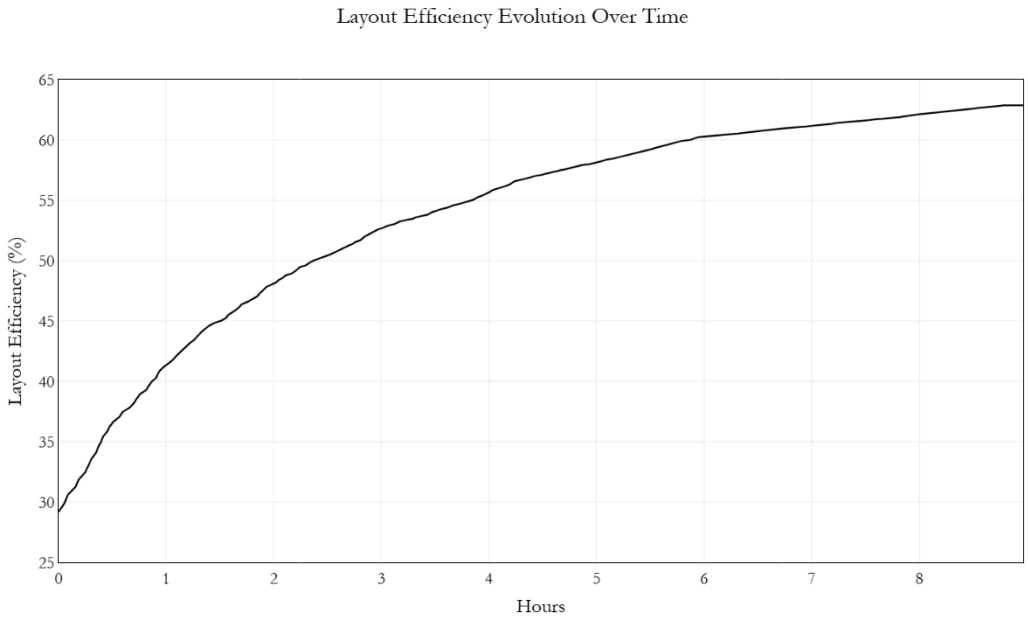


Figure 10 - Relationship between cumulative execution time of UC2 suggestions and layout efficiency improvement

The results show that if all suggestions were executed, the estimated execution time would be 8.97 hours and the impact on efficiency would be 33.7 percentage points. Here as well, the impact of the swaps on efficiency decreases marginally, since the swaps are ordered in descending order of impact and therefore the most impactful swaps appear first.

From the graph, it can be seen that this UC on its own does not allow efficiency to exceed 62.9% in its first execution. This limitation is explained by the existence of pallets containing products from multiple COIs. Since UC2 only moves full pallets, it is impossible to place these pallets in a position that is simultaneously suitable for the COI of all their products.

The results of the initial impact analysis of the UCs allow us to draw conclusions not only about the added value of each one individually but also about their complementarity. These conclusions will be explored in the next section.

6 Discussion

Having presented the results, it is now essential to reflect on their implications. The following discussion is divided into two parts. The first one focuses on exploring the impact of different storage policies and understanding why certain strategies are more suited to the warehouse under study. The second part explores the impact of the proposed use cases, showing how they interact and complement each other in practice.

6.1 Ideal Storage Plan

The most efficient storage policy for a warehouse depends on its specific context. Therefore, the analysis of historical orders provides essential support for interpreting the scenario results. As highlighted in the previous section, there is significant variability in the number of weekly orders, as confirmed by the coefficient of variation of 20.75. Given these fluctuations and the occurrence of peak-demand weeks, it is especially important that picking operations are optimized to strengthen the warehouse's ability to meet demand while maintaining a high service level.

Looking at the distribution of orders by size, single-item orders are the most frequent. From an initial perspective, this could suggest that a scenario without owner-based zoning might be advantageous, since single-product orders do not benefit from product complementarity, unlike multi-product orders. In multi-product orders, where several items must be retrieved during the same route, it is possible to shorten route times if products belonging to the same owner are located near each other – remember that owners can only request orders with their own products, meaning that all items in a multi-product order always belong to the same owner. This effect disappears in single-product routes. However, despite being the most frequent, single-product orders account for only 28.15% of all cases, meaning that, in the vast majority of orders, product complementarity can play a decisive role. Moreover, when an owner's stock is distributed across the warehouse, large multi-product orders can disproportionately increase average picking times, as dispersed storage forces pickers to traverse multiple aisles. Understanding the interaction of these factors is key to interpreting the results.

Turning to the scenario comparison results, all tested scenarios reduced the average picking time compared to the baseline. This finding support the conclusion that COI-based policies outperform the storage policy previously adopted.

However, the results also show that not all scenarios deliver the same level of improvement. The *Owner w/o Reserve* scenario stands out with a significant 24.86% reduction in average picking time compared to the baseline, while the *No Owner* scenario only achieves a 17.08% reduction. These values highlight the trade-off mentioned earlier between allocating the best products to the best positions regardless of ownership, versus keeping products of the same owner close to each other, even if that means that some high-priority class A products end up farther from the docks. The results clearly demonstrate that, in the context of the warehouse under study, zoning by owner enables more efficient picking, suggesting that product complementarity is highly relevant in this case. In the *No Owner* scenario, picking routes often extend across many aisles, particularly for large orders with multiple products, forcing operators to travel longer distances and resulting in higher picking times. Additionally, it is worth noting that the coefficient of variation of the simulation results is higher in the *No Owner* scenario. This is expected, since greater freedom in product placement amplifies the impact of randomness on results. In other words, while zoning by owner constrains where each product can be stored, a *No Owner* scenario leaves wide areas for COI A, B, and C products, increasing the range of possible positions and, consequently, the variability in picking times.

It is also instructive to compare the two owner-zoned scenarios to assess the impact of having or not having a division between picking and reserve areas. The results clearly show that eliminating the reserve area leads to lower times. However, *Figure 8*, which breaks down pure picking effort versus internal replenishment in the reserve scenario, provides valuable insights. As can be seen, operators actually spend less time performing the picking activity itself (retrieving the required products from storage positions) when the picking zone is smaller due to the existence of a reserve zone. Yet, as previously explained, this scenario requires internal replenishments. When the replenishment effort is factored in (distributed across picking times to allow a fair comparison), it becomes evident that the total operational effort is significantly higher in the reserve scenario. This supports the conclusion that, in the warehouse under analysis, avoiding the separation between picking and reserve is advantageous, as the replenishment overhead outweighs the gains in pure picking time.

Furthermore, comparing the two scenarios shows that the reduction in pure picking time with a reserve zone is minimal, dropping only from 750.6 to 740.1 seconds. This modest reduction in pure picking time when a one-week picking zone was introduced suggests that the effective size of the designated picking zone is only marginally smaller than the full

picking area in scenarios without reserve, which can be caused by the consolidation logic that tops up partially filled picking positions. Moreover, class A items occupy the prime slots under both policies, so any relocation mainly affects class B and C positions, as those accumulate the increase/decrease of previous classes. As B and C items contribute less to the total picking workload, degradations in their relative positions have only a limited impact on average picking time. In addition, non-travel components (retrieval/handling) and the routing policy attenuate the potential benefits of slightly reduced travel distances. Taken together, these structural and operational factors explain why shrinking the designated picking zone produced only a small improvement in pure picking time.

The reserve scenario also exhibits a lower CV, which is expected, since a smaller picking zone reduces the degrees of freedom through which randomness can influence picking time results.

Based on all the results presented and the points discussed here, it is reasonable to conclude that, for the warehouse in question, zoning by owner combined with no separation between picking and reserve areas is the most effective configuration. This leads to the adoption of the *Owner w/o Reserve* scenario as the macro plan, as it fully reflects these parameters.

6.2 Use Cases Discussion

The results of the estimated impact of the use cases reveal how the proposed methodology reflects in measurable changes in the warehouse and highlight their complementary roles.

The initial layout efficiency was 29.2%, which indicates that the storage policy previously in place was far from optimal, with many products stored inefficiently. This means that class A products were placed in positions harder to access, while prime locations were occupied by class B or C products.

The estimated impact of UC1 shows that, if applied solely, it would achieve a significant efficiency increase during the first weeks. This improvement is driven by the correct storage of new products but also by product shipping, which removes items from the warehouse. Among the shipped products, some were correctly placed while others were not, but since all of them were replaced by well-placed products – under UC1 storage position recommendations – efficiency rose considerably. Over time, as fewer products remain misallocated, the rate of substitution of poorly placed items with well-placed ones decreases. This explains the declining marginal impact of UC1. Another contributing factor

to this effect is the presence of low-rotation products, mostly class C, which, due to their low demand, can take a long time to be shipped. Until they leave the warehouse, these products remain poorly stored, assuming only UC1 is being used, continuously impacting efficiency.

The projection also shows that efficiency would never reach 100%. As highlighted in the results, this occurs because of products that are stored but have never been shipped, such as new products with zero historical demand, making it impossible to forecast their shipping dates based on historical data. Although such products represented less than 4% of the stock at the time of analysis, they still pose a limitation: in their initial stage, UC1 may not provide optimal recommendations for these items. These products are initially classified as class C but may later prove to belong to a higher COI category. However, as soon as demand data becomes available, along with the storage positions occupied by this products, their COI classification can gradually be corrected. The more data points are accumulated, the more accurate these classifications become.

In any case, this analysis demonstrates that UC1 on its own could eventually raise layout efficiency to very high levels, but it would take considerable time to do so. For instance, reaching 90% efficiency would require more than 50 weeks.

Looking at the projected impact of UC2, the situation is markedly different. Through pallet-level swaps of products already stored, UC2 can achieve much faster improvements in layout efficiency. The estimated execution time of the suggestions from the first run indicates that, in approximately 9 labor hours, efficiency could be increased by 33.7 percentage points to 62.9%. It is important to note that this time refers to a single operator working continuously. In practice, a team of operators could execute the swaps in parallel, and the required time would decrease proportionally to the team size – at least up to small or medium teams working in different aisles, otherwise congestion effects could arise, potentially lowering productivity.

Despite its much faster impact, UC2 also shows a lower ceiling. In its initial execution, the swaps could only raise efficiency to 62.9%. This is a direct consequence of the decision to operate at the pallet level rather than the support level. For pallets containing products from different COIs, it is impossible to position them in a location aligned with all their products' COIs. At an early stage, when many pallets are still heterogeneous, this limitation is particularly evident. Over time, however, this factor will become less restrictive, as pallets will tend to become more uniform in COI composition, if UC1 is used.

The combined analysis of the two use cases demonstrates their strong complementarity. UC1 ensures that new products are immediately stored correctly, with fine-grained recommendations at the support level, but struggles to quickly raise efficiency due to its granularity and inability to act on low-rotation products that remain inefficiently stored for long periods. UC2, in contrast, directly targets misallocated products already in storage, moving them to more suitable positions and removing their negative impact on efficiency even if they are not shipped soon. Thanks to its coarser granularity, UC2 delivers rapid efficiency gains – on the scale of hours compared to the weeks required by UC1 – but cannot achieve very high efficiency levels until pallets become harmonized by COI.

Therefore, these results prove that the combined use of UC1 and UC2 is essential. Together, they ensure that all supports are addressed in the optimization process, while also delivering short-term impact. This enables a faster convergence toward the ideal arrangement defined by the macro plan, with immediate reflection in reduced picking times.

7 Limitations and Future Work

Despite the relevance of the results obtained, it is important to acknowledge that the approach developed presents certain limitations that may constrain the flawless and seamless implementation of the proposed model. Recognizing these limitations helps properly frame the scope of the conclusions without undermining the contributions of this work to the study of order picking optimization. At the same time, this analysis allows the identification of several opportunities for future research that may deepen and expand the contributions achieved here.

From the outset, it was emphasized that this project aimed to implement a data-driven approach to the optimization of picking operations. Such an approach brings clear advantages, such as the systematization of decisions and the analytical optimization of storage policies. However, a fundamental limitation emerges: the difficulty of performing effectively in scenarios where data is lacking. This occurs, for instance, with new products. As described in the methodology, several key steps are based on historical data (e.g., the calculation of demand and the estimation of the quantity of a product that can be stored in a position). For new products, historical data is absent, meaning that initial recommendations may not always be the most accurate. Over time, as data is collected for these products, calculations can be refined, and the recommendations become more reliable as the data grows. A direct implication of this limitation is that the approach presented here would hardly be applicable in a warehouse starting operations from scratch, since the necessary historical data would not exist. However, the methodology is well-suited to warehouses that have been in operation for a sufficient period of time but whose storage organization is not yet optimized.

Another limitation relates to the fact that estimates were used to determine the quantity of a product that fits into a storage position. If these estimates are inaccurate – due, for example, to stock registration errors in the system or to product incompatibilities caused by specific shapes or packaging – some recommendations, particularly from UC1, may prove infeasible, with products not fitting into the suggested positions. Nevertheless, although imperfect, these estimates still represent a significant improvement. Without them, it would not be possible to recommend the consolidation of products into partially filled positions, forcing the use of a new position each time and compromising the efficient use of available space. Therefore, even if occasional mismatches require operators to disregard certain UC

recommendations, their use still constitutes an improvement over a scenario where new positions were always opened or where product placement was based solely on operators' intuition.

A further limitation lies in the fact that the simulation model may not capture all informal operator behaviors. This limitation has two consequences: first, in the time functions, which assume constant work during task execution and do not account for eventual informal breaks; and second, in the baseline scenario efficiency estimates, which may be affected if informal habits exist, such as placing frequently picked or heavier-to-handle products in more accessible positions.

Looking ahead, several opportunities for future research arise. One possible direction is the development of a simulation model capable of incorporating congestion effects that may impact productivity. The main challenge lies in identifying which orders are prepared simultaneously, since congestion can only occur in those circumstances. The simulation of the current routing policy itself is already accounted for in the current model.

Additionally, further efficiency gains may be achieved by jointly optimizing not only the storage policy but also the routing policy, which was outside the scope of this project. Since the operational efficiency of picking is interdependent on both policies, a combined optimization could create synergies leading to substantial reductions in picking time.

Finally, a major step for future work would be the application of this methodology to other warehouses. As demonstrated, the methodology is generalizable and can be applied to other contexts, provided the required data is available. Necessary adjustments would have to be made to account for operational constraints specific to each warehouse (e.g., weight restrictions by rack level) and to map the new layout. However, once this groundwork is completed, both the macro-plan generation and the implementation of the use cases could be replicated, yielding improvements in picking efficiency.

8 Conclusion

This dissertation set out to address the problem of optimizing picking tasks in a warehouse operating as a 3PL provider. As picking is the most time-consuming and costly activity in such a context, its optimization has a significant impact on the overall efficiency of warehouse operations. The objective was to develop a data-driven approach capable of minimizing the long-term average picking time.

It quickly became evident that picking efficiency is directly linked to the storage policy. Therefore, the methodology was developed with the purpose of identifying the most effective storage policy and defining how it could be implemented in the warehouse under study.

Once the warehouse reality was modeled – starting with the mapping of the layout, followed by the development of time estimation functions, and the construction of a simulation model capable of reproducing the preparation of historical orders under different storage policies – it was possible to test alternative strategies and to identify the most efficient ones for this specific warehouse.

The results showed that COI-based policies reduce picking times when compared to the previous policy, which largely relied on operators' intuition during storage and was limited to the assignment of aisles to owners. Furthermore, the results indicated that maintaining owner-based zoning is beneficial, underscoring the importance of storing complementary products close to each other, and that separating picking and reserve stock does not make sense in this warehouse, as the replenishment overhead outweighs the gains from operating within a smaller picking area. Among the evaluated configurations, the most efficient scenario reduced average order picking time by 24.86% compared to the previous practice.

The ideal storage policy – here referred to as the macro plan – then served as the foundation for the use cases. The two developed use cases are the tools that enable the practical implementation of this policy. UC1 supports operators in deciding where to place incoming products, ensuring that items are stored in their optimal positions from the moment they enter the warehouse. UC2 allows for the rearrangement of already stored products, correcting misplacements either from the past or arising over time due to changes in demand. The impact estimates demonstrated that UC2 can deliver significant efficiency gains in a short period, while UC1 enables finer-grained placement at the support level rather

than at the pallet level, which is essential to approach near 100% efficiency levels. These results highlight the high degree of complementarity between the use cases and the importance of employing them together. Additionally, their integration into the warehouse's existing WMS represents an added value, promoting adoption and ensuring a seamless user experience.

This project makes three main scientific contributions to the research on order-picking optimization, addressing gaps in the existing literature. First, optimizing picking in multi-product storage warehouses without product volume data was a major challenge that had not yet been studied. The approach developed here provides a robust way to handle this scenario by estimating the necessary parameters to identify the ideal storage plan and generate actionable recommendations in the use cases, representing a significant step toward optimizing operations in warehouses under such conditions.

Second, this project not only identified the optimal storage policy but also developed practical tools to implement it: the use cases. This aspect substantially increases the usefulness of the methodology, as it bridges the gap between theoretical research and operational application.

Finally, a distinguishing feature of this methodology is that, despite the scientific rigor and effort required for its development, its maintenance and replication in other warehouses are straightforward. The layout mapping performed in Excel is a clear example of leveraging a widely available tool that can be used without advanced technical qualifications, and the Python code that processes this input is directly applicable. Moreover, all code developed to calculate the COI and generate the macro plan will remain available to the company, ready to be run autonomously whenever necessary – whether to support new warehouses or to update the parameters of the one addressed in this project.

In conclusion, this project was highly successful in achieving its objectives: it was able to identify a storage policy that significantly reduces average picking time and to develop tools that enable the practical implementation of this policy in an operational setting. The result is a robust, easy-to-maintain approach that is also replicable across other warehouses.

References

Boysen, N., & de Koster, R. (2025). 50 years of warehousing research—An operations research perspective. *European Journal of Operational Research*, *320*(3), 449–464. https://doi.org/10.1016/j.ejor.2024.03.026

Caron, F., Marchet, G., & Perego, A. (2010). Optimal layout in low-level picker-to-part systems. *International Journal of Production Research*, *38*, 101–117. https://doi.org/10.1080/002075400189608

Chen, T.-L., Cheng, C.-Y., Chen, Y.-Y., & Chan, L.-K. (2015). An efficient hybrid algorithm for integrated order batching, sequencing and routing problem. *International Journal of Production Economics*, *159*, 158–167. https://doi.org/10.1016/j.ijpe.2014.09.029

Choe, K., & Sharp, G. P. (1991). *Small Parts Order Picking: Design and Operation*. https://www2.isye.gatech.edu/~mgoetsch/cali/Logistics%20Tutorial/order/article.htm

de Koster, R., Le-Duc, T., & Roodbergen, K. J. (2007). Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, *182*(2), 481–501. https://doi.org/10.1016/j.ejor.2006.07.009

Foead, D., Ghifari, A., Kusuma, M. B., Hanafiah, N., & Gunawan, E. (2021). A Systematic Literature Review of A* Pathfinding. *Procedia Computer Science*, *179*, 507–514. https://doi.org/10.1016/j.procs.2021.01.034

Gademann, N., & Velde, S. (2005). Order batching to minimize total travel time in a parallel-aisle warehouse. *IIE Transactions*, *37*(1), 63–75. https://doi.org/10.1080/07408170590516917

Gademann, Van Den Berg, J. P., & Van Der Hoff, H. H. (2001). An order batching algorithm for wave picking in a parallel-aisle warehouse. *IIE Transactions*, *33*(5), 385–398. https://doi.org/10.1023/A:1011049113445

Gu, J., Goetschalckx, M., & McGinnis, L. F. (2007). Research on warehouse operation: A comprehensive review. *European Journal of Operational Research*, *177*(1), 1–21. https://doi.org/10.1016/j.ejor.2006.02.025

Hackman, S. T., & Bartholdi, J. J. (2019). *Warehouse & Distribution Science* (Release 0.98.1). The Supply Chain & Logistics Institute, H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology.

Hall, R. W. (1993). Distance Approximations for Routing Manual Pickers in a Warehouse. *IIE Transactions*, *25*(4), 76–87. https://doi.org/10.1080/07408179308964306

Hausman, W. H., Schwarz, L. B., & Graves, S. C. (1976). Optimal Storage Assignment in Automatic Warehousing Systems. *Management Science*, *22*(6), 629–638. https://doi.org/10.1287/mnsc.22.6.629

Heragu, S., Du, L., Mantel, R., & Schuur, P. (2005). Mathematical model for warehouse design and product allocation. *International Journal of Production Research*, *43*, 327–338. https://doi.org/10.1080/00207540412331285841

Heskett, J. L. (1963). Cube-per-order index – A key to warehouse stock location. *Transport and Distribution Management 3*, 27–31.

Kallina, C., & Lynn, J. (1976). Application of the Cube-Per-Order Index Rule for Stock Location in a Distribution Warehouse. *Interfaces*, *7*(1), 37–46. https://doi.org/10.1287/inte.7.1.37

Lambert, D. M., & Cooper, M. C. (2000). Issues in Supply Chain Management. *Industrial Marketing Management*, *29*(1), 65–83. https://doi.org/10.1016/S0019-8501(99)00113-3

Lara, C. L., & Wassick, J. (2023). *Future of Supply Chain: Challenges, Trends, and Prospects* (No. arXiv:2301.13174). arXiv. https://doi.org/10.48550/arXiv.2301.13174

Malmborg, C. J., & Bhaskaran, K. (1990). A revised proof of optimality for the cube-per-order index rule for stored item location. *Applied Mathematical Modelling*, *14*(2), 87–95. https://doi.org/10.1016/0307-904X(90)90076-H

Meller, R. D., & Gau, K.-Y. (1996). The facility layout problem: Recent and emerging trends and perspectives. *Journal of Manufacturing Systems*, *15*(5), 351–366. https://doi.org/10.1016/0278-6125(96)84198-7

Park, Y. H., & Webster, D. B. (1989). Design of class-based storage racks for minimizing travel time in a three dimensional storage system. *International Journal of Production Research 27 (9)*, 1589–1601.

Petersen, C. G. (1997). An evaluation of order picking routeing policies. *International Journal of Operations & Production Management*, *17*(11), 1098–1111. https://doi.org/10.1108/01443579710177860

Petersen, C. G. (2002). Considerations in order picking zone configuration. *International Journal of Operations & Production Management*, *22*(7), 793–805. https://doi.org/10.1108/01443570210433553

Petersen, C. G., & Aase, G. (2004). A comparison of picking, storage, and routing policies in manual order picking. *International Journal of Production Economics*, *92*(1), 11–19. https://doi.org/10.1016/j.ijpe.2003.09.006

Petersen, C. G., Aase, G. R., & Heiser, D. R. (2004). Improving order-picking performance through the implementation of class-based storage. *International Journal of Physical Distribution & Logistics Management*, *34*(7), 534–544. https://doi.org/10.1108/09600030410552230

Ratliff, H. D., & Rosenthal, A. S. (1983). Order-Picking in a Rectangular Warehouse: A Solvable Case of the Traveling Salesman Problem. *Operations Research*, *31*(3), 507–521. https://doi.org/10.1287/opre.31.3.507

Roodbergen, K. (2001). *Layout and Routing Methods for Warehouses*. Erasmus Universiteit Rotterdam (EUR).

Rushton, A., Croucher, P., & Baker, P. (2014). *The Handbook of Logistics and Distribution Management: Understanding the Supply Chain*. Kogan Page Publishers.

Sprock, T., Murrenhoff, A., & McGinnis, L. F. (2017). A hierarchical approach to warehouse design. *International Journal of Production Research*, *55*(21), 6331–6343. https://doi.org/10.1080/00207543.2016.1241447

Tompkins, J. A., White, J. A., Bozer, Y. A., & Tanchoco, J. M. A. (2010). *Facilities Planning*. John Wiley & Sons.

van den Berg, J. P., Sharp, G. P., Gademann, A. J. R. M. (Noud), & Pochet, Y. (1998). Forward-reserve allocation in a warehouse with unit-load replenishments. *European Journal of Operational Research*, *111*(1), 98–113. https://doi.org/10.1016/S0377-2217(98)80013-1

van Gils, T., Ramaekers, K., Braekers, K., Depaire, B., & Caris, A. (2018). Increasing order picking efficiency by integrating storage, batching, zone picking, and routing policy decisions. *International Journal of Production Economics*, *197*, 243–261. https://doi.org/10.1016/j.ijpe.2017.11.021